

# Прототипирование и концептуальное проектирование

## Оглавление

1. Прототипирование пользовательского интерфейса .....	1
1.1. Прототипы. Размерность .....	2
1.2. Прототипы. Достоверность.....	3
2. Прототипирование .....	4
2.1. Виды прототипирования.....	5
3. Разработка прототипа: от теории к практике .....	6
3.1. Бумажное прототипирование .....	6
3.2. Презентационная версия прототипа .....	7
3.3. Псевдореальная версия прототипа .....	8
3.4. Реальная версия прототипа .....	9
3.5. Раскадровки .....	9
3.6. Инструменты для создания макетов и каркасных моделей .....	10
3.7. Больше прототипов .....	10
4. Концептуальный проект.....	11

Разработка прототипа — средство, позволяющее проанализировать идеи, прежде чем на них будут потрачены время и деньги. Это распространённый в инженерной практике метод. Основная цель, достигаемая при создании прототипа, — это экономия времени и ресурсов.

Ценность прототипа заключается в том, что он является внешней оболочкой-моделью отражающей существенные отношения разрабатываемого продукта. По сравнению с реальным продуктом прототипы просты и недороги в разработке. При минимальном вложении средств можно обнаружить ошибки создателей и юзабилити проблемы, и улучшить пользовательский интерфейс до того, как сделаны значительные инвестиции в окончательную разработку и технологии.

(по материалам сайта <http://www.4stud.info>)

Создание прототипа — следующий шаг в разработке пользовательского интерфейса. Прототипирование позволяет создавать макеты интерфейсов разной степени достоверности: от набросков «на скорую руку» и бумажных прототипов до интерактивных макетов с использованием специальных программ.

## 1. Прототипирование пользовательского интерфейса

В английском языке слово *prototype* (прототип) является существительным или глаголом. Как существительное оно ссылается на первую в своем роде модель, опытный образец или прообраз будущей системы. Как глагол — указывает на процесс разработки дизайна с использованием прототипов (прототипирование). Создание и использование прототипов позволяет дизайнерам проверять их творческие идеи. Почему используют прототипы? Потому что они дешевле, их быстрее создавать и они могут давать обратную связь.

Джефф Хокинс, изобретатель Palm Pilot, вырезал кусок дерева примерно того размера и формы (в конечном счете Palm Pilot), в которых, как он себе представлял, появится его новое устройство. Он носил его собой, выполняя повседневные действия и воображая, как он будет использовать это устройство. Что он узнал? Он, вероятно, уточнил физический размер, решив, что оно (устройство) должно быть тоньше и с закругленными краями. У него, возможно, возникла идея, насколько тяжелым может быть новое устройство. Представляя использование устройства, ему в голову, вероятно, приходили идеи задач, которые оно сможет решать. Держа в руках деревянный брусок и представляя, что это реальный, Хокинс, вероятно, обдумывал, как он сам, а в дальнейшем и пользователь, может взаимодействовать с этим устройством. Когда он показал деревянный брусок своим друзьям, те начали понимать идеи Джеффа.

В истории Дж. Хокинса достаточно морали, чтобы понять, что простой прототип позволяет проверить имеющиеся идеи, породить новые идеи и продемонстрировать их заинтересованным сторонам.

Например, в процессе детального проектирования web-сайтов проектировщик из требований к системе и анализа действий пользователей, а так же с помощью концептуального прототипа (если он есть), создаёт:

- информационную архитектуру продукта (например, дерево сайта или последовательность экранов в интернет-магазине);
- расположение контента и функционала на экранах в соответствии с используемыми сценариями взаимодействия;
- структуру контентных и функциональных блоков на экране;
- различные состояния экранов в процессе работы с пользователем (например, состояние ошибки или пустой поисковой выдачи, и т.п.).

Так же к детальному прототипу может прилагаться спецификация, где описывается реакция продукта на возможные действия пользователя.

### **1.1. Прототипы. Размерность**

*Глобальные прототипы* моделируют систему целиком. Их использование позволяет выявлять проблемы, связанные с *полнотой и непротиворечивостью* пользовательского интерфейса.

*Локальные прототипы* моделируют только небольшую часть системы. Они могут быть использованы для устранения разногласий во мнениях через сопоставление различных вариантов дизайна: достаточно сделать несколько альтернатив и оценить их.

Два вида локальных прототипов, *горизонтальные* и *вертикальные*, направлены на полноту функциональности и диапазон возможностей в прототипе. Горизонтальные прототипы имеют небольшую функциональную глубину, но широки в возможностях. Вертикальные прототипы функционально глубоки, но ограничены в возможностях. Для лучшего понимания приведём две диаграммы, иллюстрирующие отличия горизонтального и вертикального прототипов для некоторой структуры приложения.

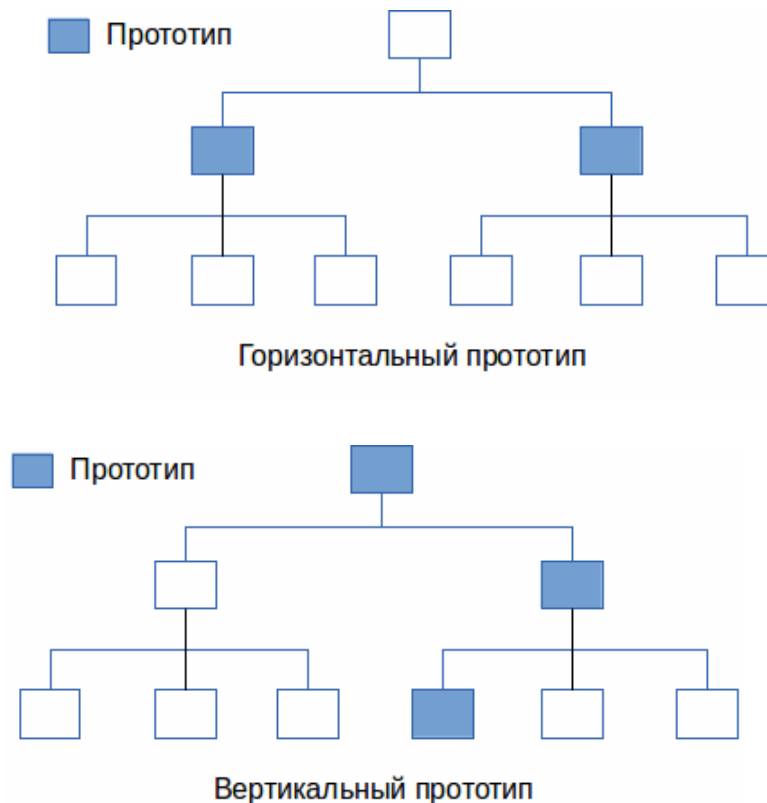


Рис. 1. Прототипы: горизонтальные и вертикальные

## 1.2. Прототипы. Достоверность

Все прототипы можно разделить на прототипы низкой достоверности и прототипы высокой достоверности.

Первые, как правило, мало похожи на окончательный продукт. Они делаются не из того же самого материала, что и окончательное устройство и не имеют всей его функциональности. Прототип низкой достоверности может симулировать некоторую интерактивность, но не отражает всех тонкостей взаимодействия.

Вторые, напротив, выглядят более похожими на законченное устройство. Окончательный, утвержденный дизайн — пример прототипа высокой достоверности. Он может иметь некоторые из функций законченного продукта и позволяет протестировать больше тонкостей взаимодействия. При этом, он требует больше времени на разработку и создание. Зачастую, после создания действующего прототипа высокой достоверности команда разработчиков или менеджеров неохотно отказывается от него и пытается развить из прообраза законченное

устройство. В этом есть минус — такой подход может вести к ограниченному прогрессу в дизайне.

Каждая из указанных разновидностей имеет сильные и слабые стороны (Таблица 1).

Таблица. 1. Преимущества и недостатки прототипов разной степени достоверности

<i>Достоверность прототипа</i>	<i>Преимущества</i>	<i>Недостатки</i>
<b>Низкая</b>	<ul style="list-style-type: none"> <li>• Меньшая стоимость разработки</li> <li>• Возможность оценки множества вариантов дизайнов</li> <li>• Представляет полезную информацию для разработчиков и дизайнеров</li> <li>• Решает проблемы создания макетов экрана</li> <li>• Моет использоваться для определения потребностей рынка</li> <li>• Доказывает или опровергает идеи и концепции</li> </ul>	<ul style="list-style-type: none"> <li>• Ограниченный контроль ошибок</li> <li>• Плохая детализация спецификаций для дальнейшей разработки</li> <li>• Процессом управляет «посредник» (см. <u>facilitator-driven</u>)</li> <li>• Ограниченная полезность после утверждения требований</li> <li>• Ограниченная пригодность для юзабилити-тестирования</li> <li>• Ограничения, связанные с навигацией и потоками активности</li> </ul>
<b>Высокая</b>	<ul style="list-style-type: none"> <li>• (Почти) полная функциональность</li> <li>• Интерактивность</li> <li>• Процесс разработки, управляемый пользователем</li> <li>• Четкая навигационная схема</li> <li>• Является инструментом для исследования и тестирования</li> <li>• Наглядно показывает конечный продукт</li> <li>• Служит «живой» спецификацией</li> <li>• Маркетинговый и торговый инструмент</li> </ul>	<ul style="list-style-type: none"> <li>• Высокая стоимость разработки</li> <li>• Большие затраты времени на создание</li> <li>• Не эффективен для проверки идей и концепций</li> <li>• Не эффективен для формирования окончательных требований</li> </ul>

Прототипы и низкой достоверности могут принимать разные формы и каждая из них позволяет подтвердить или проверить различные аспекты дизайна.

## 2. Прототипирование

Под прототипированием следует понимать ничто иное, как процесс создания прототипа. Хикс и Хартсон (Developing User Interfaces: Ensuring Usability through Product and Process, Wiley, 1993) описывают прототипирование по аналогии с артиллерийской стрельбой: артиллеристы наводят орудие в общем направлении на мишень; следует выстрел; корректировщики оценивают попадание (или промах) и радируют о необходимости корректировки прицела; после регулировки делается следующий выстрел. Так же и прототипирование: чтобы протестировать

дизайнерскую идею (попадание или промах) нужно создать и оценить прототип.

## 2.1. Виды прототипирования

Традиционный подход к разработке макета пользовательского интерфейса основан на переходе от прототипа низкой достоверности к прототипу высокой достоверности (рис. 2). На практике, эта простая и логичная схема выливается в более совершенную технологию *эволюционного прототипирования*.



Рис. 2. Традиционная модель разработки прототипа

**Эволюционное прототипирование** предполагает последовательное увеличение достоверности исходного образца, пока, в конце концов, он не становится законченной системой (рис. 3). Эволюционное прототипирование — широко распространённый подход к разработке интерфейсов, но он таит некоторую опасность: если изначально создается прототип высокой достоверности, его сложно будет расширять для проверки новых идей. Несмотря на это, эволюционное прототипирование может быть полезным для выявления все больших и больших тонкостей в аспектах дизайна и его совершенствования.



Рис. 3. Эволюционное прототипирование

**Быстрое прототипирование** подразумевает, что серии прототипов создаются, а затем, после их оценки и принятия решения о неадекватности модели, отбрасываются. Обычно разрабатываются прототипы все более и более высокой достоверности. Быстрое прототипирование может быть сложным для проектирования командой разработчиков или для приемки менеджерами, потому что выглядит так, как если бы время затраченное на разработку прототипа было потрачено впустую.

**Инкрементное прототипирование** основано на сборке окончательного продукта из нескольких прототипов. Все части (отдельные прототипы) могут разрабатываться параллельно, тем самым сокращая общее время на разработку (рис. 4).



Рис. 4. Инкрементная модель

**Экстремальное прототипирование** (рис. 5) используется при создании веб-приложений. Весь процесс разбивается на три фазы. В первой фазе создается прототип низкой достоверности, состоящий из статических веб-страниц. Во второй фазе создается работоспособный код веб-приложения, а статические веб-страницы переписываются с учетом используемого фреймворка и функциональности, создается полностью работоспособный на уровне модели пользовательский интерфейс. В третьей фазе выполняется интеграция веб-интерфейса со всеми сервисами и ресурсами.



Рис. 5. Экстремальное прототипирование

### 3. Разработка прототипа: от теории к практике

#### 3.1. Бумажное прототипирование

Вы имеете всего лишь несколько дней, оставшихся до представления ваших идей относительно приложения. Нет времени на создание полной модели (которая, на самом деле, есть другой прототип). Что вы можете сделать? Вы можете создать **бумажный прототип**.

Д. МакКракен и Р. Вольф в своей книге *User-Centered Website Development: A Human-Computer Interaction Approach* описывают очень гибкую и эффективную процедуру прототипирования веб-сайтов с помощью бумажного прототипа. Этот метод позволяет смоделировать не только визуальные аспекты веб-сайта, но и некоторые особенности взаимодействия с ним. Этот же метод может быть использован для разработки дизайнов приложений и небольших устройств.

Все, что вам потребуется для создания бумажного прототипа обычно есть под рукой:

- картон, линованная и нелинованная бумага, разноцветная бумага для заметок (sticky-notes), листы прозрачной ацетатной пленки;

- цветные ручки, фломастеры и карандаши;
- канцелярский клей, клейкая лента, клей многократного применения (подобный тому, который наносится на клейкие заметки);
- ножницы, канцелярский нож, хорошая линейка, циркуль.

Достоинствами бумаги являются исключительная простота и скорость рисования. Польза начального прототипирования на бумаге заключается в:

- во-первых, в простоте и скорости рисования,
- во-вторых, в исключительной простоте модификации по результатам тестирования,
- в-третьих, в относительной простоте привлечения представителей целевой аудитории.

Кроме того, бумага помогает не думать о том, как интерфейс выглядит, позволяя сосредоточиться на том, как интерфейс работает. Поэтому не следует делать такой прототип очень «реалистичным», он должен просто отражать функциональность. Бумажная версия прототипа рисуется на этапах постановки задачи и высокоуровневого проектирования.

В своей книге о проектировании веб-сайтов, Д. МакКракен и Р. Вольф предлагают в качестве основы для прототипа всех страниц сайта использовать распечатанный на плотной бумаге скриншот окна браузера. Они же предлагают делать все, что изменяется или исчезает на сайте (панели меню, полосы прокрутки, выпадающие меню, текстовые поля, текстовые блоки и т.п.) из бумаги. В Интернете есть множество готовых шаблонов указанных элементов, которые нужно просто скачать, распечатать, вырезать и использовать (например, шаблоны ОС Андроид). Затем вырезанные из бумаги элементы страницы нужно разместить на шаблоне. По-разному комбинируя элементы макета или заменяя их на другие, вы, в конце концов, найдете подходящий вариант дизайна. Чтобы зафиксировать его, вы можете отсканировать полученные макеты или сфотографировать их.

Этот прототип позволяет моделировать интерактивность путем перемещения, удаления и размещения различных элементов на шаблоне.

**На заметку:** МакКракен и Вольф предлагают установку крайнего срока для изготовления прототипа, потому что многие участники команды разработчиков постоянно будут предлагать какие-нибудь усовершенствования в ходе работы над моделью. Это творческое усердие дизайнеров может длиться вечно, пока у них будут идеи, которые могут быть проверены.

### **3.2. Презентационная версия прототипа**

После исчерпания возможностей бумажной версии прототипа стоит создать новую (следующую) версию. Для этого точно так же рисуется интерфейс, но уже не на бумаге, а в какой-либо презентационной программе. Такая версия прототипа может создаваться уже на этапе высокоуровневого проектирования, а на этапе низкоуровневого проектирования ПИ она обязательна.

Достаточно распространенным инструментом для создания прототипов этого типа является на сегодняшний день *MS Visio*. При работе с *Visio* можно выбрать одну из двух возможностей: можно:

- либо рисовать все экраны на одном листе, соединяя взаимосвязанные элементы управления и экраны линиями,
- либо рисовать каждый экран на отдельном листе, связывая экраны ссылками.

Первый вариант удобен, поскольку позволяет оценить интерфейс в целом, второй предпочтительнее для заказчика и субъектов тестирования, поскольку его легче понять. Как правило, превратить второй вариант в первый оказывается гораздо легче.

Большим достоинством *Visio* является возможность записывать результат в *HTML*-файл, что позволяет без проблем тестировать интерфейс на территории субъектов тестирования.

В настоящее время практически все программы машинного проектирования имеют возможность создания прототипов.

*Mockingbird* — очень популярный инструмент для создания прототипов. Он позволяет легко создавать, просматривать и редактировать макет, а так же обмениваться наработками друг с другом. Сервис обладает красивым и удобным интерфейсом, включает привязку к сетке, перетаскивания, большой набор разнообразных элементов и ряд других полезных функций и возможностей.

*Pencil Project* — скорее большой плагин для FireFox, нежели онлайн сервис или веб-приложение. После его установки создавать схемы, макеты и пользовательские интерфейсы можно прямо в браузере. Имеется большое количество инструментов и функций (многостраничные документы, поддержка текста, экспорт *html*, *png*, файлов *OpenOffice* и многое другое).

*Photoshare* предоставляет собой простой инструмент для создания каркасов сайта. С помощью этого сервиса можно создать прототип любой сложности, так как сервис обладает всеми необходимыми инструментами.

*HotGloo* — очень удобное и популярное средство для разработки макетов. Простой интерфейс содержит широкий набор инструментов, функции и всевозможные средства для создания, редактирования и калибровки прототипа сайта. Различные привязки, функции перетаскивания элементов, обширное количество свойств у объектов. Есть возможность передачи файлов при работе в команде.

*MockFlow* — онлайн и оффлайн сервис с обширной библиотекой шаблонов и различных элементов, количество которых позволяет свободно управлять разработкой и тем самым делать оригинальные и технически сложные прототипы.

Другие программы для прототипирования сайтов можно найти по ссылке: <https://netpeaksoftware.com/ru/blog/10-programm-dlya-sozdaniya-prototipa-sayta>

### **3.3. Псевдореальная версия прототипа**



В тех случаях, когда в интерфейсе появляются нестандартные элементы или необходимо проверить реальную скорость взаимодействия пользователя с системой, создаётся ещё одна версия прототипа — реально выглядящая, но лишенная каких-либо алгоритмов и, соответственно, не показывающая реальных данных. Делать этот вариант можно как в средах разработки, благо в них есть визуальные инструменты создания интерфейсов, так и в редакторах изображений, что обычно быстрее. Создаются снимки экрана, на которых и производят тестирование. Понятно, что как-то модифицировать эти экраны затруднительно, так что лучше не увлекаться такой работой, не получив каких-либо гарантий её правильности. Эта версия прототипа больше соответствует этапу низкоуровневой разработки ПИ, но может применяться и на этапе высокоуровневой разработки.

### **3.4. Реальная версия прототипа**

Иногда необходимо тестировать взаимодействие пользователя не только с интерфейсом системы, но и с обрабатываемыми системой данными. Например, работая с графической программой, пользователь не только нажимает на экранные кнопки, но также создает и модифицирует изображения мышью. Область же редактирования данных зачастую вообще не содержит каких-либо визуальных интерфейсных элементов, из чего вовсе не следует, что интерфейса в ней нет, его наоборот много. Просто счёт в нем идет не на кнопки и переключатели, но на пиксели и миллисекунды.

Понятно, что прототип в таких условиях практически не будет отличаться от готового ПИ. Поэтому лучше всего написать нужные участки программы до написания всего остального и проводить юзабилити-тестирование на реальной версии прототипа ПИ. Понятно, что прототип такой версии, если он все-таки разрабатывается, возможен только на этапе низкоуровневого проектирования.

### **3.5. Раскадровки**

Раскадровки часто используются в фильмах. Видели ли вы дополнительные возможности на DVD-диске фильма «Властелин колец»? Питер Джексон сделал раскадровки всего фильма, чтобы проиллюстрировать его исполнительным продюсерам и разработать сюжет. Вы можете спросить: что общего между кинематографом и интерактивным взаимодействием? Ответ в том, что и то, и другое — дискретные процессы, растянутые во времени.

Раскадровка — это последовательность зарисовок, показывающих, как пользователь продвигается «сквозь» задачу, используя конкретное устройство. Это могут быть эскизы графического пользовательского интерфейса (GUI), скриншоты (как сделать скриншот экрана на различных устройствах) или наброски сцен пользовательского взаимодействия с

программой или устройством. Раскадровки очень хороши для «оживления» сценариев взаимодействия.

Для того, чтобы создавать раскадровки интерфейса не обязательно быть художником. Здесь не нужна фотореалистичность, поскольку каждая сцена раскадровки является, в первую очередь, визуальным средством общения. Концентрируйтесь на общении, а не на красоте рисунка. Это должно помочь вам думать о раскадровке, как о совокупности пиктограмм, каждая из которых обозначает объект или действие. Ни что не мешает вам нарисовать пользователя в виде фигурки из черточек, а компьютер — в виде прямоугольника.

Если вы используете для раскадровки какой-нибудь графический редактор, типа «GIMP» или «Photoshop», то очень важно не поддаваться соблазну сделать ваш макет идеальным. Помните, дело не в том, чтобы получить красивую картинку, но в том, чтобы приблизить изображение интерфейса настолько хорошо, чтобы его хватило для дальнейшей работы над прототипом.

### 3.6. Инструменты для создания макетов и каркасных моделей

Для создания прототипов часто используют специализированное программное обеспечение. Лучшие приложения для макетирования и каркасного моделирования (wireframing) позволяют моделировать интерактивность и делать раскадровки. Существуют приложения, которые позволяют демонстрировать подготовленные макеты команде разработчиков или клиентам через общий доступ. В онлайн-журнале [SpecyBoo](#) вы можете найти целый список инструментов макетирования и каркасного моделирования. В принципе, для создания

### 3.7. Больше прототипов

Россон и Кэррол (Mary Beth Rosson, John M. Carroll. Usability Engineering: Scenario-Based Development of Human-Computer Interaction) предлагают следующий список различных способов прототипирования:

Таблица 2. Несколько разновидностей прототипов

<i>Тип</i>	<i>Описание</i>
Раскадровка	Наброски или снимки экрана, иллюстрирующие ключевые точки в описании использования
Картонный макет	Образец устройства с имитацией управления или экранных элементов. Выполняется не обязательно из картона.
«Волшебник Оз»	Рабочая станция, связанная с невидимым человеком-ассистентом, который симулирует ввод, вывод и исполняемую функциональность, которая еще не доступна
Видео-прототип	Видеозапись людей, разыгрывающих одну или несколько предполагаемых задач
Компьютерная	Переходы экранов, которые иллюстрируют последовательности

<i>Тип</i>	<i>Описание</i>
анимация	входных и выходных событий
«Машина сценариев»	Интерактивная система, реализующая определенный поток событий сценария
Быстрый прототип	Интерактивная система, созданная с помощью специальных инструментов прототипирования (в частности, с помощью визуальных средств разработки)
Частично работающая система	Исполнимая версия системы с ограниченной функциональностью

### Вопросы:

1. Какие из приведенных в таблице 2 прототипов низкой достоверности, а какие — высокой?
2. Какой из аспектов дизайна может проверить каждый из них?
3. Насколько труден каждый из них в изготовлении?
4. Какова выгода в стоимости для каждого прототипа?

## 4. Концептуальный проект

В промышленности концептуальный проект понимается как специфичный документ, отражающий интегрированные идеи, описывающие систему: ее функции, внешний вид и порядок взаимодействия с пользователем. Но дизайнеры часто вкладывают в понятие концептуального проекта несколько иной смысл, подразумевающий идею системы в целом.

Например в процессе создания web-страниц Концептуальное проектирование — начальный этап проектирования, при котором создается черновой вариант продукта, показывающий принцип закладываемой логики для первоначального согласования и проверки ограничений средств разработки. Самый простой концепт может быть описан словами — «Ваш сайт будет состоять из главной страницы с лучшими предложениями, на нем будут разделы магазина с поиском, новости, статьи, и раздел о магазине с вашими контактными данными». В процессе концептуального проектирования проектировщик из требований к системе и анализа действий пользователей создаёт:

- информационную архитектуру продукта (например, дерево сайта или последовательность экранов в интернет-магазине);
- расположение контента и функционала на экранах условными блоками.

После решения возникших вопросов с концептом продукта проектировщик переходит к детальному проектированию и создает детальный прототип.

Прис, Роджерс и Шарп в своей книге о проектировании взаимодействия (Yvonne Rogers, Helen Sharp, Jenny Preece. Interaction Design: Beyond Human — Computer Interaction) предполагают, что концепция может быть основана на трех перспективах:

- **Деятельность** — действия, которые пользователи выполняют чаще всего, в терминах четырех парадигм:
  - указание инструкций (как в большинстве программ: пользователь указывает системе, что она должна делать, задавая нужные команды через меню или прямым вводом);
  - общение с системой (диалоговый режим);
  - манипуляция и навигация по системе;
  - исследование и просмотр;
- **Объекты** — продукты интерфейса (отображаемые результаты) или объекты, используемые в интерфейсе.
- **Метафоры** — аналоги объектов или процессов реального мира.

Хороший концептуальный проект базируется на этих трех перспективах. Это можно проиллюстрировать примером. Один из наиболее востребованных видов офисных приложений — редакторы таблиц. Первый табличный редактор, VisiCalc, спроектировал Дэн Бриклин. До появления табличных процессоров, бухгалтеры вели учет, используя бухгалтерские книги и выполняя вычисления вручную. Бриклин понял, что бухгалтеры имеют проблемы с существующими инструментами, что компьютеры могут сделать процесс более производительным и интерактивным. Концептуальный проект будущего программного продукта был таким:

1. табличный процессор должен напоминать раскрытую бухгалтерскую книгу,
2. должна быть возможность редактировать содержимое ячеек таблиц,
3. должна быть возможность выполнять вычисления для диапазонов ячеек.

Его концептуальный дизайн учитывал деятельность (вычисления над данными, представленными в табличном виде), объекты (общие записи, ячейки, колонки) и имел хорошую метафору.

Другой пример — интерфейс «Звезда» (см. Xerox 8010 Information System) для офисных приложений, который был предшественником интерфейса типа «Рабочий стол».

Метафоры очень полезны в проектировании. Они связывают новый продукт с его предшественником. Они делают продукт более легким в освоении и использовании. Они могут помочь проектировщику в создании более согласованного дизайна интерфейса и выборе проектных альтернатив. Однако, метафора — прочная концепция и это может быть опасно. Пользователи могут верить в то, что новая система должна работать идентично аналогичной системе, с которой они уже знакомы и будут озадачены, если это окажется не так. Также проектировщики могут слишком придерживаться метафоры, что может быть причиной плохого дизайна.

Пример, иллюстрирующий и то, и другое: первые текстовые редакторы. В их основе — метафора с печатной машинкой: курсор, в виде символа подчеркивания или серого квадрата, отождествлялся с кареткой пишущей машинки. Пользователи, имевшие некоторый опыт работы с печатной

машинкой, могли довольно быстро освоить текстовый редактор. Однако, у них часто возникали сложности с впечатыванием символов в пустые области или поверх пробелов. Метафора с печатной машинкой подсказывала им, что нужно поместить курсор в нужное, визуально не занятое какими-либо символами, место, чтобы напечатать там символ. Это то, что сделала бы печатная машинка, и это называлось «перевести каретку». Пользователи не могли осознать, что символ пробела — это такой же символ, как и другие. Так дизайнеры слишком сильно придерживались аналогии. Сам курсор в виде знака подчеркивания («\_») еще больше усиливал непонимание. Позже курсоры стали вертикальными линиями и это сделало понятней различие между метафорой и приложением. В данном примере все могло быть еще хуже: дизайнеры могли бы перемещать страницу вслед за вводом текста. Понимание того, что страница должна оставаться неподвижной позволило упростить создание текстовых документов, но, заметьте, пишущая машинка функционирует иначе.