

## **Содержание:**

# **Введение**

Развитие современных технологий промышленности, транспорта и обороны сопровождаются ростом уровня автоматизации и развития компьютерной компоненты оборудования рабочих мест, при одновременном возрастании функциональной нагрузки на восприятие информации операторами. Намеченные приоритеты отмечена во всех сферах операторской деятельности современного человечества, управление подводной лодкой или пилотирование современного самолета. Автоматизированное проектирование современных рабочих мест требует от специалистов освоения компьютерных систем, выступающих в качестве инструмента. Взаимодействие с программным интерфейсом стало обязательной частью работы операторов автоматизированных систем управления.

Важно отметить значение корректно спроектированного интерфейса в эффективном взаимодействии человека с программным интерфейсом. Неверно спроектированные интерфейсы являются источниками стресса и психологического неудобства, возникающего в результате некорректного распределения функций между человеком и программным интерфейсом, в результате навязывания неудобных алгоритмов или темпов выполнения функциональной деятельности. Не выполняется учет психофизиологических возможностей операторов ПК при решении конкретных задач, нет адекватного отображения процессов взаимодействия между оператором и ПК.

Повышение эффективности и безопасности функционирования системы оператор-ПК требуют всестороннего учёта факторов взаимодействия оператора с ПК и свойств возникающей при этом среды восприятия, в которой проявляются эффекты интеллектуального и социального поведения человека и интеллектуальных компонентов, порождаемых технической системой.

## **2. Интерфейс программы**

В современном компьютерном мире наиболее часто используют программный и пользовательский виды интерфейса, и при этом имеющиеся стандарты указывают

на то, какими эти интерфейсы быть не должны, т. е. весьма незначительно уменьшают множество возможных вариантов построения интерфейса.

Исследования в области эволюции разработки интерфейса программ показывают, что любой ИП должен обеспечивать выполнение следующих четырёх основных функций:

1. Управления компьютером путём действий пользователя: инициация, прерывание, отмена компьютерных процессов и т. п.
2. Ввод данных, осуществляемый оператором, и отклик системы.
3. Отображение данных, включающее отображение данных, вводимых оператором, который может управлять процессом отображения данных.
4. Поддержка оператора в процессе деятельности, осуществляемая по каналам обратной связи, в которых циркулирует информация об ошибочных или случайных (не по алгоритму) действиях оператора.

Эффективный ИП должен обеспечивать всестороннее использование потенциальных возможностей оператора, технических и информационно-программных средств автоматизированных рабочих мест, высокие безошибочность и быстродействие деятельности оператора при применении ИП по назначению. Хорошо спроектированный ИП обеспечивает максимальный комфорт деятельности оператора и должен:

- способствовать быстрому освоению средств вычислительной техники оператором, формированию у него устойчивых навыков владения ПК;
- быть разработан таким образом, чтобы оператор вводил информацию естественным образом, не заботясь о ходе вычислительного процесса;
- удовлетворять рабочие потребности человека-оператора, а не обслуживать процесс обработки данных.
- Синтаксическая структура, реализованная в интерфейсе должна:
- соответствовать ожиданиям оператора решающего профессиональную задачу;
- содержать систему правил работы оператора, обеспечивающую прозрачное управление системой;
- постоянно находиться под контролем оператора ПК, никакие действия которого не должны приводить к зависанию программы.[1]

Интерфейс и справочный механизм информационной системы должны:

- обеспечивать возможность лёгкого исправления ошибок ввода, не должен требовать повторного ввода данных;

- обеспечивать оператора информацией, позволяющей ему управлять диалогом, распознавать и исправлять ошибки, определять последующие действия, входящие в алгоритм;
- обеспечить конкретность и понятность оператору выдаваемой компьютером информации;
- согласование объёма представляемой оператору информации с возможностями его кратковременной памяти;
- в информации об ошибках оператора делать акцент не на неправильные действия последнего, а на то, чем и как можно исправить ошибки;
- предусматривать использование четырех видов диалога: меню, команды, манипуляции и диалог посредством заполнения форм.
- при решении каждой задачи предоставить оператору возможность использования не менее двух видов диалога. Критерием выбора вида диалога в ходе решения конкретной задачи является обеспечение заданных показателей безошибочности и быстродействия.

### **3. Структура интерфейса программ**

Пользовательский интерфейс состоит из трёх основных частей:

1. Внешнее оформление, которое отвечает за представление информации оператору.
2. Функциональные возможности системы, включающие набор возможностей для эффективного выполнения его профессиональной и трудовой деятельности.
3. Техники взаимодействия оператора с программным интерфейсом.[\[1\]](#)

Однако нередко разработчики ПО (программного обеспечения) рассматривают функциональность системы отдельно от её интерфейса программ, и редко рассматривают элементы взаимодействия пользователя и системы посредством интерфейса пользователя. При этом предполагается, что ИП является своего рода дополнением к функциям системы. Со своей стороны, пользователи программ, как правило, не разделяют функциональность и пользовательский интерфейс. Для пользователей именно ИП является вычислительной программой. Впечатление от взаимодействия с программным обеспечением формируется непосредственно от работы с интерфейсом.

Поэтапная разработка интерфейса программ позволяет повысить эффективность программного обеспечения, уменьшить время освоения ПО пользователями,

снизить стоимость доработки системы после её внедрения, а также полностью использовать заложенную в программный продукт функциональность.[2]

## 4. Стили интерфейса программ

Существует ряд стилей интерфейса программ, которые нашли свое место в проектировании программных средств. Основные виды стилей ИП представлены на рисунке 1.

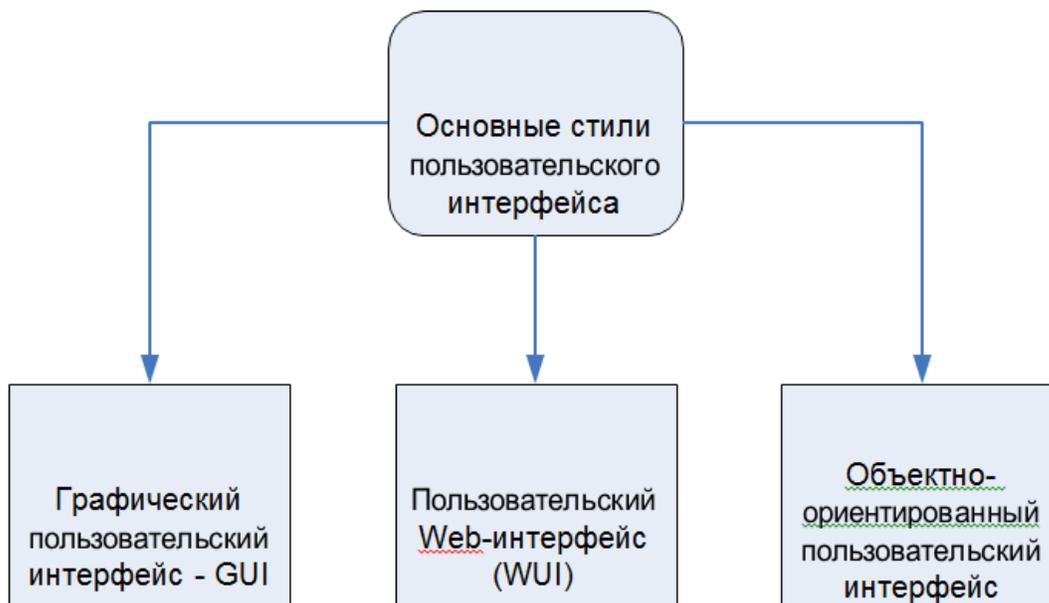


Рис.1. Основные стили интерфейса программ

Наиболее известными являются GUI-интерфейсы (GUI – Graphical User Interface) и разработанные на их основе WUI-интерфейсы (WUI – Web User Interface). Стилиевые детали WUI-интерфейсов не сильно разнятся от GUI-интерфейсов, подтверждением этому выступают диалоговые окна Web-браузеров.

Графический пользовательский интерфейс (Graphical User Interface – GUI) определяется как стиль взаимодействия «оператор – компьютер», в котором применяются четыре основных элемента: окна, пиктограммы, меню и указатели (рис. 2). Иногда GUI-интерфейс называют WIMP-интерфейсом (Windows – окна, Icons – пиктограммы, Menus – меню и Pointers – указатели).

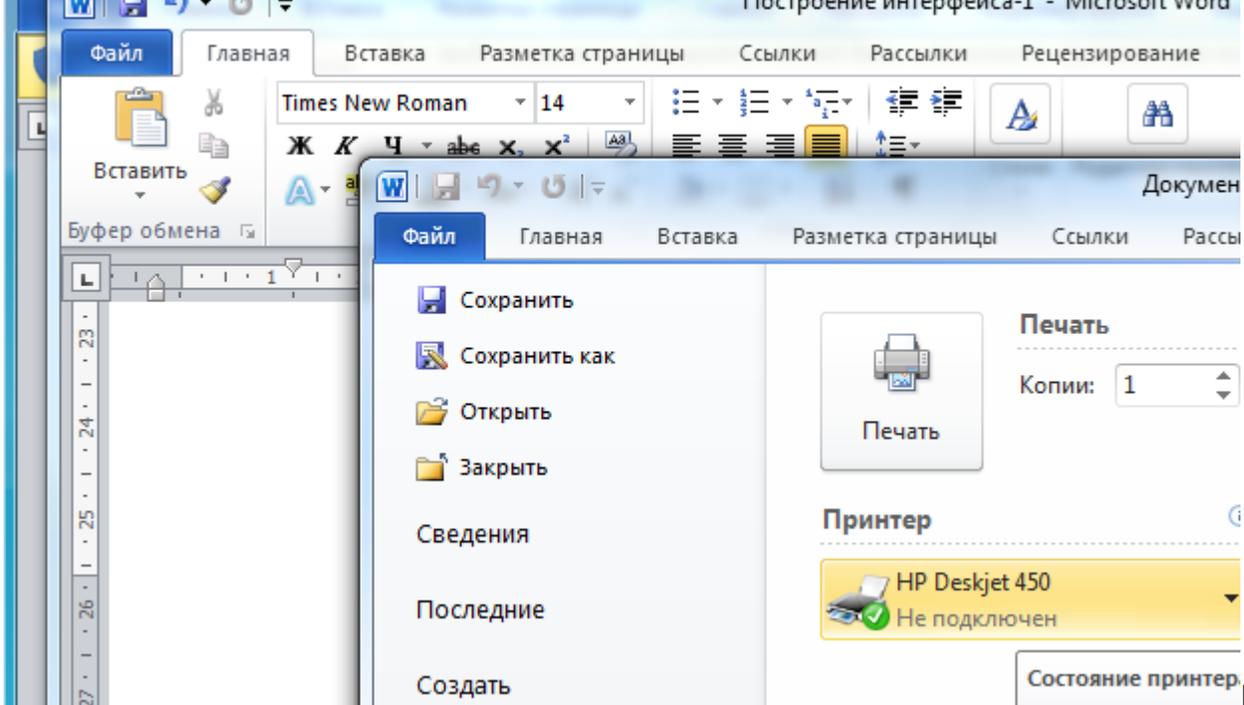


Рис. 2.

## Пример GUI-интерфейса

Важнейшие свойства GUI-интерфейса — это возможность непосредственного манипулирования, поддержка мыши или указателя (консоли пользователя), использование графики и наличие области для функций и данных приложения. Начальный анализ основ GUI-стиля ведётся отдельно от прикладного уровня GUI-ориентированных приложений.

Непосредственное манипулирование. Наиболее значительное свойство GUI-интерфейса заключается в непосредственном манипулировании, которое позволяет пользователю взаимодействовать с объектами интерфейса с помощью консоли. Например, окно можно передвинуть по экрану с помощью консоли, установив указатель на строку заголовка окна, нажав и удерживая кнопку мыши и двигая мышью (иногда эту операцию называют «захватить и перетащить» — «grab and drag»). Другой пример непосредственного манипулирования с помощью консоли — это выделение текста («занять [место] и ввести» — «swipe and type») или рисование непосредственно в графической области с использованием консоли и графических инструментов наподобие кисти (paint brush).

Подавляющее количество действий, выполняемых с помощью выбора альтернатив или меню, можно произвести, воспользовавшись непосредственным манипулированием. Например, во многих системах результатом перетаскивания пиктограммы документа на пиктограмму принтера на рабочем столе является

печать документа. К другим действиям, которые выполняются с помощью непосредственной манипуляции, относятся такие операции, как Move (Переместить), Copy (Копировать), Delete (Удалить) и Link (Связать).

Другие свойства. К некоторым другим свойствам работы ИП, присущим GUI-интерфейсу, принадлежат буфер обмена, сочетания клавиш, которые ускоряют клавиши в меню и диалогах, а также дополнительные возможности взаимодействия консоли. Несмотря на свою полезность, эти механизмы не рассматриваются как существенные свойства GUI-интерфейса.[4]

Базовый WUI-стиль (Web User Interface) сильно похож на меню иерархической структуры, которое пользователям известно по опыту работы в средах с графическим интерфейсом за исключением более наглядного представления и применения гиперссылок. Нужные переходы выполняются в рамках одного или нескольких приложений с использованием текстовых или визуальных гиперссылок. В зависимости от структуры гиперссылок приложения навигация в пределах WUI-интерфейса приводит к отображению Web-страниц в иерархии приложения по одной за раз — в линейном или нелинейном стиле внутри одного GUI-окна. Во многих отношениях WUI-ориентированные приложения это «шаг назад в будущее» — или, может быть, нечто худшее, учитывая объёмы электронных документов и других материалов в формате Web.

- Следствие рассматривает три версии авиакатастрофы
- Госдума разрешила признавать СМИ иностранными агентами
- Кремль прокомментировал ошибку Минобороны словами «ничего страшного»
- Актер Виталий Шаповалов скончался после операции
- Депутат рады обвинила Минск в ударе «ножом в спину» из-за Крыма

USD ЦБ 60,25 +0,63    EUR ЦБ 71,17 +1,49    НЕФТЬ 61,51 -0,85%    ...



18+

Тест: кем бы вы были в 1917 году?

Издательство Яндекса

Яндекс

[Видео](#) [Картинки](#) [Новости](#) [Карты](#) [Маркет](#) [Переводчик](#) [Музыка](#) [е](#)

Найдётся всё. Дата открытия америки

Рис. 3. Пример WUI-интерфейса — Web-браузер Internet Explorer

Основные принципы приложения, использующего WUI-стиль:

- ○ Информация обычно отображается в единственном GUI-окне, называемом Web-браузером, хотя для представления данных в приложении могут использоваться и несколько окон.
- Web-браузер обеспечивает меню для Web-приложения.
  - Выбор действий ограничен, так как меню, обеспечивающее обращение к функциям, не является легкодоступным для приложения.
  - Web-страница обладает малой степенью внутреннего контроля над клиентской областью для открытия специализированных всплывающих меню.
  - Создание специализированных меню требует от разработчиков дополнительной работы по программированию.
  - Функциональные возможности приложения должны преобразовываться в методы для вызова команд.
- Клиентская область не содержит традиционные пиктограммы. Большинство приложений используют графику и анимацию в эстетических или навигационных целях. Это содержит в себе потенциальную угрозу возникновения внешнего визуального воздействия и увеличения времени отклика при загрузке и открытии графических файлов.
- Web-браузер и приложения обеспечивают возможности отклонения графики, содержащейся в Web-страницах, поэтому на экране отображается только их текстовая версия.
- Поддержка консоли осуществляется в основном для выбора с помощью одинарного щелчка мышью или выбора по навигационным ссылкам. Технология «drag and drop» («перетащить и поместить») не поддерживается за исключением случаев специального программирования в определенных средах.

Web-ориентированное ПО становится все более похожим на GUI-ориентированное ПО (возможно потому, что пользователи неизменно требуют наличия популярных и полезных свойств GUI-интерфейса наподобие функции «drag and drop» или всплывающих меню).

Навигация. Переход от одной страницы к другой с использованием гиперссылок или механизма поиска — наиболее часто выполняемая процедура WUI-интерфейса. Страницы, которые встречает оператор, существуют в пределах того же самого или другого Web-сайта.

Web-браузер обеспечивает базовые возможности навигации для перемещения по меню Web-сайта и в пределах Web-сайтов линейным способом с помощью кнопок панели инструментов Back (Назад) и Forward (Вперед). Переходы от одной страницы сайта к другой в пределах одного и того же Web-сайта приложения выполняется с использованием гиперссылок, схемы Web-сайта, кнопок и панели меню.

Представление и поведение. Основное назначение Web-страницы представлено в обеспечении искомой информацией, включая навигационную структуру и организацию Web-узла. Web-страницы составлены из нескольких конструкций, которые представляют собой сочетание мозаик цветных графических элементов. По сравнению с GUI-ориентированными приложениями WUI-ориентированные приложения включают большое количество элементов поведения, которые не вызываются пользователем, например, анимационных. С точки зрения ИП в Internet царит полное безвластие.

Компоненты WUI-интерфейса. К наиболее распространенным компонентам WUI-интерфейса относятся баннеры (заголовки), навигационные панели и визуальные или текстовые гиперссылки, упорядоченные различными способами. Также применяются разнообразные подходы к использованию графики, анимации и цвета.

- Баннер представляет собой визуальный заголовок (header), отображаемый в левом верхнем углу Web-страницы.
- Панель навигации — это меню, состоящее из вариантов выбора гиперссылок, которые обеспечивают доступ к информации на сайте.
- Гиперссылка - это вариант перехода, при помощи которого осуществляется переход на следующую страницу информации или перемещает место курсора на другую область той же страницы.
- Макеты Web-страниц. Информация представляется на Web- страницах с использованием одного или нескольких разработанных макетов и выбранных навигационных стилей.
- Браузер. Обычно браузер обладает заголовком, навигационной панелью и основной областью, отображаемой в пределах экрана.
- Каталог. Каталог представляет собой визуальный поисковый механизм, в котором обозначены варианты выбора гиперссылок, используемых для перехода по дополнительным вариантам выбора до тех пор, пока не будет найден искомый результат. Допускаются навигационные панели в виде заголовков и другие типы навигации по вариантам выбора гиперссылок.

- Поиск и результаты поиска. Один или несколько элементов управления, с помощью которых оператор осуществляет ввод или выбор критерия поиска информации. Результаты поиска отображаются в том же или другом окне Web-браузера.
- Документ. Во многом похожий на свой бумажный двойник Web-документ отображает текстовую информацию вместе со ссылками на дополнительные источники или развернутое представление информации.
- Записная книжка. Некоторые Web-узлы представляют визуальную записную книжку в качестве метафоры для организации данных. Она почти не отличается от навигационной панели, с разницей в том, что содержит меньшее количество вариантов выбора.

Сбор данных. Дополнительной целью некоторых Web-страниц является сбор имен и адресов, а также другой информации о пользователях. Для достижения этого используются обычные элементы управления GUI-интерфейса, иногда с некоторыми функциональными ограничениями.

Проблемы разработки. Факторы успешной разработки, которые влияют на качество приложений, использующих Web-стиль ИП: простота переходов по иерархическим информационным ресурсам, лёгкость и быстрота поиска, а также быстрая реакция. К другим важным факторам относятся эстетические характеристики и ценность текущего содержания информации.[5]

## **5. Объектно-ориентированный интерфейс программы**

Проектирование программных объектов даёт возможность предоставить в распоряжение оператора приложение, наделенное объектно-ориентированным стилем ИП и/или объектно-ориентированной внутренней структурой (реализацией). Многие объектно-ориентированные свойства внешнего мира нашли отражение во внешнем виде, поведении, требованиях к взаимодействию и функциональных возможностях. Компьютеризованное усовершенствование или дополнение объектов реального мира, если только оно плохо спроектировано или реализовано, не очевидно для конечного пользователя и не в состоянии преодолеть его устоявшиеся знания и восприятие. Представленные в явном виде при проектировании, обозначения классов объектов, иерархии классов и наследование посредством иерархии классов остаются прозрачными для пользователя.

- Объектно-ориентированный прикладной интерфейс программы должен обладать следующими свойствами:
- обеспечивать непосредственное манипулирование (перетаскивать любые объекты куда угодно);
- обеспечивать непосредственный ввод данных (записывать любую информацию);
- обеспечивать контекстную зависимость от объектов (всплывающие (контекстные) меню, справки, согласованность и т. д.).[7]

Хороший прикладной объектно-ориентированный ИП прост в использовании; это значит, что его механизмы интерфейса программ прозрачны.

## 6. Проектирование интерфейса программ

Для получения эффективного результата разработки ИП используют различные подходы к проектированию:

1. Пользовательский подход (User Centered) — основными признаками этого подхода выступает ориентация на пользователя, т.е. в главную очередь требуется узнать, в чем нуждается пользователь и что хочет получить от разрабатываемого интерфейса. И далее в процессе разработки полученные требования реализуются в продукте. При сборе информации используются методы наблюдения за работой пользователя, проводятся опросы пользователей.
2. Системный подход (System). Пользователь рассматривается как маленькая интеллектуальная часть системы «человек – программный продукт».
3. Подход деятельности (Activity Centered). Изучается деятельность пользователя в как система, и постепенно оптимизируются её отдельные функции.
4. Итеративный подход (Agile) — метод последовательных приближений. Суть данного подхода заключается в создании чального самого простейшего примерного кода с целью показать заказчику и затем постепенно подрабатывать прототип, изучая реакцию заказчика после каждой итерации.
5. Экспертный подход (Genius). Заключается в следующем: эксперты собирают важную, по их мнению, информацию, ведут переговоры с заказчиком, задают нужные вопросы. На основе полученной информации создаётся интерфейс.
6. Целеориентированный подход проектирования (Goal Centered Design). Разработка интерфейса ориентируется на цель, которая будет достигнута

данным программным продуктом.

7. Средоориентированный подход. Разрабатывается среда интерфейса как место деятельности оператора.

При создании программного интерфейса рекомендуется широко пользоваться указанными подходами, учитывая при выборе методов: назначение разрабатываемого продукта, целевых пользователей, время и деньги для разработки.

Проектирование интерфейса программ представляет собой довольно сложный и многоэтапный процесс, каждый этап которого состоит в свою очередь из отдельных ступеней. В общем случае весь процесс можно представить в виде четырёх этапов (рис. 4), направленных на решение основной задачи — обеспечить оптимальное взаимодействие пользователя с системой.

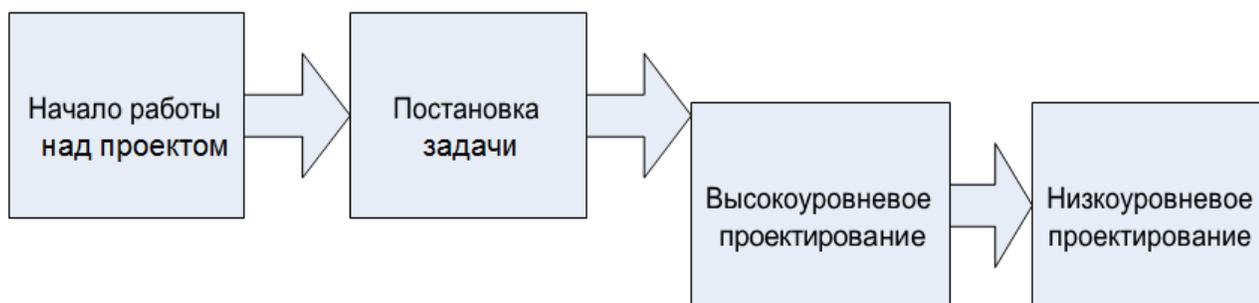


Рис. 4. Этапы эргономического проектирования интерфейса программ

При решении данной задачи нельзя рассматривать составляющие её части отдельно друг от друга. В системе «оператор – машина» главным звеном является оператор, а программа, являясь подчиненным звеном, должна реагировать на действия человека, а не наоборот. В связи с данным положением, возникает задача проектирования опыта взаимодействия.

Проектирование опыта взаимодействия — это новая область научно-практической деятельности, которая в последние годы выделяется как самостоятельная дисциплина, сосредоточенная на проектировании поведения пользователя продукта.

Проектирование взаимодействия — это описание возможного поведения оператора и определение того, как программа будет реагировать на его поведение, и приспосабливаться к нему.

Потребовалось немало времени, прежде чем проектировщики пришли к мысли, что нужно перейти от разработки программного обеспечения, хорошо работающего с точки зрения машины, к созданию программ, хорошо работающих с точки зрения человека.

Проектирование взаимодействия касается не столько моментов эстетики, сколько понимания потребностей пользователей и принципов их деятельности познания. Форма и эстетическая привлекательность программного продукта должны работать в гармоничной среде при достижении целей пользователей посредством правильно спроектированного поведения программного продукта.

Именно опыт взаимодействия с программным продуктом формирует у пользователя впечатление о товарах или услугах, предлагаемых компанией-разработчиком, именно опыт взаимодействия отличает компанию от её оппонентов, определяет — будет ли пользователь в дальнейшем работать с нашим программным продуктом или предпочтёт конкурента.

Неотрицательный опыт взаимодействия является интегралом множества крупных и мелких проектов о том, как продукт выглядит, как себя ведёт, и какие действия позволяет совершать операторам. Эти решения воздействующие друг на друга, формируют опыт взаимодействия и влияют на все его стороны.

С переходом на каждый последующий уровень вопросы, на которые проектировщики ищут ответы, становятся всё более конкретными. На нижнем уровне мало кто думает об окончательном внешнем виде продукта. Волнует лишь то, насколько проектируемый продукт впишется в стратегию проектирования, при этом удовлетворяя потребности пользователей. От этапа к этапу (уровня к уровню) разрабатываемые решения становятся более конкретными и обретают новую степень детализации.

Каждый уровень зависит от уровней расположенных ниже: поверхность зависит от компоновки, которая зависит от структуры, которая зависит от набора возможностей, которые в свою очередь зависят от стратегии.

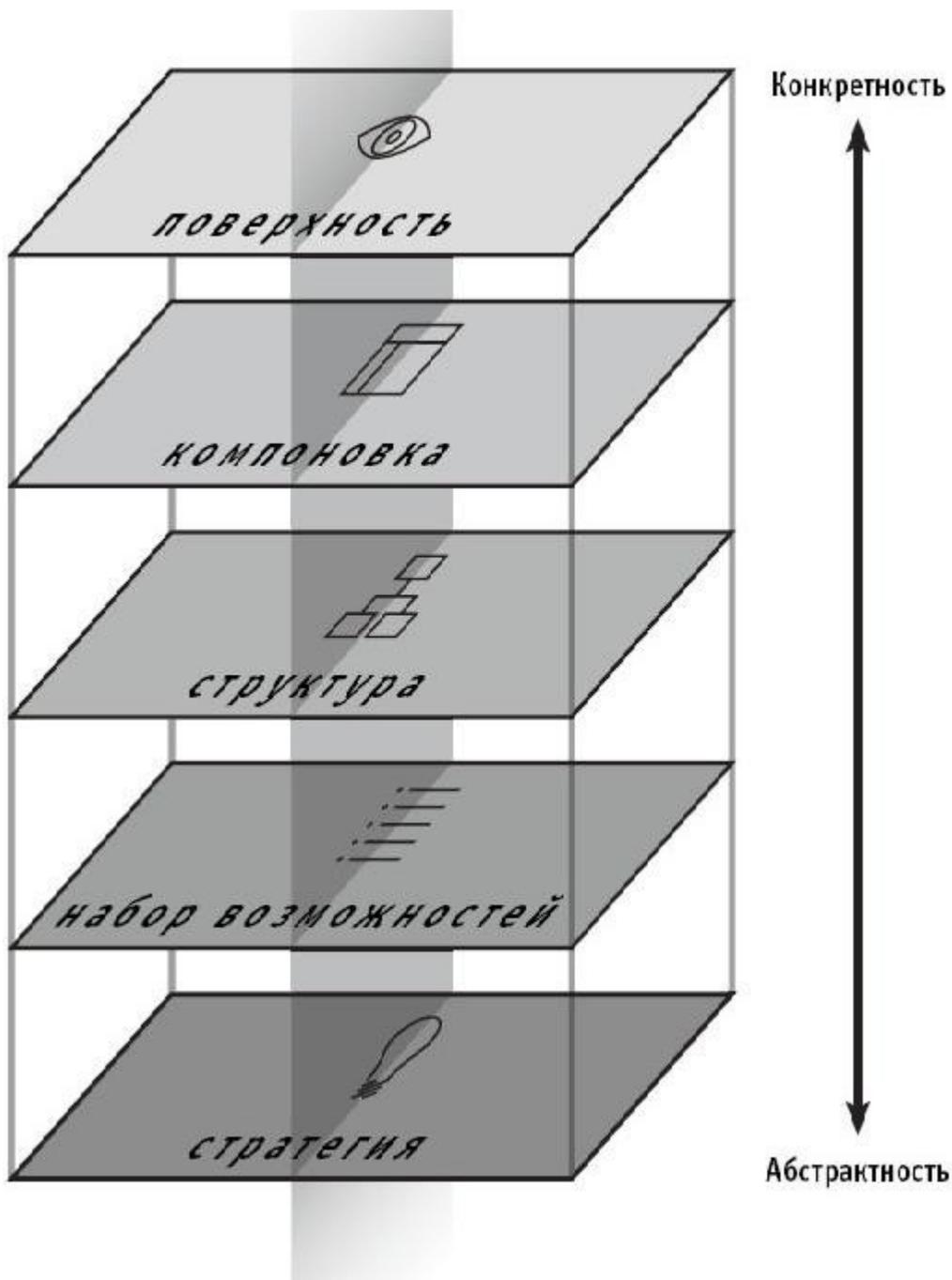


Рис. 5. Основные уровни проектирования взаимодействия

Данный подход представлен в виде уровней (рис. 5). Обобщая всё вышесказанное, рассмотрим каждый из них более детально.

## Начало работ над проектом

На этом уровне определяются объёмы работ, планируются затраты и т. п. Длительность этого этапа, как правило, не превышает 5-8% от общей

продолжительности проектирования. Для адекватной оценки ресурсов (времени, денег, количества специалистов) требуемых для разработки (переработки) программного интерфейса, необходимо достаточно хорошо представлять себе тот объём информации, с которой следует ознакомиться. К ней относится информация о предметной области и прототипах. Она получается из письменных источников и опросов экспертов. Результатом этой работы является количественная оценка трудоемкости проекта.

Чтобы предлагать соответствующие решения программного интерфейса, необходимо иметь ясное представление о предметной области программного обеспечения. Предметная область изучается по литературе, кроме того, весьма полезны беседы с опытными пользователями, другими сотрудниками (экспертами) для выяснения всех деталей и характеристик предметной области. Вместе с «жалобами» заказчика на текущую версию программного обеспечения результаты этого этапа составляют основное содержание работы над проектом (экспертная оценка часто обнаруживает проблемы, которые заказчику не видны, маскируясь под другие). Проблемы, выявленные на данном этапе, должны быть решены в новом интерфейсе. Удачные решения желательно сохранить, чтобы имеющимся пользователям не пришлось переучиваться (и чтобы сократить затраты на переделку). По сути, этот этап завершается созданием перечня удачных и неудачных интерфейсных решений (основное внимание уделяется решениям неудачным). На этом этапе проводится удобство тестирование текущей версии интерфейса, и составляются краткие протоколы и перечень выводов исследования.

## **Постановка задачи. Сбор информации о разрабатываемом продукте**

На этой стадии анализируют данные о пользователях, формализуется функциональность и определяются критерии оценки проекта.

Залогом успешного проектирования интерфейса программ является наличие наиболее полной информации об аудитории пользователей: их целях, задачах, предпочтениях, привычках и представлениях, и о заказчиках. Чем более полная информация о продукте будет собрана и передана проектировщикам, тем более чёткое и правильное представление о его качествах будет у них сформировано и, соответственно, тем эффективнее будет проходить процесс разработки на всех последующих стадиях.

Данные о пользователях и о проекте должны содержать следующие позиции:

- характеристики пользователей: их опыт работы с компьютером, знание предметной области, мотивы, размер/важность групп пользователей, примеры (типичные ситуации) использования;
- цели и задачи пользователей;
- задачи проекта: что послужило причиной создания проекта, этапы создания проекта, какие результаты должны быть получены, какая информация необходима и когда;
- технология разработки и платформа, на которой будут работать пользователи;
- среда, в которой будет создаваться и использоваться проект (программная, физическая, рыночная, организационная и культурная).

Эта работа предполагает доступ к имеющимся и потенциальным пользователям системы, экспертам и проектной документации. На этом этапе разрабатываются пользовательские профили, модели пользователей. Обязательно должна присутствовать информация о субъективных ожиданиях пользователей системы. Без этого трудно или невозможно предугадать отношение пользователей к будущей системе. Поэтому должны быть описаны свойства, которым должен отвечать интерфейс для повышения субъективного удовлетворения, приведён перечень значимых для пользователей характеристик системы. Завершается эта часть работы описанием среды, в которой используется система, и основных характеристик ИП. Характеристики ИП отражаются в версии прототипа ИП, которая на данном этапе будет, скорее всего, бумажной. Проводится тестирование удобства использования этой версии прототипа, и определяются скорость работы, количество ошибок оператора, скорость обучения, удовлетворенность оператора и т. п.

Иными словами, на данном этапе обобщаются настоящие итоги разработки готового программного интерфейса.

Здесь нужно убедиться, что пользователи не нуждаются в инструментах сами по себе, они нуждаются в результатах работы этих инструментов. Никто не нуждается в программе обработки космических изображений — нуждаются в уже обработанных изображениях. Поэтому пользователям нужно просто надежное средство, наделенное инструментарием выполнять работу определенного характера, достигать поставленных целей.

Алгоритмизация характерных действий пользователя начинается с документирования различных формуляров типовых алгоритмов. В связи с этим формируется информация, необходимая пользователям для выполнения работы, поэтапность самой работы, наличие признаков окончания самой работы. На выходе исследования появляется логическое обозначение взаимодействия пользователя с разрабатываемым программным интерфейсом, не углубляясь, каким образом происходит взаимодействие, но уделяя максимальное внимание всем задачам, разрабатываемым для пользователей. Конечное число вариантов сценариев может быть бесконечным, за исключением того, что они в них должны быть включены все виды подзадач, стоящих перед системой, и быть приближенными к реальности. Сценарии очень удобно различать по именам существующих в них вымышленных персонажей.

На данном этапе также необходим анализ уже существующих программных интерфейсов. Подавляющее число пользователей существующих систем имеют навыки использования нескольких информационных систем различных брендов; если разрабатываемая система полностью отличена от конкурирующих брендов, то пользователям как ни печально, но придется переучиваться. Кроме того, системы конкурирующих брендов часто содержат более нужные решения, которые будет полезно применять (или учитывать при разработке интерфейса). Как и в случае экспертной оценки текущего интерфейса системы, отчет по выполнению этого этапа работ содержит перечень удачных и неудачных интерфейсных решений, фокусируясь на более удачных решениях.

Проанализируем основные методы сбора нужной информации, которая пригодится в процессе написания интерфейса программ.

## **Исследование пользователей**

Оценить результаты разработки в конечном итоге следует исходя из того, насколько более точно он отвечает пожеланиям оператора и фирмы — заказчика проектирования. Если у разработчиков нет четкого понимания о пользователях, для которых осуществляется разработка, а также у него нет понимания возможных имеющихся ограничений, производственных задач и целей, которые сами по себе являются двигателем разработки, то вероятность хорошего результата невероятно мала — не имеет значения, насколько при этом хороши профессионализм и способности разработчиков.

Пользующиеся спросом методы сбора и обработки данных используют математические методы. Эта группа методов даёт ответ на вопрос «Сколько?», а данные, получаемые в результате применения математических методов, обрабатываются с использованием статистических вариантов сбора и анализа. Пример использования математических методов сбора информации — получение процентной дисперсии, какая часть выборки потребляет тот или иной продукт, знает данную марку, алгоритмы использования, свойства продукта и так далее.

В основе методик количественных исследований лежат чёткие математические и статистические модели, что позволяет в результате иметь не мнения и предположения, а точные количественные (числовые) значения изучаемых показателей. Особенность этой группы методов заключается в их высокой степени формализованности. Используемый инструментарий состоит из переменных, заданных исследователем заранее, он достаточно «жёсткий» и практически не меняется в рамках проекта. Высокая степень формализации количественных методик сочетается с их ориентацией на массовый сбор первичных данных и их статистическую обработку. При использовании количественных методов сбора информации исходной позицией является выборочная совокупность (выборка) и принцип репрезентативности.

Вместе с тем необходимо отметить, что настоящего понимания в изучаемых областях невозможно достичь, изучая только многочисленные отчёты, полученные в результате количественных исследований, хотя для ответов на другие вопросы эти данные могут оказаться крайне важными. Глубокое знание вопроса можно получить, лишь дополнив количественные данные качественными моделями и интерпретациями. Существует много видов качественных исследований, и каждое из них может сыграть важную роль в формировании общей картины проектируемого продукта.

## **Качественные исследования**

Качественные методы сбора данных позволяют получить информацию, которая отвечает на вопрос «Почему?» и не может быть обработана при помощи статистических методов анализа. Иными словами, из информации, собранной с использованием какого-либо из качественных методов, нельзя вывести какие-либо проценты и распределения, она позволяет лишь понять различные факторы и мотивы определенных действий, выявить модели поведения и т. п. Качественные

(неформализованные) методы ориентированы не на массовый сбор данных, а на достижение углубленного понимания исследуемых явлений. Отсутствие формализации делает невозможным массовый охват обследуемых объектов, в результате чего число единиц обследования часто снижаются до минимума. Отказ от широты охвата компенсируется

«глубиной» исследования, т. е. детальным изучением явления в его целостности и непосредственной взаимосвязи с другими явлениями.

Качественные исследования позволяют получить глубокую, развернутую информацию о предмете исследования. В отличие от количественных, качественные исследования фокусируются не на статистических измерениях, а опираются на понимание, объяснение и интерпретацию эмпирических данных и являются источником формирования гипотез и продуктивных идей. В качественных исследованиях широко используются проективные и стимулирующие техники — неструктурированные, не директивные способы задавать вопросы, которые помогают исследователю раскрыть мотивы, верования, установки, отношения, предпочтения, ценности, степень удовлетворенности, проблемы респондентов и прочее относительно разрабатываемого продукта или брендов. Проективные техники способствуют преодолению таких трудностей коммуникации, как вербализация чувств, отношений и т. п., а также выявлению латентных мотивов, неявных установок, вытесняемых чувств.

Качественные исследования помогают понять предметную область, контекст и ограничения продукта, более действенным способом, чем количественные исследования. Они также помогают выявить шаблоны поведения потенциальных пользователей продукта быстрее и проще по сравнению с количественными методами. Качественные методы позволяют изучить:

поведение, взгляды, склонности потенциальных пользователей продукта;

предметную область — технический, экологический, и деловой контексты разрабатываемого продукта;

используемый лексикон и прочие социальные аспекты предметной области;

способы применения существующих продуктов. Качественные исследования способствуют ходу проектирования,

- ○ ■ поскольку:

обеспечивают доверие и уважение к команде проектировщиков;

объединяют команду общим для всех пониманием особенностей предметной области и проблем пользователей;

дают руководителям возможность принимать решения по тем или иным вопросам проектирования продукта на основе данных — вместо догадок и личных предпочтений.

Значение качественных исследований не ограничивается поддержкой процесса проектирования. Время, потраченное на подробное изучение пользовательской аудитории, способно принести плоды в виде важных для бизнеса решений, к которым невозможно прийти средствами традиционных исследований рынка.[7]

## **Методы качественных исследований**

Существует множество методов и приёмов проведения качественных исследований, но сосредоточим внимание на методиках, которые хорошо зарекомендовали себя в международной практике разработки пользовательских интерфейсов в течение последних десяти лет.

Перечень основных методов качественных исследований:

- консультации заинтересованных лиц;
- консультации экспертов в предложенной предметной области ;
- консультация пользователей и покупателей;
- наблюдение за пользователями;
- изучение документации;
- аудит продукта/прототипа и конкурирующих решений.

## **Консультации заинтересованных лиц**

Перед созданием любого нового программного продукта должно проводиться комплексное исследование, с помощью которого получают представление о техническом окружении и о составляющей бизнеса продукта. В большинстве случаев продукт разрабатывается для достижения одной или нескольких конкретных бизнес идей. Обязанность разработчиков — предлагать концепции, не теряя из виду это бизнес цели, и поэтому крайне важно, чтобы бригада

разработчиков начинала работу с изучения возможностей и ограничений, стоящих за спецификацией проекта.

В общем случае заинтересованное лицо — это любая организация, наделенная полномочиями в отношении разрабатываемого программного продукта или несущая ответственность за какую-либо его часть. Говоря более конкретно, заинтересованные лица — это ключевая команда предприятия, иницилирующей работы по проекту; обычно в эту команду входят высшие менеджеры, менеджеры среднего звена и начальники отделов разработки, сектора производства, маркетинга, удобства, дизайна.

Консультации заинтересованной группы лиц должно проводиться до старта всяких исследований пользовательской группы, поскольку вполне могут возникнуть спорные ситуации и обсуждения нередко задают способы проведения пользовательских исследований. Нередко наиболее эффективно консультировать заинтересованных лиц поодиночке, а не в группах, объединяющих несколько подразделений.

От заинтересованных лиц важно получить информацию по следующим вопросам:

- ○ ■ предварительное видение продукта;
- бюджет и график проекта;
- технические возможности и ограничения;
- потребности бизнеса;
- представления заинтересованных лиц о пользователях.

Обсуждение этих тем важно для выработки общего языка и взаимопонимания между группами проектировщиков, руководителей и разработчиков. Функция проектировщика — создать видение продукта, в которое верит вся команда. Участники проекта вряд ли увидят в предлагаемых решениях отражение своей системы ценностей, если не потратят достаточное время на то, чтобы понять их взгляды.

Консультация экспертов-разработчиков в предметной области

На начальных этапах проектирования неоценимый вклад вносит выявление и консультация группы экспертов в предметной области — людей, разбирающихся в области разработки, на которую ориентирован разрабатываемый продукт. Как и группа заинтересованных лиц, эксперты предметной области способны выставить продукт и его пользователей с выгодной точки зрения, при этом проектировщикам

требуется проявлять осторожность и понимать, что мнение эксперта предметной области в определенном смысле неверна.

### Консультация заказчиков

Заказчики — это пользователи, принимающие решение о приобретении разрабатываемого программного продукта. Если речь идет о потребительских продуктах, покупатели нередко являются пользователями продукта, хотя если продукты ориентированы на детей или подростков, то покупателями будут их родители или опекуны. В случае с большим количеством социальных, технических, и техногенных продуктов покупателем зачастую оказывается не конкретный пользователь, а кто-то среди высших менеджеров фирмы или соответствующий директор или его заместитель, имеющий свои цели и потребности, которые разнятся с операторами программ. Чтобы сгенерировать жизнеспособный программный продукт, важно понимание целей пользователей ПК. Также важно понимать, что те люди, которые приобретают программу, редко сами пользуются продуктом, но когда все же делают это, то иначе чем пользователи.

В консультации с заказчиком важно выяснить:

- ○ ■ ■ цели заказчика;
- неудовлетворенность настоящей ситуации;
- алгоритм приобретения программного продукта;
- место заказчика при покупке программы;
- изучение предметной области.

### Консультация операторов интерфейса программы

Пользователи продукта в создании программного продукта находятся в главном контуре внимания. Именно пользователи личными действиями стараются достичь каких-нибудь выводов при помощи приобретаемой программы. При перекодировке или оптимизации настоящего ПО важным моментом является общение не только с настоящими, но и с возможными пользователями продукта и входят в его намеченных пользователей. Интервьюирование обеих категорий позволяет выявить влияние, которое оказывает на поведение и образ мысли пользователя опыт работы с существующей версией продукта.

Разработчиков интересует следующая информация:

- для каких целей подразумевается применение программного продукта;

- образованность пользователя, уровень знаний применяемой области;
- настоящая область деятельности пользователей существующая и предполагаемая;
- цель внедрения программы;
- мнение пользователей о своей повседневной работе и деятельности;
- возможные подводные камни.

## **Мониторинг состояния пользователей**

Масса населения земного шара не способно точно выразить свое собственное настроение, особенно когда находятся в разрезе повседневной жизни. Также многие люди, из-за боязни показаться некомпетентными или невежливыми избегают обсуждать поведение программ, которое кажется им проблемным и непонятым. Из этого следует, что интервью, проводимое вне контекста ситуации, которую стремится понять проектировщик, даст менее полные и менее точные данные. Можно обсудить с пользователями их представление о собственном поведении, а можно непосредственно наблюдать это поведение. Второй вариант даёт лучшие результаты.

Для фиксации того, что говорят и делают пользователи, многие специалисты применяют технические средства, такие как аудио и видеозапись. Следует проявлять осторожность и не пользоваться такими устройствами слишком бесцеремонно по отношению к пользователям, в противном случае они будут отвлекаться и вести себя не так, как в отсутствие записывающих устройств.[7]

## **Обзор литературы**

Вместе с консультацией заинтересованных лиц команде разработчиков надлежит изучить множество книг и литературы, касающихся продукта и его предметной области. Сюда могут и должны быть включены маркетинговые исследования, стратегия бренда, исследования рынка, опросы пользователей, технические спецификации и информационные материалы, статьи в деловых и технических журналах, связанных с предметной областью, сравнительный анализ конкурентных решений и т. п.

Команда проектировщиков должна собрать эту литературу и использовать её как основу для формирования списка вопросов к заинтересованным лицам и ЭПО, а

затем в качестве источника дополнительных данных о предметной области и терминологии, а также для сравнения с уже собранными данными о пользователях.

## **Прочие виды исследований. Группы фокуса**

В настоящее время очень сильно распространен метод сбора пользовательских данных при помощи групп фокуса, в котором группа пользователей, отобранных из предварительной выборки по заранее выявленным демографическим параметрам, собирается в одной аудитории и пытается ответить на подобранный набор вопросов или выбирает ответы из предложенных вариантов. Встреча записывается на видео или аудио аппаратуру для последующего после анализа. Группы фокуса — это стандартный приём в маркетинге традиционных и нетрадиционных продуктов. Группы фокуса оказываются полезны для оценки прямых реакций на форму продукта, его техническое содержимое или внешний вид. Группы фокуса также позволяют оценивать отзывы на продукт, которым опрашиваемые лица уже пользовались в течение определенного интервала времени.

Способ групп фокуса во многих случаях непригоден в качестве инструментария разработчика. Группы фокуса подходят в тех случаях, когда нужно получить сведения о продуктах, которыми люди уже владеют или хотели бы приобрести, но не очень подходят для сбора данных о том, что люди делают с этими продуктами, а также каким образом и почему они это делают. Кроме того, группы фокуса в силу своих коллективных истоков имеют свойство приходить к согласию: мнением группы становится мнение, высказанное большинством голосов или произнесенное громче других. Это напрямую противоречит пониманию процесса разработки, потому что проектировщикам необходимо усвоить все способы поведения, чтобы учесть их при разработке продукта. Группы фокуса обычно вырождают ту полифонию поведения и мнений, которые как раз и должны приниматься в расчёт разработчиками.

## **Заключение**

Проектирование интерфейса пользователя представляет собой инженерную систематическую задачу. На практике невозможно создать многопользовательский программный интерфейс, который решает все задачи, поставленные перед разработчиком. Это поиск компромисса. Интерфейс должен быть скорректирован

под определенные исследователями задачи. Его внешний вид и свойства определяются мнением группы пользователей экспертов. При этом учитывается область трудовой деятельности самого пользователя, его опыт и содержимое сферы взаимодействия. Интерфейс пользователя изменяет свой вид в зависимости от содержания задачи и содержимого окружающей среды. Каждый новый вид интерфейса сопровождается рождением новых объектов, которые отображают параметры окружения и средства действий с ними для достижения поставленной целевой задачи. Такой интерфейс с приспособленными по отношению к задаче свойствами отличается от традиционных систем вывода информации, которые реализуют главным образом функции визуализации и манипуляции. Высокая степень интерактивности, внедряемая в интерфейс, не должна приравниваться к высокой визуальной и детальной насыщенности среды, которая должна соответствовать решаемой задаче. Высокая нагрузка зрительного восприятия подробностями зрительной сцены может снизить скорость и качество деятельности оператора ПК.

К сожалению, приоритет техно-проектирования над инженерно-психологическим всё еще присутствует в отечественных фирмах по разработке ПО. В результате чего практикуется низкая визуальная проработка систем автоматизации и интеллектуальных функций интерфейса. Эта проблема, переросшая из области проектирования в сферу отставания отечественных технологий интерфейса от передовых европейских стран. Отдельными программными методами невозможно решить задачи, требующие специальных знаний в области инженерной психологии и восприятия. В новых системах должно быть выделено самое серьёзное внимание решению вопросов повышения эффективности человеческого звена. Человек по-прежнему является главным элементом, определяющим эффективность работы любой визуальной системы.

## Список литературы

- ○ ■ 1. Баканов, А. С. Проектирование интерфейса программ: эргономический подход / А. С. Баканов, А. А. Обознов. – М.: Изд-во «Институт психологии РАН», 2009.
- ○ ■ 1. Бондаровская, В. М. Эргономика периферийного оборудования для систем диалогового взаимодействия / В. М. Бондаровская // Приборы и системы управления. – 1981. – № 7.

2. Войненко, В. М. Эргономические принципы конструирования / В. М. Войненко, В. М. Мунипов. – Киев: Техника, 1988.
3. Вудсон, У. Справочник по инженерной психологии для инженеров и художников-конструкторов / У. Вудсон, Д. Коновер. – М.: Мир, 1968.
4. Гаррет, Дж. Веб-дизайн: книга Джесса Гаррета. Элементы опыта взаимодействия / Дж. Гарретт. – СПб: Символ-Плюс, 2008.
5. Инженерная психология в применении к проектированию оборудования: пер. с англ.; под ред. Б. Ф. Ломова и В. И. Петрова. — М.: Машиностроение, 1971.
6. Купер, А. Алан Купер об интерфейсе. Основы проектирования взаимодействия / А. Купер, Р. Рейман, Д. Кронин. Пер. с англ. – СПб.: Символ-Плюс, 2009.
7. Ломов, Б. Ф. Справочник по инженерной психологии / Б. Ф. Ломов. – М.: Машиностроение, 1982.