

Билет 1

1. Принципы, относящиеся к восприятию

1. Сделайте дисплей

чётким. Читаемость дисплея является важным критерием в проектировании дисплея. Если символы или объекты отображаются нечётко, то пользователь не может эффективно их использовать.

2. Избегайте абсолютно строгих границ.

Не просите пользователя определить уровень переменной на основе лишь одной сенсорной переменной (например цвет, размер, громкость). Эти сенсорные переменные могут содержать множество различных уровней.

3. Обработка сверху-вниз.

Сигналы воспринимаются и толкуются в соответствии с ожиданиями, сформированными на основе более раннего опыта пользователя. Если сигнал представлен вопреки ожиданиям пользователя, то потребуются больше его представления, чтобы доказать, что сигнал был понят верно.

4. *Чрезмерная выгода.* Если сигнал представлен более одного раза, больше шансов, что он будет понят верно. Это возможно сделать с помощью представления его в альтернативных физических формах (например, цвета, форма, голос и т. д.), так как избыточность не подразумевает повторения. Светофор является прекрасным примером избыточности, так цвет и положение являются избыточными.

5. *Сходства приводят к путанице.* Используйте отличающиеся элементы. Похожие сигналы будут приводить к путанице. Соотношение схожих признаков к различным признакам является причиной схожести сигналов. Например, A423B9 больше похоже на A423B8, чем 92 на 93. Ненужные похожие признаки должны быть удалены, а

непохожие признаки должны быть выделены.

6. *Принцип изобразительного реализма.* Экран должен выглядеть как переменная, которую он представляет (например, высокая температура на термометре показана высшим вертикальным уровнем). Если есть множество составляющих, то они могут быть настроены так, как они будут выглядеть в среде, где они будут представлены.

7. *Принцип движущейся части.* Движущиеся элементы должны двигаться по той схеме и в том направлении в каком это происходит в мысленном представлении пользователя, как оно движется в системе. Например, движущийся элемент на высотометре должен двигаться вверх с набиранием высоты.

2. Метрики предпочтений, производительности, предсказывающие Метрики предпочтений

Один из наиболее простых и популярных методов оценки практичности заключается в использовании метрик предпочтений. Технология такова: берете несколько ничего не подозревающих субъектов — пользователей или потенциальных пользователей — и спрашиваете их, что они думают о данной системе. Естественно, у субъективной пользовательской оценки есть как плюсы, так и минусы. Плюсы — в относительной дешевизне и простоте получения информации, минусы — в том, что не всегда именно эта информация требуется разработчикам. Оценка пользователей не только субъективна сама по себе, но и пользовательские предпочтения весьма относительно коррелируют с реальной простотой использования и другими стандартными показателями практичности. Пользователи будут говорить вам про невиданное удобство использования и указывать при этом на запутанные или

неэффективные системы. Попробуйте спросить, что они думают по поводу Web-сайтов, и в их ответе вы найдете подтверждение этим словам. Тем не менее субъективные впечатления конечных потребителей продукта игнорировать нельзя. Удовлетворение пользователя — это, несомненно, одна из составляющих практичности и важный фактор успеха на рынке. Ведь в самом деле, удовлетворение пользователя лучше подскажет, какой продукт «пойдет» на рынке, чем практичность сама по себе.

Метрики производительности

Некоторые вопросы удастся решить, только поработав с реально функционирующей системой; соответственно, в задачи тестирования практичности входит создание условий, достаточно реалистично моделирующих такую работу. Мера «достаточности» в данном случае определяется мерой искажения ответов пользователей. Метрики производительности служат показателями того, как пользователи выполняют задания с помощью системы в лабораториях тестирования практичности либо на своих рабочих местах. Можно измерять множество аспектов производительности, например время, затрачиваемое на решение задачи или нескольких задач, количество ошибок, частоту обращения к справочной системе.

Предсказывающие метрики

Метрики проектирования являются объективными показателями качества, которые можно получить, используя элементы проекта, такие как визуальный дизайн экранных решений. Эти метрики называются предсказывающими, поскольку они позволяют предположить или предсказать некоторые аспекты реальной производительности уже построенной системы.

Корректные метрики проектирования будут достаточно точно коррелировать с реальной простотой использования, эффективностью, уровнем ошибки другими показателями прикладной практичности. Разработчики ПО могут рассматривать метрики проектирования как альтернативу субъективным пользовательским оценкам или как тестирование практичности, проводимое уже после создания функционального прототипа или системы. Большим преимуществом метрик проектирования является то, что они позволяют быстро и дешево оценивать и сравнивать разные проекты. При этом не требуется заранее специально для тестирования разрабатывать систему, модель или рабочий прототип.

Билет 2

1. Принципы, основанные на внимании, Принципы умозрительной модели

1. *Принцип изобразительного реализма.* Экран должен выглядеть как переменная, которую он представляет (например, высокая температура на термометре показана высшим вертикальным уровнем). Если есть множество составляющих, то они могут быть настроены так, как они будут выглядеть в среде, где они будут представлены.

2. *Принцип движущейся части.* Движущиеся элементы должны двигаться по той схеме и в том направлении в каком это происходит в мысленном представлении пользователя, как оно движется в системе. Например, движущийся элемент на высотомере должен двигаться вверх с набиранием высоты.

3. *Минимизация времени доступа к информации.* Когда внимание пользователя перемещается из одного места в другое в целях доступа к необходимой информации, то затрачивается много времени и усилий. Конструкция дисплея должна уменьшить данные затраты, так часто используемый источник должен находиться в ближайшей позиции. Однако не должна быть утеряна понятность.

4. *Принцип совместимости.* Разделённое внимание между двумя источниками может быть необходимо для выполнения одной задачи. Эти источники должны быть мысленно взаимосвязаны и иметь мысленную близость. Время доступа к информации должно быть небольшим и это может быть достигнуто различными способами (например, близкое расположение, одинаковый цвет, узоры, формы и т. д.). Однако, близость отображения может привести к путанице.

5. *Принцип большого количества ресурсов.* Пользователь может более просто обрабатывать

информацию с разных ресурсов. Например, зрительная и слуховая информация может быть представлена одновременно, чем представлять всю зрительную и всю аудио информацию.

2. Модель GOMS

Модель GOMS (расшифровывается как "thetmodelofgoals, objects, methods, andselectionrules") позволяет предсказать время, необходимое для выполнения задачи с помощью конкретного интерфейса. Разработчики этой модели обнаружили, что для решения поставленной задачи при работе с компьютером пользователь совершает определенный набор жестов. Таким образом, время выполнения задачи складывается из времени выполнения этих жестов и из промежутков между жестами, которые требуются для обдумывания.

В результате тщательного лабораторного исследования удалось получить среднее время выполнения различных жестов:

$K = 0.2$ с - время, необходимое для нажатия клавиши клавиатуры

$P = 1.1$ с - время, необходимое для перемещения указателя мыши к определенной позиции на мониторе

$H = 0.4$ - время, необходимое для перемещения руки пользователя с клавиатуры на мышь

$M = 1.35$ - ментальная пауза, т.е. время, необходимое пользователю на обдумывание следующего шага.

Билет 3

1. Закон Хика

Время реакции при выборе из некоторого числа альтернативных сигналов зависит от их числа.

Закон Хика — утверждение, что время реакции при выборе из некоторого числа альтернативных сигналов зависит от их числа. Впервые эта закономерность была получена в 1885 г. немецким психологом И. Меркелем, а в 1952 г. получила экспериментальное подтверждение в исследованиях В.Э. Хика, в которых она приобрела вид логарифмической функции:

$BP = a \cdot \log(n+1)$, где BP — среднее значение времени реакции по всем альтернативным сигналам; n — число равновероятных альтернативных сигналов; a — коэффициент пропорциональности. Единица в формулу введена для учета еще одной альтернативы, в виде пропуска сигнала.

2. Принципы памяти

1. Замените память наглядной информацией: мировое знание. Пользователь не должен сохранять важную информацию исключительно в рабочей памяти или извлекать её из долговременной памяти. Меню/перечень могут помочь пользователю упростить использование памяти. Однако, использование памяти иногда может помочь пользователю, так как избавляет от необходимости ссылаться на некоторые типы знаний в мире (например, компьютерный специалист скорее использовал бы прямые команды из памяти, чем обращался к руководству). Для эффективной разработки должны быть сбалансированы знания в голове пользователя и знания в мире.

2. Принцип предиктивной помощи. Проактивные действия, как правило, более эффективны, чем реактивные действия. Дисплей должен исключать ресурсоёмкие когнитивные задачи и заменить их более простыми задачами, чтобы

сократить использование умственных ресурсов пользователя. Это позволит пользователю сконцентрироваться не только на текущей ситуации, но и также подумать о возможных ситуациях в будущем. Пример предиктивной помощи — дорожный знак, который информирует о расстоянии до пункта назначения.

3. Принцип

совместимости. Старые особенности других дисплеев легко перенести в разработку новых дисплеев, если их разработки совместимы. Долговременная память пользователя будет срабатывать на выполнение уместных действий. В ходе разработки должен быть принят во внимание данный факт и учитывать совместимость между разными дисплеями.

Билет 4

1. Когнитивная психология.

Когнитивное сознательное, когнитивное бессознательное

При проектировании пользовательских интерфейсов необходимо учитывать психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации.

Исследованием принципов работы мозга человека занимается когнитивная психология.

Информация о внешнем мире поступает в наш мозг в огромных количествах. Часть мозга, которую условно можно назвать «процессором восприятия», постоянно без участия сознания перерабатывает ее, сравнивая с прошлым опытом, и помещает в хранилище уже в виде зрительных, звуковых и прочих образов. Любые внезапные или просто значимые для нас изменения в окружении привлекают наше внимание, и тогда интересующая нас информация поступает в кратковременную память. Если же наше внимание не было привлечено, то информация в хранилище пропадает, замещаясь следующими порциями.

Когнитивное бессознательное — это те ментальные процессы, которые вы не осознаете в тот момент, когда они происходят.

Когнитивное сознательное включается в тот момент, когда вы сталкиваетесь с ситуацией, которая кажется новой или представляет угрозу, или, когда требуется принять нешаблонное решение.

2. Методология исследования удобства использования на основе фокус группы

Фокус-группа — это метод исследования, при котором группа пользователей приглашается для принятия участия в тестировании пользовательского интерфейса. Как правило, пользователи,

которым предлагается оценить удобство интерфейса, являются потенциальными клиентами.

Проведение интервью с пользователями ведется по заранее подготовленному сценарию. Пользователю ставятся вопросы либо задачи, которые ему необходимо выполнить при работе с интерфейсом приложения. Все действия пользователя, реакция на определенные элементы интерфейса, его реплики и эмоции записываются на аудио и видео носители для дальнейшего анализа.

Исследования этого типа включают четыре общих элемента:

1. Вовлечение нескольких респондентов, собранных в одном месте.
2. Взаимодействие участников. Если во многих других типах исследований считается, что любая дискуссия между участниками искажает чистоту ответов, то на заседаниях фокус-групп субъекты поощряются к взаимодействию друг с другом.
3. Весь ход обсуждения осуществляется профессионалом-модератором. Он направляет течение групповой дискуссии в соответствии с целями, поставленными на предварительной стадии.
4. При проведении фокус-групп используется сценарий. Если в количественном исследовании при сборе информации применяется законченный, формализованный, структурированный инструментарий, то путеводитель обычно имеет форму относительно незавершенного руководства. Его основное назначение — фокусирование проблемы, настраивание на определенную тему. Одновременно он должен давать возможность спонтанным высказываниям участников, обеспечивать групповую динамику.

Билет 5

1. Локус внимания, одновременное выполнение задач

Локус внимания — это некоторое место или область, на которое может быть сосредоточено ваше внимание. В отличие от фокуса, часто обозначающего не только место, но и действие (сфокусировать ваше внимание), локус обозначает только место и переводится с латинского, как место положения или область.

В каждый момент времени фокус внимания может фиксироваться в одной точке. Поэтому, если возникает необходимость «одновременно» отслеживать несколько ситуаций, то обычно фокус перемещается с одного отслеживаемого элемента на другой. При этом внимание «рассредоточивается», и какие-то детали могут быть упущены.

Обработка процессором восприятия требует некоторого времени и, если сигнал выдается в течение времени, меньшем времени обработки, то наш мозг его не воспринимает.

Восприятие во многом основано на мотивации. Например, если человек голоден, то он в первую очередь будет замечать все съедобное, а если устал — то, войдя в комнату, он в первую очередь увидит диван или кровать.

В процессе переработки информации мозг сравнивает поступающие данные с предыдущими.

При смене кадра мозг на некоторое время блокируется: он «осваивает» новую картинку, выделяя наиболее существенные детали. А значит, если необходима быстрая реакция пользователя, то резко менять картину не стоит.

2. Диаграмма взаимодействий

Диаграммы последовательностей - это отличное средство документирования поведения системы, детализации логики сценариев использования; но есть

еще один способ - использовать диаграммы взаимодействия.

Диаграмма взаимодействия *показывает поток сообщений между объектами системы и основные ассоциации между ними* и по сути, как уже было сказано выше, является альтернативой диаграммы последовательностей. Внимательный читатель, возможно, скажет, что *диаграмма объектов* делает то же самое, - и будет не прав. *Диаграмма объектов показывает статику*, некий снимок системы, связи между объектами в данный момент времени, диаграмма же взаимодействия, как и диаграмма последовательностей, показывает взаимодействие (извините за невольный каламбур) объектов во времени, т. е. в динамике.

Следует отметить, что использование диаграммы последовательностей или диаграммы взаимодействия - личный выбор каждого проектировщика и зависит от индивидуального стиля проектирования. Мы, например, чаще отдаем предпочтение диаграмме последовательностей. На обозначениях, применяемых на диаграмме взаимодействия, думаем, не стоит останавливаться подробно. Здесь все стандартно: объекты обозначаются прямоугольниками с подчеркнутыми именами (чтобы отличить их от классов, помните?), ассоциации между объектами указываются в виде соединяющих их линий, над ними может быть изображена стрелка с указанием названия сообщения и его порядкового номера.

Необходимость номера сообщения объясняется очень просто - в отличие от диаграммы последовательностей, *время на диаграмме взаимодействия не показывается в виде отдельного измерения*. Поэтому последовательность передачи сообщений можно указать только с помощью их нумерации. В этом и состоит вероятная причина пренебрежения этим видом

диаграмм многими проектировщиками.

Билет 6

1. Диаграмма классов

Диаграмма классов (англ. StaticStructuredDiagram) — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Входит в UML.

Существует два вида:

- Статический вид диаграммы рассматривает логические взаимосвязи классов между собой;
- Аналитический вид диаграммы рассматривает общий вид и взаимосвязи классов, входящих в систему.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

- Концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
- Точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
- Точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

- В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом.
- Посередине располагаются поля (атрибуты) класса. Они

выровнены по левому краю и начинаются с маленькой буквы.

- Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы.

2. Формирование привычек

По мере повторения — или с практикой — выполнение того или иного действия становится для вас привычным, и вы можете выполнять его не задумываясь. Печатать на машинке слепым методом, так же, как и ездить на велосипеде или ходить пешком по тропинке, лучше всего получается, если об этом не задумываться.

Когда вы выполняете какую-то задачу многократно, то с каждым разом делать это становится всё проще.

По мере повторения — или с практикой — выполнение того или иного действия становится для вас привычным, и вы можете выполнять его не задумываясь.

При постоянном использовании какого-либо интерфейса у вас формируются определенные привычки, которые впоследствии трудно преодолеть. В этом смысле задача дизайнеров заключается в том, чтобы создавать интерфейсы, которые не позволяют привычкам вызывать проблемы у пользователей.

Билет 7

1. Перцептивная память

Перцептивные действия — структурные единицы процесса восприятия у человека. Перцептивные действия обеспечивают сознательное выделение того или иного аспекта чувственно заданной ситуации, а также преобразование сенсорной информации, приводящее к построению образа, адекватного предметному миру и задачам деятельности.

Большинство восприятий утрачиваются после того, как затухают.

С точки зрения разработки интерфейсов из быстрого затухания сенсорных восприятий следует, что человек, прочитавший или услышавший 5 секунд назад некоторое сообщение, необязательно сможет вспомнить его содержание.

Если такое сообщение важно само по себе или содержит важную деталь, то оно должно оставаться на экране до тех пор, пока не перестанет быть актуальным.

2. Гибкость интерфейса

Гибкость интерфейса — это способность самонастраивания интерфейса, который учитывает уровень подготовки и производительность труда пользователя.

Полностью гибких интерфейсов не существует, но элементы гибкости должны присутствовать.

Концепция гибкого интерфейса в настоящее время является одной из основных областей исследования взаимодействия человека и ЭВМ.

Основная проблема состоит не в том, как организовать изменения в диалоге, а в том, какие признаки нужно использовать для определения необходимости внесения изменения и их сути.

Билет 8

1. Естественность интерфейса

Естественный интерфейс - интерфейс, который не вынуждает пользователя существенно изменять привычные для него способы решения задачи. Это, в частности, означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений. Целесообразно также сохранить систему обозначений и терминологию, используемые в данной предметной области.

Использование знакомых пользователю понятий и образов (метафор) обеспечивает интуитивно понятный интерфейс при выполнении его заданий. Вместе с тем, использование метафоры не должно ограничивать их машинную реализацию полной аналогией с одноименными объектами реального мира. Например, в отличие от своего бумажного аналога, папка на Рабочем столе Windows может использоваться для хранения разных объектов (таких, например, как принтеры, калькуляторы, другие папки). Метафоры являются своего рода "мостиком", связывающим образы реального мира с теми действиями и объектами, которыми приходится манипулировать пользователю при его работе на компьютере; они обеспечивают "узнавание", а не "вспоминание". Пользователи запоминают действие, связанное со знакомым объектом, более легко, чем они запомнили бы имя команды, связанной с этим действием.

2. UML в общем (все типы диаграмм, основная терминология – актер, отношение и т.д.)

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-

процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

В UML используются следующие виды диаграмм:

Структурные диаграммы:

- Диаграмма классов
- Диаграмма компонентов
- Диаграмма композитной/составной структуры
 - Диаграмма кооперации (UML2.0)
- Диаграмма развёртывания
- Диаграмма объектов
- Диаграмма пакетов
- Диаграмма профилей (UML2.2)

Диаграммы поведения:

- Диаграмма деятельности
- Диаграмма состояний
- Диаграмма вариантов использования

Диаграммы взаимодействия:

- Диаграмма коммуникации (UML2.0) / Диаграмма кооперации (UML1.x)
- Диаграмма обзора взаимодействия (UML2.0)
- Диаграмма последовательности
- Диаграмма синхронизации (UML2.0)

Билет 9

1. Согласованность интерфейса

Согласованность позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на выяснение различий в использовании тех или иных элементов управления, команд и т.д. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым.

Согласованность важна для всех аспектов интерфейса, включая имена команд, визуальное представление информации и поведение интерактивных элементов. Для реализации свойства согласованности в создаваемом программном обеспечении необходимо учитывать его различные аспекты.

1. Согласованность в пределах продукта. Одна и та же команда должна выполнять одни и те же функции, где бы она ни встретилась, причем одним и тем же образом. Например, если в одном диалоговом окне команда копировать означает немедленное выполнения соответствующих действий, то в другом окне она не должна требовать от пользователя дополнительно указать расположение копируемой информации. Другими словами, нужно использовать одну и ту же команду, чтобы выполнить функции, которые кажутся подобными пользователю.

2. Согласованность в пределах рабочей среды. Поддерживая согласованность с интерфейсом, предоставляемым операционной системой (например, ОС Windows), приложение может "опираться" на те знания и навыки пользователя, которые он получил ранее при работе с другими приложениями.

3. Согласованность в использовании метафор. Если

поведение некоторого программного объекта выходит за рамки того, что обычно подразумевается под соответствующей ему метафорой, у пользователя могут возникнуть трудности при работе с таким объектом. Например, если для программного объекта Корзина определить операцию Запуск, то для выяснения ее смысла пользователю, скорее всего, потребуется посторонняя помощь.

2. Use case диаграмма

Диаграмма прецедентов (диаграмма вариантов использования) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью *модели прецедентов*, позволяющей описать систему на концептуальном уровне^[1].

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному МУ пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

- чётко отделить систему от её окружения;
- определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую

функциональность системы;

- определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть, прецедентов).

Работа над диаграммой может начинаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ)

Для отражения модели прецедентов на диаграмме используются:

- рамки системы (англ. *system boundary*) — прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации,
- актёр (англ. *actor*) — стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующего с некоторой сущностью (системой, подсистемой, классом). Актёры не могут быть связаны друг с другом (за исключением отношений обобщения/наследования),
- прецедент — эллипс с надписью, обозначающий выполняемые системой действия (могут включать возможные варианты), приводящие к наблюдаемым актёрами результатам. Надпись может быть именем или описанием (с точки зрения актёров) того, «что» делает

система (а не «как»).

Имя прецедента связано с непрерываемым (атомарным) *сценарием* — конкретной последовательностью действий, иллюстрирующей поведение^[2]. В ходе сценария актёры обмениваются с системой сообщениями. Сценарий может быть приведён на диаграмме прецедентов в виде UML-комментария. С одним прецедентом может быть связано несколько различных сценариев.

соответствующему результату.

Отношения между прецедентами

Часть дублирующейся информации в модели прецедентов можно устранить указанием связей между прецедентами:

- *обобщение прецедента* — стрелка с незакрашенным треугольником (треугольник ставится у более общего прецедента),
- *включение прецедента* — пунктирная стрелка со стереотипом «include»,
- *расширение прецедента* — пунктирная стрелка со стереотипом «extend» (стрелка входит в расширяемый прецедент, в дополнительном разделе которого может быть указана *точка расширения* и, возможно в виде комментария, условие расширения).

При работе с вариантами использования важно помнить несколько простых правил:

- каждый прецедент относится как минимум к одному действующему лицу;
- каждый прецедент имеет инициатора;
- каждый прецедент приводит к

Билет 10

1. Дружественность интерфейса

Дружественность интерфейса (принцип "прощения" пользователя)

Пользователи обычно изучают особенности работы с новым программным продуктом методом проб и ошибок. Эффективный интерфейс должен принимать во внимание такой подход. На каждом этапе работы он должен разрешать только соответствующий набор действий и предупреждать пользователей о тех ситуациях, где они могут повредить системе или данным, еще лучше, если у пользователя существует возможность отменить или исправить выполненные действия. Даже при наличии хорошо спроектированного интерфейса пользователи могут делать те или иные ошибки. Эти ошибки могут быть как "физического" типа (случайный выбор неправильной команды или данных), так и "логического" (принятие неправильного решения на выбор команды или данных). Эффективный интерфейс должен позволять предотвращать ситуации, которые, вероятно, закончатся ошибками. Он также должен уметь адаптироваться к потенциальным ошибкам пользователя и облегчать ему процесс устранения последствий таких ошибок.

2. Диаграмма последовательностей

Диаграмма последовательности (*англ. sequencediagram*) — диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов). Используется в языке UML.

Основными элементами диаграммы последовательности

являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (*англ. lifeline*), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Билет 11

1. Принцип «обратной связи»

Дизайн должен информировать пользователей о выполняемых действиях, изменениях состояния или условий, об ошибках или исключениях. Эта информация должна быть актуальна и интересна пользователю и представлена в четкой, компактной и недвусмысленной форме.

2. Модель Фитса

$$T = \log_2 \left(\frac{D}{W} + 1 \right)$$
 где T — среднее время, затрачиваемое на совершение действия, a — время запуска/остановки устройства, b — величина, зависящая от типичной скорости устройства, D — дистанция от точки старта до центра объекта, W — ширина объекта, измеренная вдоль оси движения.

Главным образом это означает, что время, затрачиваемое на достижение цели, является функцией расстояния и размера цели. На первый взгляд это кажется очевидным: чем дальше мы от цели и чем меньше она по размеру, тем больше времени потребуется для позиционирования.

Билет 12

1. Простота интерфейса

Интерфейс должен быть простым.

При этом имеется в виду не упрощенность, а обеспечение легкости в его изучении и в использовании. Кроме того, он должен предоставлять доступ ко всему перечню функциональных возможностей, предусмотренных данным приложением.

Реализация доступа к широким функциональным возможностям и обеспечение простоты работы противоречат друг другу. Разработка эффективного интерфейса призвана сбалансировать эти цели.

Один из возможных путей поддержания простоты - представление на экране информации, минимально необходимой для выполнения пользователем очередного шага задания. В частности, нужно избегать многословных командных имен или сообщений. Непродуманные или избыточные фразы затрудняют пользователю извлечение существенной информации.

Другой путь к созданию простого, но эффективного интерфейса - размещение и представление элементов на экране с учетом их смыслового значения и логической взаимосвязи. Это позволяет использовать в процессе работы ассоциативное мышление пользователя.

Можно также помочь пользователям управлять сложностью отображаемой информации, используя последовательное раскрытие (диалоговых окон, разделов меню и т.д.). Последовательное раскрытие предполагает такую организацию информации, при которой в каждый момент времени на экране находится только та ее часть, которая необходима для выполнения очередного шага. Сокращая объем информации, представленной пользователю, вы тем самым уменьшаете объем информации, подлежащей обработке. Примером такой

организации является иерархическое (каскадное) меню, каждый уровень которого отображает только те пункты, которые соответствуют одному, выбранному пользователем, пункту более высокого уровня.

2. Eyetracking

Окулография (отслеживание глаз, трекинг глаз; айтрекинг) — определение координат взора («точки пересечения оптической оси глазного яблока и плоскости наблюдаемого объекта или экрана, на котором предъявляется некоторый зрительный раздражитель»).

Отслеживатель глаз — устройство, используемое для определения ориентации оптической оси глазного яблока в пространстве (то есть для отслеживания глаз). Отслеживатели глаз используются в исследованиях зрительной системы, психологии, когнитивной лингвистике. Для отслеживания глаз используется несколько методов. Самый популярный — покадровый анализ видеосъемки глаза, также используются контактные методы, такие как электроокулография.

Данная технология позволит вам определить:

Насколько удобна и понятна система навигации вашего сайта

Насколько эффективна ваша банерная реклама

Какие элементы сайта привлекают наибольшее внимание, а какие — остаются незаметными

Порядок действий от начала сеанса до совершения покупки

Эмоциональную реакцию посетителя на тот или иной элемент сайта (на основе измерения диаметра зрачка)

Билет 13

1. Эстетическая привлекательность

Внешний вид самого приложения и способа его представления (вплоть до упаковки) позволяет сформировать у потребителя положительное мнение о программе. Однако эстетические характеристики весьма субъективны и описать их количественно гораздо труднее, чем функциональные требования или показатели производительности. Вся эстетика приложения зачастую сводится к простому выбору: соотносятся ли между собой используемые цвета, передают ли элементы интерфейса их назначение и смысл представляемых операций, что ощущает человек при использовании тех или иных элементов управления и насколько успешно он их использует.

Принцип стилистической целостности: Стилистическая целостность пользовательского интерфейса достигается единообразием оформления, под которым подразумеваются в выборе гарнитур и размеров шрифта, а также равномерное распределение свободного пространства.

Принцип выравнивания: Ни один элемент пользовательского интерфейса не должен располагаться случайным образом, т.е., нельзя располагать элементы или подсистемы где-либо только потому, что там есть свободное пространство.

Принцип повтора: Некоторые компоненты стилового оформления должны повторяться, что делает пользовательский интерфейс более цельным, согласованным и упорядоченным. Повторяться могут маркеры списков, шрифтовое оформление, цветовые решения, пространственные взаимосвязи, толщина соединительных линий, распределение свободного пространства и т.п. При отсутствии повторяющихся элементов, каждая страница лишится

связанности и стилового единства. Но необходимо избегать избыточного повтора, который будет подавлять смысловое содержание.

Принцип визуальной иерархии: Элементы, являющиеся частями друг друга, представляются в виде вложений. Необходимо организовать внешний вид элементов таким образом, чтобы пользовательский интерфейс ясно и четко отображал отношения между элементами — какие элементы связаны между собой, какие являются частями других элементов (подсистем). Равнозначные элементы необходимо обозначать одинаковыми графическими объектами.

2. Модель Фитса

$$T = \log_2 \left(\frac{D}{W} + 1 \right)$$
 где T — среднее время, затрачиваемое на совершение действия, a — время запуска/остановки устройства, b — величина, зависящая от типичной скорости устройства, D — дистанция от точки старта до центра объекта, W — ширина объекта, измеренная вдоль оси движения.

Главным образом это означает, что время, затрачиваемое на достижение цели, является функцией расстояния и размера цели. На первый взгляд это кажется очевидным: чем дальше мы от цели и чем меньше она по размеру, тем больше времени потребуется для позиционирования.