***Лекция 1 ПрЧМИ***

***Что такое пользовательский интерфейс***

***Интерфейс*** - система правил и средств, регламентирующая и обеспечивающая взаимодействие нескольких процессов или объектов.

***Пользовательский интерфейс (ПИ)*** - система правил и средств, регламентирующая и обеспечивающая взаимодействие программы с пользователем.

Далее под программой, программным обеспечением или программным продуктом будет также подразумеваться и Интернет-сайт.

ПИ объединяет в себе все элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением (ПО).

К элементам относятся:

* набор задач пользователя, которые он решает при помощи системы;
* используемая системой метафора (например, рабочий стол в MS Windows); **Мета́фора** — слово или выражение, употребляемое в переносном значении, в основе которого лежит сравнение неназванного предмета или явления с каким-либо другим на основании их общего признака. В основе метафоры лежит сравнение предмета или явления с другим предметом и явлением на основании их общего признака. Слово, которое подменяет метафора, при этом не называется напрямую.
* элементы управления системой;
* навигация между блоками системы( Как создать элемент управления, управляемых данными, для перехода к определенному продукту данных.);
* визуальный (и не только) дизайн экранов программы;
* средства отображения информации, отображаемая информация и форматы;
* устройства и технологии ввода данных;
* диалоги, взаимодействие и транзакции между пользователем и компьютером;
* обратная связь с пользователем;
* поддержка принятия решений в конкретной предметной области;
* порядок использования программы и документация на нее.

***Ситуация на мировом рынке ПО***

Развитие информационных систем показывает, что конкуренция продуктов из области функциональности переходит в область их удобства и комфортности для пользователей. Перед разработчиками встает проблема проектирования ПИ, позволяющего обеспечить эффективное и экономичное использование ПО. В этих условиях, эргономические методы проектирования становятся технологиями, обеспечивающими рыночный успех проекту.

Проведенные исследования показывают, что:

* ПИ составляет от 47 до 60 процентов кода программы;
* на разработку ПИ уходит как минимум 29 процентов проектного бюджета и в среднем 40 процентов всех усилий разработчиков по созданию системы.

**Ситуация на российском рынке ПО**

Эргономические методы проектирования ПИ остаются в тени. Разработчики с неохотой тратят деньги на грамотное проектирование ПИ и последующее юзабилити-тестирование ПП.

У развития эргономики ПО в России есть несколько предпосылок:

* Конкуренция обостряется, что заставляет всё время повышать качество продукта.
* Наступает фаза зрелости отечественных программных продуктов (ПП). Любой ПП в момент своего появления обладает сравнительно небольшой функциональностью и сравнительно простым интерфейсом. Со временем функционально возрастает, а интерфейс соответственно усложняется. Рано или поздно продукт взрослеет, на этом этапе добавление новой или изменение существующей функциональности требует параллельных шагов по оптимизации интерфейса, без которой эффект от этой функциональности не может быть получен.
* Рост количества пользователей, не имеющих навыков работы с компьютером. Количество людей, не склонных разбираться с тонкостями настройки ПО многократно возросло и составляет на настоящий момент очень большой процент от общего числа пользователей. Тем не менее, использование компьютера в повседневной работе является для них необходимостью. Именно поэтому важным критерием при выборе ПП становится простота и удобство интерфейса.

**Преимущества хорошего ПИ**

Системы, разработанные с учетом требований удобства пользователя, эргономичны.

***Существует четыре основных критерия эргономичности (качества) любого интерфейса, а именно:***

***Скорость работы пользователей***

***Количество человеческих ошибок***

***Скорость обучения***

***Субъективная удовлетворенность пользователей***

Системы работают именно так, как пользователи ожидают, и позволяют пользователю сосредоточиться на собственно задачах, а не на особенностях взаимодействия с системой. Эффективный интерфейс является результатом того, что разработчик уделяет внимание не только данным, с которыми работает пользователь, но и собственно пользователю, его задачам и деятельности.

Сами по себе ориентированные на пользователей методы проектирования ПИ обладают определёнными преимуществами. Идентификация и устранение ошибок на ранних этапах проектирования системы ведет к ее значительному удешевлению. Бумажное макетирование ПИ совместно с конечными пользователями, ведет к установлению более полного понимания между разработчиком и заказчиком ПО. Это снижает вероятность последующих переделок. Полное и четкое определение задач и договоренность относительно принципов построения ПИ ведет к более адекватной оценки задачи как заказчиком, так и исполнителем. Это позволяет заказчику убедиться в том, что исполнитель действительно заботится об его потребностях.

Существенные преимущества хорошего пользовательского интерфейса:

* ***Повышение конкурентоспособности***.   
  Разработчики, уделяющие внимание интерфейсу своего продукта могут оставить далеко позади своих конкурентов, сделав свой продукт простым и удобным в использовании. Есть примеры, когда разработка ПП, проведенная с соблюдением требований юзабилити, увеличивала доходы производителя на 80%. При этом пользователи всегда оценивают простоту использования ПП, как самую важную характеристику.
* ***Снижение стоимости разработки.***   
  Реальная себестоимость ПП, как правило, значительно выше стоимости их разработки. Себестоимость возрастает за счет внедрения и поддержки продукта, причем она может возрастать на 80% от стоимости разработки. Это объясняется непониманием программистами целей и ожиданий конечных пользователей продукта, причем это не понимание обнаруживается сразу после сдачи продукта в эксплуатацию. За счет локализации проблем пользовательского интерфейса на ранних этапах разработки, можно почти всегда снизить затраты на 60-90%.
* ***Увеличение аудитории продукта.***   
  Интернет переполнен сайтами, которые, не учитывают потребностей посетителей, отталкивают потенциальную аудиторию. Такие сайты сложны в использовании, на них трудно найти нужную информацию, они не доносят до посетителя послание владельца. Используя ориентированный на цели пользователей подход, можно быть уверенным в том, что ресурс будет отвечать нуждам посетителей, что поможет делать успешный бизнес.  
  Армия пользователей постоянно пополняется людьми, не готовыми к использованию компьютера. Например, люди предпенсионного возраста до внедрения компьютерной техники успешно решали свои профессиональные задачи. С появлением компьютера на рабочих местах эти люди переживают огромный эмоциональный стресс. До внедрения компьютера они отчетливо представляли себе всю картину делопроизводства, но теперь они такую способность утратили. Более того, из-за возрастного снижения способности к обучению, они зачастую не могут обучиться пользоваться даже очень простыми системами.  
  В то же время, проведение пользовательского тестирования и этнографического анализа будущей аудитории системы, позволяет значительно снизить требования к навыкам работы на компьютере. За счет этого удается сохранить в производственном процессе большое количество трудоспособного (и опытного) населения.
* ***Уменьшение затрат на обучение и поддержку пользователей.***   
  Использование юзабилити методов при проектировании ПО значительно снижает время, необходимое для обучения пользователей, равно как и ресурсы технической поддержки. В среднем использование таких методов при проектировании продукта снижает время обучения на 25%, а количество обращений в службу технической поддержки – на 60%.
* ***Уменьшение потерь продуктивности работников при внедрении системы и более быстрое восстановление утраченной продуктивности.***   
  Часто после внедрения новой системы на предприятие производительность работников значительно падала. Бывали случаи, когда работа и вовсе замирала. Чем удобнее и проще интерфейс, тем легче происходит обучение и привыкание к новой системе, и соответственно, быстрее эта система начинает окупаться и приносить прибыль.
* ***Доступность функциональности системы для максимального количества пользователей.***   
  Поскольку с точки зрения пользователя ПИ является ключевым фактором для понимания функциональности программы, плохо разработанный интерфейс резко ограничивает функциональность системы в целом.
* ***Снижение риска катастроф.***По отчетам различных правительственных и независимых комиссий по расследованию катастроф, одной из наиболее распространенных причин аварий является т.н. «человеческий фактор». Под этим термином понимается ситуация, при которой человек в некоторой ситуации принял неправильное или не принял правильного решения. На самом деле, значительной части аварийных ситуаций, причиной которых стал человеческий фактор, можно было бы избежать, если бы интерфейс систем, вызвавших катастрофу, препятствовал операторам совершать действия неправильные и помогал бы совершать действия правильные. Учитывая количество человеческих жизней, которые с удручающей регулярностью уносят катастрофы, создание таких интерфейсов является нравственным долгом разработчиков.

**История развития интерфейсов**

В дизайне пользовательского интерфейса можно условно выделить ***декоративную*** и ***активную*** составляющие. К первой относятся элементы, отвечающие за ***эстетическую привлекательность*** программного изделия. Активные элементы подразделяются на ***операционные*** и ***информационные*** образы моделей вычислений и ***управляющие средства*** пользовательского интерфейса, посредством которых пользователь управляет программой. Управляющие средства различных классов программных изделий могут значительно различаться.

Известен устойчивый и предсказуемый прогресс в аппаратном обеспечении, выражаемый в постоянном улучшении соотношения цена / производительность. Ничего подобного в отношении пользовательских интерфейсов не наблюдается. Пожалуй, вся история развития интерфейсов лучше всего характеризуется понятием - "прерывистое равновесие", когда долгие периоды стабильности прерываются быстрыми изменениями. Можно идентифицировать четыре качественно отличных друг от друга поколения, которые характеризуются четырьмя интерфейсными стилями.

**Первое поколение ПИ**

В первый период (50-е и начало 60-х гг.) компьютеры, работали в основном в пакетном режиме, используя перфокарты для ввода и устройство построчной печати для вывода. При этом фактически не было смысла говорить о пользовательском интерфейсе - не существовало самого понятия "интерактивного пользователя" в современном смысле этого слова (хотя некоторые специалисты умудрялись выполнять отладку прямо с консоли, используя переключатели и световые индикаторы как "пользовательский интерфейс").

**Второе поколение ПИ**

Во втором периоде в эволюции интерфейсов (с начала 60-х до начала 80-х гг.) господствовал режим разделения времени на мэйнфреймах и мини-компьютерах с использованием алфавитно-числовых дисплеев, пользователи могли взаимодействовать с компьютером путем ввода с клавиатуры команд с параметрами. Этот тип взаимодействия захватил и век ПК с MS DOS и ОС Unix.

**Третье поколение ПИ**

Третье поколение пользовательских интерфейсов взяло старт еще в 70-е г. - при режиме разделения времени и ручного ввода команд. В научно-исследовательском центре Xerox PARC были созданы графические интерфейсы пользователя (GUI), предназначенные для работы на растровых графических сетевых рабочих станциях. Эти интерфейсы принято обозначать аббревиатурой WIMP (Windows-Icons-Menus-Pointing device), что отражает задействованные интерактивные сущности - окна, пиктограммы, меню и позиционирующее устройство (обычно мышь). Именно интерфейсы этого типа, завоевавшие популярность вместе с Macintosh в 1984 году и позднее скопированные, в частности, в Windows для ПК, доминируют и по сей день. Сегодняшние приложения имеют интерфейсы того же типа, как и ранние "настольные" приложения, разве что увеличилась степень "реализма" благодаря применению современных интерфейсных инструментов, позволяющих, например, использовать тени для экранных кнопок.

Третье поколение WIMP GUI доминирует очень долго (более двух десятилетий!); видимо, интерфейсы этого типа полностью соответствуют требованиям значительной части современных настольных приложений. Без преувеличения можно сказать, что метод "указания и щелчка", отличительная особенность WIMP-интерфейсов, уже стал частью современной культуры.

По существу, WIMP GUI стали стандартом для прикладных интерфейсов

**Недостатки WIMP-интерфейсов**

***Во-первых, чем более сложным является приложение, тем труднее осваивать интерфейс, причем эти трудности возрастают нелинейно.*** Взятые в отдельности интерфейсные особенности и инструменты могут быть вполне простыми, но, будучи в большом количестве интегрированы в одно приложение, они образуют новое качество сложности. Многие современные настольные приложения столь объемны, чтобы не сказать громоздки, что пользователь, погрузившись однажды в их функциональность, начинает даже отказываться от новейших версий, продолжая использовать то малое подмножество возможностей, которое удалось изучить, - это выражено в ставшем классическом "правиле 90/10".

***Во-вторых, пользователи проводят слишком много времени, манипулируя интерфейсом, а не работая с самим приложением.*** Квалифицированные пользователи часто бывают раздражены слишком большим количеством интерфейсных элементов (использование сокращенных комбинаций клавиш - это суррогатный метод решения этой проблемы).

***В-третьих, WIMP GUI вместе с их 2D- интерфейсными элементами проектировались для работы с двухмерными же приложениями - такими, как обработка текстов, компоновка документов и электронные таблицы.*** Если же приложение является по своей сути трехмерным, то работа с ним с помощью стандартных 2D компонентов становится не слишком естественной. 3D-приложения, как правило, имеют большую визуальную сложность, чем двухмерные, что еще более усиливает связанные с WIMP-интерфейсами проблемы.

В-четвертых, не все пользователи способны эффективно использовать мышь и клавиатуру - либо оттого, что им это не кажется естественным в контексте их задачи, либо из-за вызываемых этими устройствами чисто физиологических неудобств, связанных с постоянными нажатиями на клавиши при сильном напряжении зрения (не говоря уже о специальных категориях пользователей с физическими недостатками).

**Главным недостатком WIMP-интерфейсов является** то, что они никак не используют такие каналы взаимодействия, как:

* речь,
* слух
* прикосновения.

Еще одно ограничение WIMP-интерфейсов в том, что они предназначены для одинокого пользователя настольной системы, который управляет объектами, не обладающими автономным поведением, реагирующими в основном на манипуляции с мышью. Соответственно имеется один, не разделяемый во времени полудуплексный канал взаимодействия; система откликается на каждое дискретное событие ввода, и эти события могут быть легко распознаны - они состоят из простых нажатий на клавиши и выбора с помощью мыши. Самый сложный ввод - последовательность позиций мыши, которая может представлять, например, путь закрашивающей кисти.

**Четвертое поколение ПИ**

Сегодня на повестке дня новые формы взаимодействия с компьютером, как в концептуальном, так и в программно-аппаратном плане, настойчиво требующие интерфейсов нового - четвертого поколения, которые пока получили название post-WIMP-интерфейсы. Они не используют меню, формы и панели инструментов; вместо них при задании спецификаций операций и операндов упор идет на:

* обучающие примеры (examples),
* жесты
* распознавание речи.

***Цель создания интерфейсов нового поколения***:

* минимизировать использование тех механизмов манипулирования, которые столь существенны в сегодняшних интерфейсах,
* сократить "когнитивную" дистанцию между намерением и реализацией этого намерения.

Примеры взаимодействия с помощью Post-WIMP-интерфейсов:

* ***распознаватели жестов***, основанные на технике рисования пером, - они используются в некоторых моделях карманных компьютеров. Эти устройства более или менее успешно сочетают методы, свойственные как WIMP-, так и post-WIMP интерфейсам для 2D-задач.
* Другим показательным примером естественного человеко-машинного взаимодействия, но не использующим какие-либо WIMP-устройства и методы, являются ***диалоговые видеоигры, такие как тренажеры с рулевым колесом*** управления с переключателем передачи, а также имитаторы игр, вроде гольфа, в которых игрок может бить реальной клюшкой по реальному мячу, траектория полета которого затем моделируется и изображается на экране дисплея.
* ***Объекты, инкапсулирующие 3D-геометрию и предназначенные для управления другими объектами*** в сцене начинают с успехом использоваться в задачах трехмерного моделирования; при этом они являются частью 3D-сцены, что позволяет не прибегать к привычным двухмерным компонентам, обычно накладываемым на 3D-сцену.
* В качестве примера сошлемся на 3D-компонент "рычаг", которые используются для манипулирования в задачах моделирования 3-х мерных тел.

Еще один метод – это ***ввод с помощью двух рук***, при котором "вспомогательная" рука управляет крупными (грубыми) движениями (например, перемещением инструмента), в то время как "доминирующая" рука выполняет тонкую настройку (например, манипуляции инструментом).

***"Осязательные" пользовательские интерфейсы***

Это еще одна мало исследованная область: основанная на осязательных (тактильных) ощущениях аппаратура появилась совсем недавно. Тактильные устройства, в отличие от других интерактивных устройств, способны как "чувствовать", так и передавать информацию. Таким образом, дизайнеры тактильных интерфейсов рассматривают две равно важные стороны: тактильные ощущения (чувство касания) и "кинестетическое" (kinesthetic) чувство (ощущение, где находится тело). Эти устройства имеют общую особенность: они снабжены средством силовой обратной связи - таким, как PHANToM (от фирмы SensAble Devices - [*http://www.sensable.com*](http://www.sensable.com/)), которое получает информацию о положении и жесте, а возвращает величину приложенной в точке силы. Таким образом, пользователь может ощущать форму жесткого объекта, в том числе через несколько слоев различного сопротивления при надавливании на внешнюю поверхность (что полезно, например, в хирургических симуляторах).

***Распознавание речи*** так же может использоваться для подачи команд и вообще для неограниченного текстового ввода, но соответствующая технология еще недостаточно зрела, чтобы действительно широко применяться. И привлекательность, и трудность реализации распознавания непрерывной жестикуляции и речевого ввода определяются тем, что при этом очень нелегко производить разбивку на значимые лексемы и однозначно выделять такие компоненты, как глагол, существительное и модификаторы. Боб Железник (Bob Zeleznik) со своими коллегами в Brown University недавно продемонстрировал 3D-интерфейс моделирования, который полностью полагается на ввод геометрических эскизов и команд вручную и имеет словарь, включающий более 20 жестов [9]; они уже расширили этот тип интерфейса, с тем чтобы включить взаимодействие "с двух рук" и тем самым обеспечить хорошую комбинацию методов [8]. Эта система, Sketch [9], использовалась для создания трехмерной модели бейсбольного поля

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Класс** | **Подкласс** | **Типы управляющих средств** |
| 1 | Пакетный |  | Перфокарты |
| Перфоленты |
| 2 | Символьный | Командный интерфейс | Вопрос-ответ |
| Командная строка |
| …. |  |  |  |
| 3 | Графический | Простой графический | Экранные формы |
| Управляющие клавиши |
| Истинно графический, двухмерный | Меню |
| Графические элементы управления |
| Прямое манипулирование |
| Трехмерный |  |
| 4 | Пост. графический | Речевой |  |
| биометрический (мимический) |  |