Тема 3

**Проектирование пользовательского интерфейса**

Создание проекта включает следующие стадии:

* начало работы над проектом;
* постановка задачи;
* высокоуровневое проектирование;
* создание прототипа;
* тестирование прототипа;
* создание системы;
* тестирование пользовательского интерфейса.

**Начало работы над проектом**

На этой стадии определяется окончательный объем работ, планируются затраты и т.п. Длительность этой стадии как правило не превышает 3-7 процентов общей длительности проекта. При этом осуществляются следующие этапы:

* Оценка объема предметной области;
* Изучение предметной области;
* Экспертная оценка текущего интерфейса.

**Оценка объема предметной области**

На этом этапе адекватно оцениваются ресурсы (время, деньги, количество экспертов), которые будет необходимо потратить на разработку (переработку) интерфейса, необходимо четко представлять объем информации, с которой следует ознакомиться. Именно этой цели и служит данный этап.

* На входе: доступ к информации о предметной области (литература, эксперты).
* На выходе: количественная оценка ресурсоемкости проекта.

**Изучение предметной области**

На этом этапе предлагаются адекватные интерфейсные решения, необходимо иметь четкое представление о предметной области системы. Предметная область изучается по литературе; также, как правило, проводится серия интервью с экспертами для выяснения основных аспектов и характеристик предметной области.

* На входе: доступ к информации о предметной области (литература, эксперты).
* На выходе: знания основных аспектов предметной области.

**Экспертная оценка текущего интерфейса системы**

На этом этапе формируется 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; наоборот, удачные решения желательно сохранить, чтобы имеющимся пользователям не пришлось полностью переучиваться (и чтобы сократить затраты на переделку).

* На входе: доступ к текущему интерфейсу системы, пользователям, знания основных аспектов предметной области.
* На выходе: рекомендации по оптимизации интерфейса, перечень удачных и неудачных интерфейсных решений (основное внимание уделяется решениям не удачным). Если на этом этапе проводилось юзабилити-тестирование текущей версии, отчет содержит краткие протоколы и перечень выводов исследования.

**Постановка задачи**

На этой стадии собираются и анализируются данные о пользователях, формализуется функциональность и определяются объективные критерии успеха проекта. Важность этой стадии трудно переоценить. Здесь закладываются основные концепции системы, влияющие абсолютно на все показатели качества её интерфейса.

На этой стадии выделяют следующие этапы:

* формализация контекста использования;
* формализация объективных критериев успеха;
* определение необходимой функциональности системы;
* анализ целей;
* анализ действий пользователей;
* определение функциональности системы;
* формализация бизнес-ролей пользователей;
* формализация функциональности;
* формализация сценариев действий пользователей;
* обзор интерфейса конкурирующих систем;
* формализация привычек и ожиданий пользователей.

**Формализация контекста использования**

На этом этапе собирается большинство сведений о пользователях. Описываются следующие свойства аудитории системы:

* Характеристики пользователей: их опыт работы с компьютером, знание предметной области, мотивы, размер/важность групп пользователей, образцы (типовые 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 (физическая, рыночная, организационная и культурная).

На входе: доступ к имеющимся и потенциальным пользователям системы, экспертам и проектной документации.

На выходе: описание контекста использования системы, возможно более детальное описание свойств пользователей.

**Формализация объективных критериев успеха**

На этом этапе выделяются объективные критерии оценки эргономичности интерфейса, такие как показатели эффективности, продуктивности, удовлетворенности пользователей (на более ранних этапах выделить эти критерии невозможно). На данном этапе создается реальное задание на проектирование интерфейса.

На входе: доступ к пользователям, экспертам и проектной документации.

На выходе: список объективных критериев успеха.

**Определение необходимой функциональности системы**

На этом этапе необходимо определить функциональность будущей системы. Современная наука выдвинула два основных способа, а именно анализ целей и анализ действий пользователей. Эти способы фактически не конфликтуют друг с другом, более того, в процессе определения функциональности желательно использовать оба.

**Анализ целей**

Результатом этого этапа должен являться список целей. После того, как истинные цели пользователей установлены (и доказано, что таких пользователей достаточно много, чтобы оправдать создание системы), приходит время выбирать конкретный способ реализации функции, для чего используется этап анализа действий пользователя.

**Анализ действий пользователей**

На этом этапе требуется формулировкасовершения определенных действий. При этом используется наблюдение за людьми, выполняющими свою задачу, пользуясь уже имеющимися инструментами, а именно системами конкурентов (если они есть) и предметами реального мира (поскольку очень немного новых действий появилось только после появления компьютеров). Неплохим источником материала для анализа часто служит даже не наблюдение за людьми, но анализ результатов их работы – если оказывается, что результат работы практически не зависит

от используемого инструмента, это значит, что нужна только та

функциональность, которая оказала воздействие на результат

(т.е. функции, которыми никто не воспользовался, не нужны).

**Низкоуровневые и высокоуровневые функции**

На этом этапе существуют два принципиально разных подхода к определению функциональности системы. При первом подходе система снабжается максимальным количеством функций, при этом результаты многих из них являются суммой результатов других функций. При другом подходе все составные функции (метафункции) из системы изымаются.

Оба подхода имеют как недостатки, так и достоинства. Подход, при котором количество функций ограничено, позволяет упрощать интерфейс, но при этом требует от пользователя понимать, как из многих низкоуровневых функций «собирать» функции более сложные. Подход, при котором помимо низкоуровневых функций есть высокоуровневые, позволяет потенциально обеспечивать большую скорость работы (за счет отсутствия пауз между низкоуровневыми функциями), но зато требует от пользователя знаний о том, где эти высокоуровневые функции найти и как с ними работать, при этом они перегружают интерфейс. Судя по всему, людям больше нравится пользоваться низкоуровневыми функциями, поскольку это позволяет добиваться более тонких и предсказуемых результатов.

**Формализация бизнес-ролей пользователей**

На данном этапе функциональность любой системы разделяется на несколько ролей пользователей: разным пользователям нужны разные блоки функциональности (в системах автоматизации эти роли называются бизнес-ролями). Навигация по системе прямо зависит от этих ролей, поскольку в пределах одной роли в навигацию не желательно включать функции из чужой роли. Соответственно, на этом этапе выделяются основные роли пользователей с относящимися к этим ролям функциями.

На этом этапе можно применять метод наблюдения за

людьми, выполняющими свою задачу, пользуясь уже

имеющимися инструментами, и именно системы конкурентов (если они есть) и предметами реального мира. Неплохим источником материала для анализа часто служит даже не наблюдения за людьми, но анализ результатов их работы – если оказывается, что результат работы практически не зависит от используемого инструмента, это значит, что нужна только та функциональность, которая оказала воздействие на результат (т.е. функции, которыми никто не воспользовался, не нужны). Обычно есть несколько разных способов реализации одной и той же функции.

Анализ действий пользователей как раз и позволяет определить, какой именно способ следует реализовать. Поскольку на этом этапе мы узнаём, какая именно функциональность нужна для каждой бизнес-роли, можно избрать верный путь по правилу «чем меньше действий требуется от пользователя, тем лучше».

На входе: доступ к пользователям, экспертам и проектной документации, знание основных аспектов предметной области.

На выходе: описание бизнес-ролей пользователей.

**Формализация функциональности**

На этом этапе, основываясь на информации, выработанной на предыдущих этапах, окончательно формируется список функциональных возможностей новой версии системы. Ранее сформированное ТЗ порой не включает части необходимой функциональности, либо содержит функциональность, реально не требующуюся пользователям.

* На входе: доступ к пользователям, экспертам и проектной документации, знание основных аспектов предметной области.
* На выходе: описание функциональности системы (отчет по выполнению этого этапа работы обычно не создается, вместо этого модернизируется уже созданное техническое задание).

**Формализация сценариев действий пользователей**

На этом этапе частично изучаются, частично разрабатываются типовые сценарии действий пользователя: формализуются данные, необходимые пользователям для выполнения работы, последовательность самой работы, критерии завершенности этой работы. Цель этого этапа – написать словесное описание взаимодействия пользователя с системой, не конкретизируя, как именно проходит взаимодействие, но уделяя возможно большее внимание всем целям пользователей.

На входе: доступ к пользователям, экспертам и проектной документации, знание основных аспектов предметной области.

На выходе: сценарии работы пользователей (разработанные сценарии, как правило, представляются в виде блок-схем, описывающих весь процесс использования системы для выполнения той или иной задачи).

**Обзор интерфейса конкурирующих систем**

На этом этапе следует учесть, что большая часть аудитории любой системы обладает навыками использования нескольких конкурирующих систем; если разрабатываемый интерфейс полностью несхож с конкурентами, пользователям придется переучиваться. Кроме того, конкурирующие системы часто содержат эффективные решения, которые полезно перенять (или, что чаще случается, учесть при проектировании интерфейса). Как и в случае экспертной оценки текущего интерфейса системы, отчет по выполнению этого этапа работ содержит перечень удачных и неудачных интерфейсных решений; в целом, однако, отчет более сфокусирован на удачных решениях.

На входе: доступ к конкурирующим системам.

На выходе: обзор преимуществ и недостатков интерфейса конкурирующих систем.

**Формализация привычек и ожиданий пользователей**

На этом этапе изучаются субъективные ожидания пользователей от системы. Без этого исследования трудно или невозможно предугадать отношение пользователей к будущей системе.

На входе: доступ к пользователям.

На выходе: описание характеристик, которым должен отвечать интерфейс для повышения субъективного удовлетворения, перечень значимых для пользователей характеристик системы. В зависимости от выбранного метода исследования, содержит либо числовые, либо оценочные данные.

**Прилив вдохновения**

Этот этап либо происходит сам, либо не происходит вовсе. Предыдущие этапы дают достаточное представление о сущности создаваемой системы. Поэтому всегда не вредно несколько задержаться на этом этапе и спросить себя «верно ли то, что методы, которые я собираюсь избрать, самые лучшие? Нельзя ли их сделать принципиально лучше?» Как правило, вдохновение, пришедшее на дальнейших этапах, обходится значительно дороже.

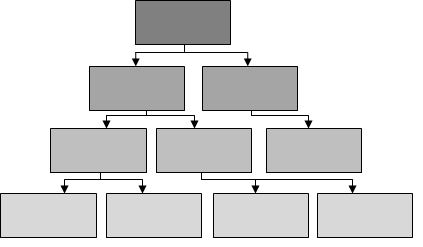
**Высокоуровневое проектирование**

На этой стадии начинается непосредственное проектирование интерфейса; предыдущие стадии и этапы посвящены исключительно сбору данных и постановке задачи. Эта стадия содержит следующие этапы: проектирование структуры экранов системы; выделение независимых блоков; проектирование отдельных блоков; проектирование навигационной системы; проектирование структуры справочной системы; низкоуровневое проектирование.

**Проектирование структуры экранов системы**

На этом этапе, основываясь на сценариях работы и ролях пользователей, формируется структура экранов системы, т.е. определяется количество экранов, функциональность каждого из них, навигационные связи между ними, формируется структура меню и других навигационных элементов.

По сути, на этом этапе выделяются отдельные функциональные блоки и определяется, как именно эти блоки связываются между собой. Под функциональными блоками подразумевают функцию или группу функций, связанных по назначению или области применения в случае программы и группу функций/фрагментов информационного наполнения в случае сайта.



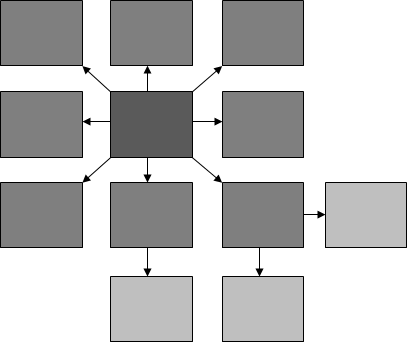
На рисунке выше изображена типичная структура сайта. В сайтах обычно функции размещаются в отдельных экранах.

Программы обычно имеют один изменяющийся экран, в котором вызываются почти все функции.

Это не догма.

Проектирование общей структуры состоит из двух параллельно происходящих процессов: выделения независимых блоков и определения связи между ними. Если проектируется сайт, в завершении необходимо также создать схему навигации.

Типичная структура программы изображена ниже



**Выделение независимых блоков**

Для этого этапа трудно дать какие-либо конкретные рекомендации, поскольку очень многое зависит от проектируемой системы. Тем не менее, можно c уверенностью рекомендовать избегать помещения в один блок более трех функций, поскольку каждый блок в результирующей системе будет заключен в отдельный экран или группу управляющих элементов. Перегружать же интерфейс опасно. Результатом этой работы должен быть список блоков с необходимыми пояснениями.

Существует три основных вида связи между блоками. Это логическая связь, связь по представлению пользователей и процессуальная связь.

Логическая связь определяет взаимодействие между фрагментами системы с точки зрения разработчика (суперпользователя).

Пользователи имеют свое мнение о системе, и это мнение тоже является важным видом связи.

Процессуальная связь описывает пусть не вполне логичное, но естественное для имеющегося процесса взаимодействие: например, логика напрямую не командует людям сначала мыть руки, а уж потом обедать, но обычно получается именно так.

Все три типа взаимосвязи должны быть заранее предусмотрены при конструировании системы.

**Логическая связь**

Она определяет взаимодействие между фрагментами системы с точки зрения разработчика. Полученные связи очень существенно влияют на навигацию в пределах системы (особенно, когда система многооконная). Поэтому чтобы не перегружать интерфейс, стоит избегать как слишком уж отдельных блоков (их трудно найти), так и блоков, связанных с большим количеством других.

На этом этапе выделяются объективные критерии оценки эргономичности интерфейса, такие как показатели эффективности, продуктивности, удовлетворенности пользователей (на более ранних этапах выделить эти критерии невозможно).

***Связь по представлению пользователей***

Пользователи имеют свое мнение о системе, и это мнение тоже является важным видом связи. В информационных системах, когда необходимо гарантировать, что пользователь найдет всю нужную ему информацию, необходимо устанавливать связи между блоками, основываясь не только на точке зрения разработчика, но и на представлениях пользователей. Дело в том, что самый распространенный способ поиска, а именно поиск по классификации признаков, работает только в том случае, когда пользователи согласны с принципами этой классификации.

Существует простой способ классификации – способ карточной сортировки. Все понятия, которые требуется классифицировать, пишутся на карточках из расчета «одно понятие – одна карточка». После чего группе пользователей из целевой аудитории предлагается эти карточки рассортировать (при этом каждый субъект получает свой набор карточек). Получившиеся кучки из карточек нужно разобрать на составляющие и свести результаты от разных субъектов в один способ классификации.

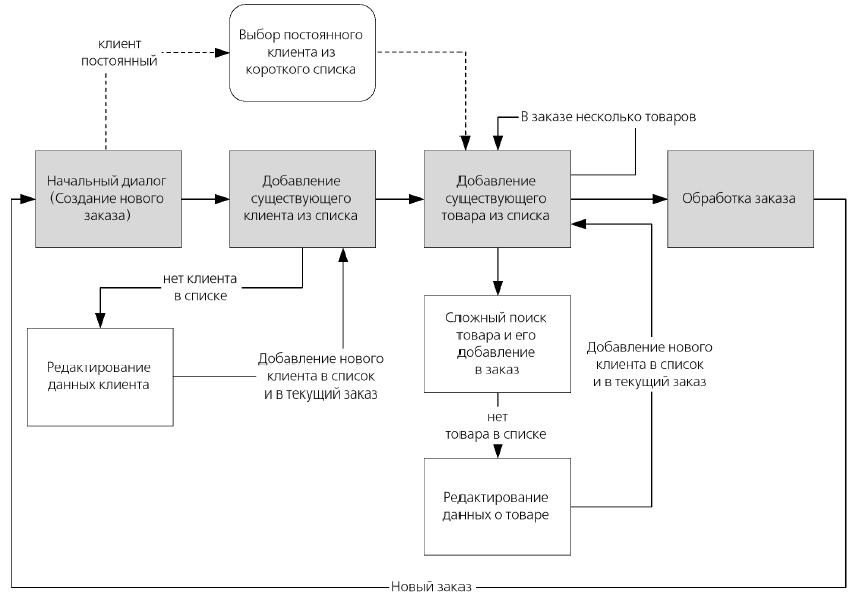
Ничего более работоспособного до сих пор человечеством не придумано. В то же время этот способ имеет определенные недостатки: во-первых, трудно заполучить на несколько часов представителей целевой аудитории, а во-вторых, при малом количестве субъектов результаты могут быть сомнительны (как минимум, нужно 4-5 человек).

**Процессуальная связь**

На этом этапе описывается не вполне логичное, но естественное для имеющегося процесса взаимодействие. Установление качественной процессуальной связи обычно довольно трудная задача, поскольку единственным источником информации является наблюдение за пользователем. В то же время установление такой связи дело исключительно полезное. Зачем, например, рисовать на экране сложную систему навигации, если точно известно, к какому блоку пользователь перейдет дальше? В этом смысле зачастую оправдано навязывать пользователю какую-либо процессуальную связь, жертвуя удобством, зато выигрывая в скорости обучения (поскольку пользователю приходится думать меньше). Жестко заданная связь позволяет также уменьшить количество ошибок. Классическим примером жестко заданной процессуальной связи является устройство мастеров, при котором пользователя заставляют нажимать кнопку «Далее».

Рассмотрим гипотетическую программу ввода данных. От пользователя требуется выбрать из списка клиента (или добавить в список нового) и указать, какие именно товары клиент заказал (товары в список тоже можно добавлять). Несколько клиентов постоянно что-то заказывают, так что заставлять пользователя каждый раз искать в списке такого клиента неправильно. При этом блоки разделяются следующим образом:

|  |  |
| --- | --- |
| **Основной экран между функциями системы** | **Навигация** |
| Создание нового заказа |  |
| Добавление существующего товара в заказ | Простой поиск товара в списке |
|  | Сложный поиск товара |
| Добавление нового товара в список |  |
| Добавление существующего клиента в заказ | Простой поиск клиента в списке |
| Добавление нового клиента в список |  |
| Выбор постоянного клиента |  |
| Обработка заказа Печать и его переход в папку *Исполняемые* |  |



Пример общей схемы (вторая версия). Прямоугольник обозначает отдельный экран, прямоугольник со скругленными углами – область экрана, пунктирная линия – альтернативное действие. Обратите внимание, что в этой схеме интерфейс заставляет пользователя выполнять задачу в сугубо определенной последовательности.

Существует любопытная закономерность: чем эстетически привлекательней выглядит схема (без учета цветового кодирования и веселеньких шрифтов), тем она эффективней. Всегда надо стараться сделать схему возможно более стройной и ясной.

На основе разработанной ранее структуры экранов на этом этапе выбирается наиболее адекватная **навигационная система** и разрабатывается её детальный интерфейс.

Навигационная система показывает механизм распределения функций и задач между окнами программы. Навигационная система определяет, каким образом пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи.

На входе: структура экранов.

На выходе: описание навигации по системе.

**При проектировании структуры справочной системы** разрабатывается справочная система (к настоящему этапу уже известна вся информация, необходимая для выработки структуры справочной системы, не просто описывающей интерфейс, но и помогающей пользователю решать его задачи).

На входе: информация из предыдущих этапов.

На выходе: описание примерной структуры и связей между темами справочной системы.

**При низкоуровневом проектировании** разрабатываются интерфейсы конкретных экранов системы (состав, взаимное расположение и поддерживающие текст интерфейсных элементов).

**При проектировании основных экранов** разрабатываются интерфейсы основных экранов системы.

На входе: информация их предыдущих этапов.

На выходе: полное описание интерфейса главных экранов системы (без обработки исключительных ситуаций). К отчету прилагаются макеты экранов с описаниями функциональности каждого интерфейсного элемента.

На основе объективных критериев успеха интерфейса и сценариев действий пользователей разрабатываются **тестовые задания**, которые выполняются пользователями с фиксацией всех значимых характеристик деятельности (таких как производительность труда, количество человеческих ошибок). После этого выполняется подсчет соответствующих показателей и сравнение их с заданными. На основании полученных данных интерфейс либо дорабатывается, либо считается разработанным.

На входе: описание интерфейса экранов системы, доступ к пользователям.

На выходе: краткие протоколы тестирования, выводы из полученных данных, числовые значения по выбранным параметрам.

**При проектировании второстепенных экранов**

разрабатываются интерфейсы второстепенных экранов системы. К ним относятся диалоговые окна и всевозможные сообщения.

На входе: информация из предыдущих этапов.

На выходе: полное описание интерфейса оставшихся экранов системы и описание обработки исключительных ситуаций, влияющих на интерфейс. К отчету прилагаются макеты экранов с описаниями функциональности каждого интерфейсного элемента.

**При финальном тестировании** на основе объективных критериев успеха интерфейса и сценариев действий пользователей разрабатываются оставшиеся тестовые задания, которые выполняются пользователями с фиксацией всех значимых характеристик их деятельности. После этого выполняется подсчет соответствующих показателей и сравнение их с заданными. На основании полученных данных интерфейс либо дорабатывается, либо считается разработанным.

На входе: описание интерфейса экранов системы, доступ к пользователям.

На выходе: краткие протоколы тестирования, выводы из полученных данных, числовые значения по выбранным параметрам, глоссарий.

Создание максимально эффективного **прототипа интерфейса** является чрезвычайно важной задачей. Прототип должен хорошо выглядеть, чтобы понравиться заказчику и не вызвать вопросов у субъектов тестирования, он должен быть максимально дешев, максимально полон и, что немаловажно, должен с легкостью обновляться.

При создании прототипа наиболее частой ошибкой является чрезмерное наведение глянца и вообще стремление сделать прототип возможно более похожим на результирующую систему. В самом таком подходе нет ничего плохого (всё равно определенные части прототипа приходится делать максимально совершенными), проблема в том, что в большинстве случаев прототип после тестирования оказывается неправильным. Его приходится переделывать, причем иногда полностью, при этом все вложенные в прототип ресурсы (в основном время) оказываются выброшенными на ветер.