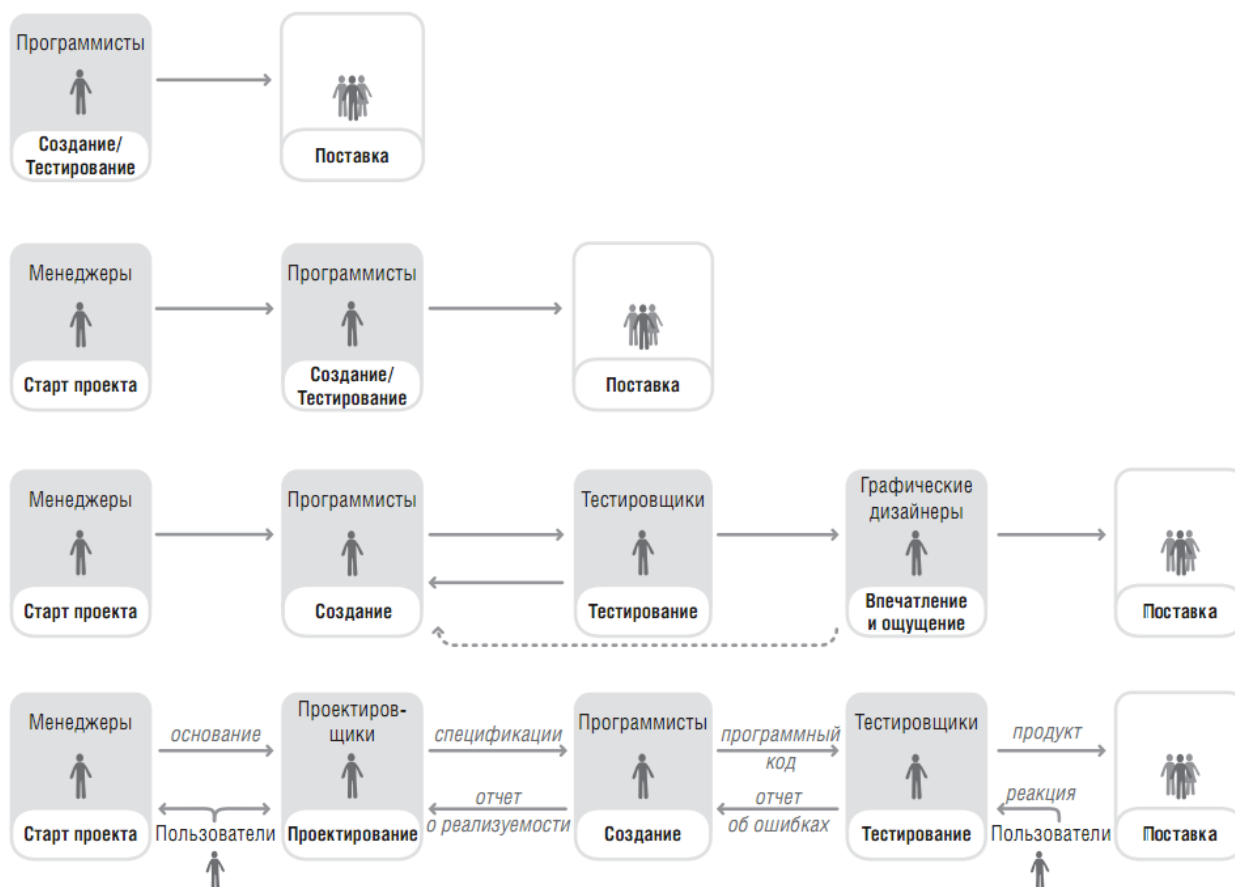


## Эволюция разработки программного обеспечения. Место проектирования человеко-машинных интерфейсов в процессе разработки.

### Целеориентированное проектирование.



На рисунке представлена эволюция процесса разработки программного обеспечения и указано то место, которое в этом процессе отводится (когда отводится вообще) проектированию.

1. Умные программисты вынашивали идею продукта, а затем создавали и самостоятельно тестировали его.
2. На каком-то этапе в процесс вошли профессиональные управленцы, которые помогали переводить благоприятные рыночные возможности на язык требований к продукту.
3. Индустрия повзрослела, и тестирование превратилось в самостоятельную дисциплину, а с распространением графических пользовательских интерфейсов (GUI) к процессу подключились графические дизайнеры, которые создавали визуальные элементы.
4. Решения о возможностях продукта, его форме и поведении принимаются до начала дорогостоящей и сложной фазы создания продукта.

Сложные строения в реальном мире требуют значительных предварительных усилий профессиональных архитекторов и проектировщиков. Необходимо понять, как живут и работают люди, использующие здание, чтобы спроектировать здание и пространство так,

чтобы им было удобно. В случае цифровых продуктов необходимо понять, как живут и работают пользователи продукта и спроектировать поведение и форму таким образом, чтобы им было удобно.

Проектирование взаимодействия касается не столько эстетических вопросов, сколько понимания пользователей и принципов их познавательной деятельности. При проектировании необходимо ориентироваться на реализацию целей и мотивов пользователей. Такое проектирование называется **целеориентированным**.

Разработка большинства цифровых продуктов застыла на первом, втором или третьем шаге этой эволюции, где проектирование либо не играет роли, либо становится косметической заплаткой поверх некачественного взаимодействия («макияжем для свиньи»). Процесс проектирования должен предшествовать созданию кода и тестированию, иначе нельзя гарантировать, что продукт действительно будет соответствовать потребностям пользователя.

## **Проектирование человеко-машинных интерфейсов. Введение в проблему.**

**Интерфейс** — совокупность возможностей, способов и методов взаимодействия двух систем, устройств или программ для обмена информацией между ними.

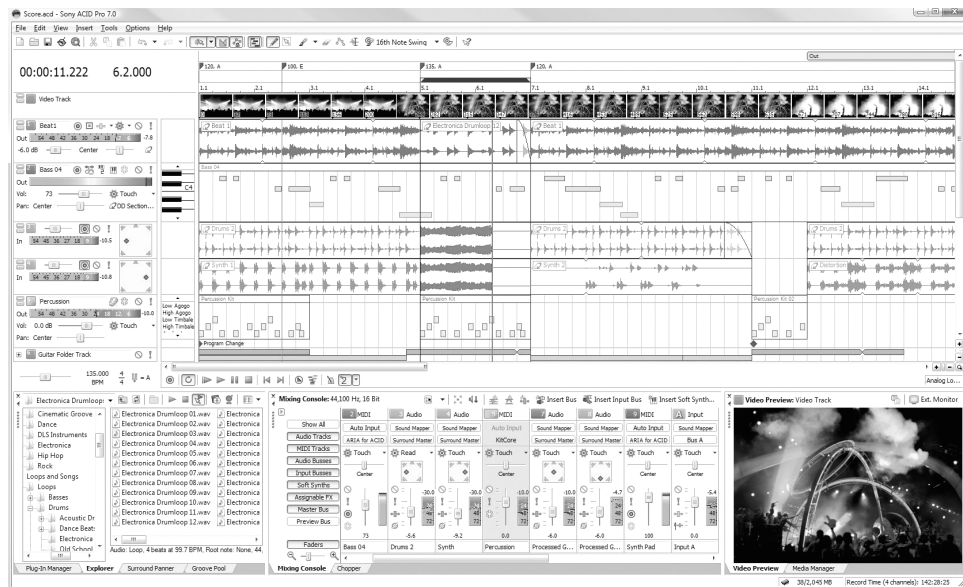
Интерфейсом обладают фактически все функциональные вещи, которые нас окружают. Пример: интерфейс кроссовок и тапочек. Можно сказать, что интерфейс кроссовок сложнее, чем интерфейс тапочек, однако понятно как пользоваться и тем, и другим.

Есть двери без ручек, краны без вентилей. Иногда не сразу можно додуматься, как этим пользоваться. Тогда идет речь о непонятных интерфейсах (подробнее в книге Дональда Нормана «Дизайн привычных вещей»).

**Интерфейс пользователя** — разновидность интерфейсов, в котором одна сторона представлена человеком (пользователем), другая — машиной/устройством.

Когда мы переходим в среду машина-человек, возникают проблемы понимания интерфейса. Причины:

1. Историческое развитие области. Сначала для общения с компьютером появились перфокарты, такой способ общения очень сложно назвать человеческим. Дальше начали развиваться языки программирования и интерфейсы следовали этому развитию, то есть по инерции внешний вид был продиктован какими-то сущностями из базы данных, архитектурой кода и другими вещами без учета того, как эта функциональность будет использоваться.
2. В какой-то момент, когда программные продукты стали продаваться, подключилась гонка функциональности (чем больше функций содержал продукт, тем по определению он был лучше продуктов-конкурентов). Это очень сильно повлияло на интерфейс — все функции, которые были, вываливали в интерфейс. Из-за этого получались вот такие монстры:



3. В искусственном интерфейсе нет моментальной обратной связи. Пример с открыванием двери: тыходишь к двери, толкаешь её, она не открывается, ты тянешь её на себя, она открывается. Таким образом действие выполнено (несмотря на ошибку). В случае с программами такой обратной связи по определению нет. Её нужно программировать. Не всегда понятно, как ошибка произошла, какие процессы в системе её повлекли.

Сначала на проблемы понимания интерфейса никто не обращал внимания, потому что компьютер был только у определенного малого количества людей (программистов, математиков), но далее, по мере распространения компьютеров, проблемы стали очевидны. Постепенно фокус начал смещаться в сторону интересов пользователей.

**Проектирование** — сознательные и интуитивные усилия по созданию значимого порядка. Это значит:

1. Понимание желаний, потребностей, мотивации пользователей и контекста, в котором эти пользователи находятся;
2. Понимание возможностей, требований и ограничений бизнеса, технологии и предметной области;
3. Использование этих знаний в качестве основы всех планов по созданию программных продуктов, форма, содержание и поведение которых делают их полезными, удобными и желанными, а так же экономически жизнеспособными и технически осуществимыми.

Чтобы не сводить с ума пользователей нужно (важнейшие интересы пользователей):

1. **Ставить на первое место материальные ценности** (пример — одни банкоматы возвращают карту сразу после выполнения операции, другие спрашивают, хочет ли пользователь выполнить еще одну операцию и после нажатия кнопки «нет» возвращают карту. Первый способ лучше, потому что цена ошибки в случае забытой карты выше, чем цена времени, потраченного на повторное действие по вводу пин-кода);

2. **Сохранять введенные данные** (пример — потеря нескольких часов работы после отключения компьютера из-за ошибки);
3. **Экономить время** (шорткаты, не нужно заставлять пользователя каждый раз вбивать информацию, которую можно было запомнить, нужно предугадывать действия пользователя).

За последнее время качество программ и интерактивных продуктов улучшилось. Многие компании начали уделять пристальное внимание тому, чтобы их продукция удовлетворяла нужды людей, и стали тратить время и деньги на раннее проектирование. Однако гораздо большему количеству компаний все еще не удалось включиться в эту игру, и до тех пор, пока они сосредоточены на технологии и рыночных данных, они будут продолжать создавать цифровые продукты, вызывающие негативную реакцию.

### **Основные положения современного состояния проектирования пользовательских интерфейсов.**

1. **Отсутствие представления о пользователях.** Мы можем знать, в каком сегменте рынка находятся пользователи, сколько денег они зарабатывают, какие автомобили приобретают. Но это не позволяет узнать, как осчастливить пользователей, почему они занимаются какой-либо деятельностью, почему они могут отдать предпочтение нашему продукту, а не продукту конкурента, как этого добиться.
2. **Конфликт интересов.** Проектируют продукты те же люди, которые эти продукты разрабатывают — программисты. Программистам часто приходится выбирать между простотой создания кода и простотой использования продукта. Поскольку о производительности программистов судят по их способности эффективно писать код и сдавать его в невероятно сжатые сроки, несложно понять, в какую сторону склоняется чаша весов.
3. **Отсутствие процесса.** Нет полноценного процесса для преобразования представления о пользователях в продукты, одновременно удовлетворяющие потребности пользователей и вызывающие живой отклик.

Цифровые продукты нередко рождаются в результате перетягивания каната двумя враждующими силами — разработчиками и маркетологам.

Маркетологи хорошо замечают и оценивают шансы продукта по выходу на рынок, однако зачастую их вклад в проектирование заключается в передаче разработчикам списка требований, которые далеки от нужд и целей пользователей. Вместо тонкой гармонии получается список из сотни требований, которые не делают технологию полезной.

Участие разработчиков ничем не ограничено. Поскольку за строительство отвечают они, они и решают, что строить. Их набор представлений также отличается от набора представлений пользователей. Хорошие разработчики концентрируются на том, чтобы найти решение сложных технических проблем, применить подходящие инженерные методы и сдать продукт вовремя.

## Модели реализации и ментальные модели.

В компьютерной отрасли часто используют термин **компьютерная грамотность** — фраза, с помощью которой пользователей заставляют подстраиваться под машинную логику, вместо того, чтобы приблизить программные продукты к образу мышления пользователей.



Чтобы пользоваться сложным устройством, человеку не нужно знать в деталях, как оно работает на самом деле. Поэтому человек создает упрощенную мысленную схему, достаточно мощную для того, чтобы взаимодействовать с устройством. Таким образом формируется **ментальная модель**.

**Модель реализации** — то, как в действительности работает машина или программа.

**Модель представления** — избранный проектировщиком способ предъявления пользователю функционирования программы.

Возможность демонстрировать функционирование компьютера в отрыве от производимых им в реальности действий проявляется в программном обеспечении сильнее, чем в любой другой среде. Сообразительный проектировщик скрывает некоторые из наиболее малопривлекательных особенностей того, как программа выполняет свою работу.

Чем ближе к пользовательской ментальной модели окажется модель представления, тем легче пользователю будет работать с программой и понимать её. Таким образом, проектировщикам очень важно детально знать, как целевые пользователи видят работу, выполняемую с помощью программы.

Большинство программных продуктов следуют модели реализации, потому что их проектируют программисты. С точки зрения инженера модель реализации логична и понятна, но к сожалению она малопонятна пользователям. Математическое мышление приводит к созданию интерфейсов, следующих модели реализации. Обычные пользователи не понимают булеву алгебру (пример — «и» и «или» трактуется прямо противоположным образом, слова «найди людей, живущих в Белгороде и Москве» не имеет смысла для программиста, т.к. это будет пустым множеством).

## **Цели пользователей.**

Если не принимать во внимание цели пользователей, на выходе мы получим продукцию, лишаящую пользователей целостного опыта. Продукт будет раздражать пользователя, снижать его производительность и не будет отвечать его потребностям.

Если проектирование осуществляется подходящими методами, оно способно восстановить отсутствующую связь человека с технологическими продуктами. На это направлено целеориентированное проектирование. Чтобы понять суть целеориентированного проектирования, нужно понять цели пользователей и осознать их ключевую роль в проектировании соответствующего интерактивного поведения.

Функциональность и поведение продукта должны быть ориентированы на достижение целей пользователей через решение задач. Задачи — средство движения вперед, цель — действительный пункт назначения.

### **Три типа пользовательских целей:**

#### **1. Эмоциональные цели (как пользователь хочет себя чувствовать)**

Примеры эмоциональных целей:

- Чувствовать уверенность в том, что ситуация под контролем
- Получать удовольствие
- Ощущать душевный подъем или расслабленность
- Быть собранным и сосредоточенным

Когда продукт заставляет пользователей чувствовать себя глупо или неудобно, их производительность труда снижается, а недовольство растет.

#### **2. Конечные цели (что пользователь хочет делать)**

Примеры конечных целей:

- Узнавать о проблемах до того, как они послужат причиной катастрофы
- Поддерживать контакт с друзьями и близкими
- Найти музыку, которая мне понравится
- Получить наилучшее предложение

Этот вид целей является основой при проработке поведения, задач, вида продукта, а также создаваемых им впечатлений.

#### **3. Жизненные цели (кем пользователь хочет быть)**

Примеры жизненных целей:

- Прожить хорошую жизнь
- Преуспеть в реализации амбиций
- Стать знатоком в определенной области
- Быть привлекательным, популярным, завоевать уважение коллег

Это личные стремления пользователя, которые обычно выходят за пределы контекста работы с программным продуктом, помогают объяснить, почему пользователь пытается достичь конечных целей. Описывают долгосрочные желания и мотивы персонажа, которые служат маяком для проектирования продукта в целом, создании стратегии и брендинга для него. Продукт, приближающий пользователя к достижению жизненных целей завоюет пользователя более решительно. Ориентация на жизненные цели пользователей и составляет разницу между пользователем удовлетворенным и пользователем фанатично преданным.

## **Прочие типы целей**

Пользовательские цели — не единственная разновидность целей, которую проектировщики должны принимать во внимание. Цели покупателей, бизнес-цели и технические цели следует также учитывать и рассматривать, однако они не влияют на основное направление проектирования. На эти цели нужно ориентироваться, но их нельзя удовлетворять за счет пользователя.

### **1. Цели покупателей.**

Рядовые покупатели — родители, родственники или друзья, которые озабочены личной безопасностью и настроением человека, для которого приобретается продукт.

Еще один пример покупателей — менеджеры по информационным технологиям, озабоченные информационной безопасностью, простотой обслуживания и настройки.

Они могут иметь свои цели, которые нужно учитывать.

### **2. Бизнес-цели и цели организации**

Примеры:

- Увеличить прибыль
- Расширить свое присутствие на рынке
- Удержать клиентов
- Обойти конкурентов
- Эффективнее использовать ресурсы
- Расширить спектр предлагаемых продуктов или услуг

При проектировании для некоммерческой организации (например, музея) будут другие цели:

- Повысить уровень образованности публики
- Получить достаточно денег, чтобы покрыть расходы

Организации предъявляют собственные требования к продуктам, услугам и системам. Часто бывает полезно определить бизнес-цели организации, создающей и продающей (или иным способом распространяющей) проектируемый продукт. Очевидно, что организация надеется достичь чего-то, создавая продукт, именно поэтому тратит ресурсы на проектирование и разработку.

### **3. Технические цели:**

Примеры:

- Поддерживать различные браузеры
- Сохранять целостность данных
- Повысить эффективность работы программы
- Использовать какой-то конкретный язык разработки или библиотеку
- Сохранить межплатформенность

Большинство программных продуктов, которыми мы пользуемся, создавались с учетом определенных технических целей. Многие из таких целей состоят в том, чтобы облегчить создание программ, то есть являются целями программистов. Именно поэтому они часто удовлетворяются в ущерб целям пользователей.

Пользователям все равно, какая БД используется — реляционная или нет. Более важно быстрое эффективное выполнение работы, простота работы с программой, сохранение чувство собственного достоинства.

**Успешные продукты в первую очередь служат целям пользователей.**

**Не нужно заставлять пользователей чувствовать себя дураками.**

**При проектировании необходимо считать, что пользователи умные, но сильно занятые люди.**