

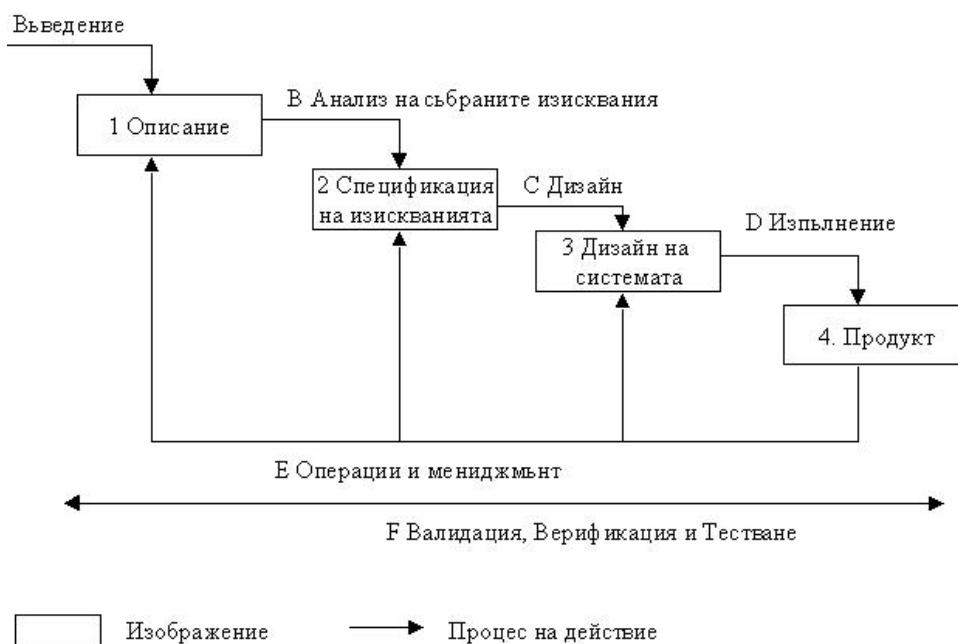
ОСНОВНИ МОДЕЛИ И МЕТОДИ ПРИ СЪЗДАВАНЕ НА ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС: ПОДХОДИ, ОСНОВНИ ПРОЦЕСИ, АНАЛИЗ НА ЗАДАЧИТЕ, СПЕЦИФИЦИРАНЕ НА ВЗАИМОДЕЙСТВИЯТА, ОСНОВНИ ТЕХНИКИ И ИНСТРУМЕНТАЛНИ СРЕДСТВА.

За да се изгради някой продукт, трябва да се извършат две главни дейности: дизайнерът трябва да разтълкува потребителските **изисквания** за продукта и да **развие** самият продукт. Тълкуването на изискванията включва разглеждане на подобни продукти, разглеждане на нуждите на хората, които ще ползват този продукт и анализиране на съществуващи системи, за да се открият проблемите на настоящия дизайн. Развитието може да включва представяне на различни образци, докато се получи удобен артефакт.

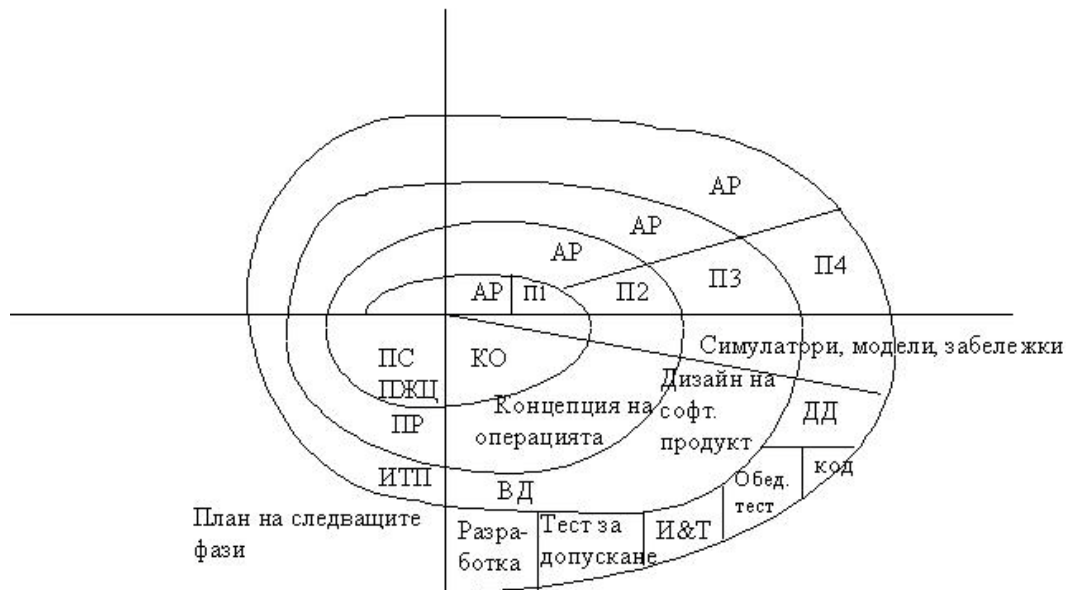
ОПРЕДЕЛЯНЕ НА ИЗИСКВАНИЯТА

Определяне на изискванията или **анализиране** се нарича процесът на откриване на това какво клиента изисква от една софтуерна система. Обикновено, документът, който се предоставя от клиента описва какво се иска от системата, но в доста неясни термини. Едно от основните предназначения на процеса на определяне на изискванията е да изясни нуждите на клиента и да определи "невидими" изисквания, пропуски и неясноти.

Изискванията към една система могат да бъдат разделени в три категории. **Функционалните изисквания** специфицират това, което системат трябва да прави. **Изискванията към данните** специфицират структурата на системата и данните необходими за успешното протичане на процеса. **Изискванията за използваемост** специфицират приемливото ниво на потребителска работа и удовлетворение от системата.



Приложение 1. Каскаден модел на софтуерното развитие



Приложение 2. Спирален модел на Гоем

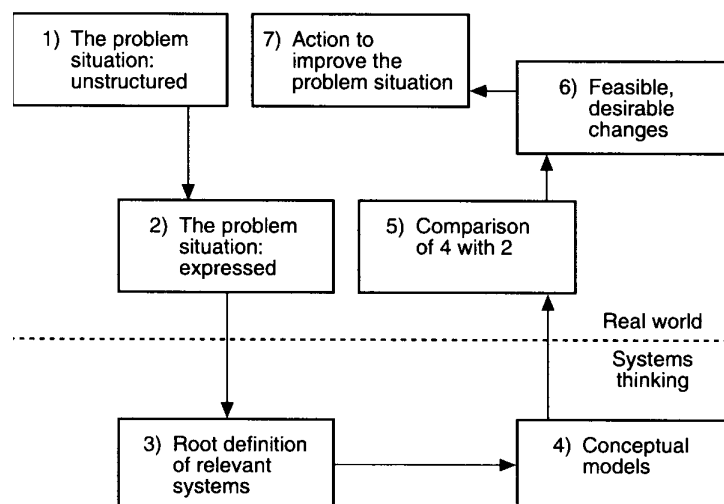
МЕТОДИ ЗА ПОТРЕБИТЕЛСКИ-ЦЕНТРАЛИЗИРАН ДИЗАЙН. МЕТОДОЛОГИИ

ВАЖНОСТТА ОТ РАЗБИРАНЕТО НА ЦЯЛАТА ЧОВЕКО-МАШИННА СИСТЕМА

ПЧМИ методите търсят начин да въведат потребителите във всички аспекти на системното разработване. Основните принципи на потребителско-централизирания дизайн поставят потребителите на централно място в процеса на проектиране, провеждайки ранни тестове, оценяване с потребителите и итеративно проектиране.

СОФТУЕРНО СИСТЕМНА МЕТОДОЛОГИЯ (SOFT SYSTEM METHODOLOGY)

SSM не набляга на намирането на разрешение на определен проблем, а на разбирането на ситуацията, в която проблемът е възникнал. Фокусът е установен над целите, хората, "световния възглед" на човеко-машинната система и на разработването на концептуални модели за тази идеална система.

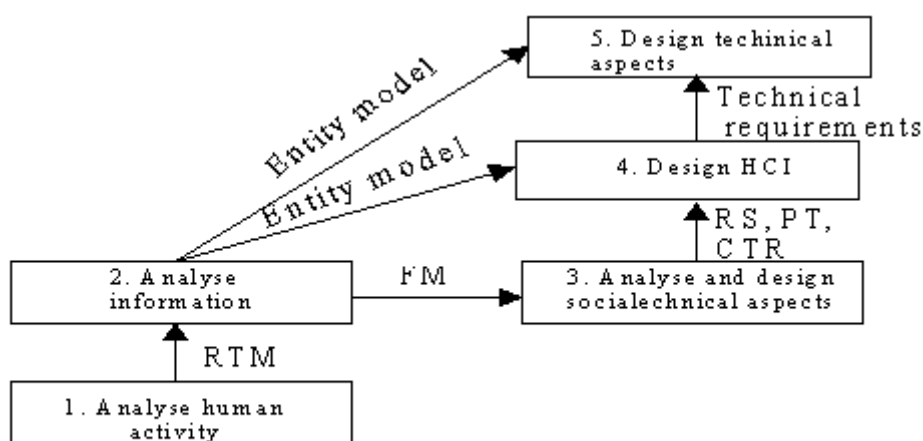


Етапи 1 и 2 на метода се отнасят към даването на пълно изложение на проблемната ситуация. По време на тези етапи се включват и "участниците". Често различните участници имат различно виждане; могат да наблягат върху различни аспекти на цялостната ситуация, но трябва да се **съгласуват** помежду си.

Етап 3 дава точна дефиниция на системата. Тя се използва (в етап 4) за изработването на **концептуални модели - абстрактни представяния на системата**. Един от важните аспекти на подхода е, че етапите 3 и 4 се изпълняват далеч от реалния свят - това позволява на презентацията да се представи без да се ограничава в използването на наистина реални думи. Концептуалният модел на ситуацията се сравнява с оригиналното изложение (етап 5) на ситуацията и в резултат се излагат възможните и желателни промени (етап 6). Етап 7 може да покаже резултатите в по-нататъшните итерации на процеса - новия проблем "какво е необходимото действие" може да бъде анализирано използвайки SSM подхода.

MULTIVIEW: ПОТЕБИТЕЛСКО - ЦЕНТРАЛИЗИРАНА МЕТОДОЛОГИЯ

Комбинира социотехническата и софтуерна система в по-обхващаща ги методология. **Multiview** (Avison and Wool-Harper, 1990) е **информационна система** проектираща **методология, която комбинира различните подходи в една платформа**.



Multiview methodology (Avison and Wood-Harper, 1990)

Първият етап определя **целите на системата**, включените **участници** и **перспективите** на собствениците на системата. Етап **2**, **"анализ на информацията"** се отнася до **концептуалното моделиране на потоквата информация** и информационната структура. **Функционалният модел (FM)** се използва в етап **3** като основа за **определение на задачите**: задачи на проектантите (**People Task**), ролеви множества (**Role Sets**) и компютърно зададени изисквания (**Computer Task Requirements**). Тази част от методологията включва **силен социотехнически подход**. **Социотехническите опции** заедно със същинския **модел на информационната структура** се използват за довършване на проектирането на ЧМИ. Само когато бъде дефинирана човешката система е възможно завършване на техническите аспекти на компютърната система.

КООПЕРАТИВЕН ДИЗАЙН

Има няколко различни вида. Те се различават обикновено във философията и практиката, която всяка една от тях възприема. Те наблягат на основните **различия при проектирането** на човеко-машинни системи; различните модели на работа, области и системи, които са употребявани от потребителите и дизайнерите.

- **Участническо проектиране** - включването на потребителите в процеса на проектиране е от голямо значение. Потребителите участват анализирайки организационните изисквания и планирайки

подходящи социални и технически структури поддържащи индивидуалните и организационните нужди.

- **Социотехническо проектиране** - форма на кооперативно проектиране, която е съсредоточена върху разработването на пълни и свързани човеко-машинни системи. В подхода е наблегнато на разглеждането на социалните и технически алтернативи на проблемите.

АНАЛИЗ НА ЗАДАЧИТЕ

Задачата е това, което някой трябва да направи, за да постигне цел. **Действие** е работата, която не включва разрешаване на проблем или контрол. **Метод(план)** е последователност от работа или действия. **Обекти** са центъра на действието.

МЕТОДИ ЗА АНАЛИЗ НА ЗАДАЧИ

ЙЕРАРХИЧЕН АНАЛИЗ НА ЗАДАЧИ (ЙАЗ)

Йерархичният анализ на задачи (ЙАЗ) използва като основна конструкция **графично представяне на разлагането на задача от високо ниво** на нейните подзадачи или действия. Това е базирано на означаването със **структурна диаграма**. ЙАЗ се занимава с **логиката подредба на задачата**. Той включва **последователен процес** на идентифициране на задачите, категоризирането им, разбиването им на подзадачи и проверка на точността на разбиването. **Целта** на ЙАЗ е да **представи задачата в йерархия от действия и планове**. Целта е исканото състояние на системата, задачите, да описват начина, по който цел може да бъде достигната. Планове също се включват в представянето. Те определят точно условията, при които всяка подзадача трябва да бъде изпълнена. **Някои от подзадачите в диаграмата се разпадат на други подзадачи и са номерирани по реда, по който се извършват**. Тези номера съставят план за извършване на задачата, който **може да се представи отделно**. ЙАЗ може да бъде описан в 3 етапа:

1. Стартиране на анализа:
 - a. първо, областта на работа или **главната дейност** трябва да бъде указана
 - b. **главната задача** трябва да бъде **разбита** на между 4 и 8 подзадачи. Тези подзадачи трябва да бъдат подробно и конкретно описани
 - c. **подзадачите** трябва да бъдат **описани** като **многослойни планове**, като се осигури тяхната логическа и техническа коректност, както че и всички са налице.
2. Развиване на анализа:
 - a. следващия стадий включва **решаване за нивото на детайлност**, което се изисква и на кое ниво да се спре разлагането.
 - b. придържайки се към етапа на разлагане трябва **да се реши** дали да се **продължи с анализиране на всяка подзадача на съответното ниво** или да се **слезе на следващото ниво** и да се анализират всички подзадачи в него.
 - c. **анализът продължава**
3. Завършване на анализа
 - a. анализа трябва да бъде **проверен**, за да се уверим, че разлаганията на задачите и номерацията са последователни.
 - b. полезна практика е да се представи анализа на някого, който не е бил включен в подробния анализ, но който знае задачите, за да провери последователността

ПОЗНАВАТЕЛЕН АНАЛИЗ НА ЗАДАЧИ (ПАЗ)

ПАЗ набляга на **техниките за представяне на знанията, които хората имат или са им необходими за изпълнението на задача**. Познавателният анализ на задачи се занимава с информиране на дизайнерския процес чрез прилагането на **познавателни теории**. Дефинирахме **задача** като това, което човек или друг агент трябва да извърши, за да постигне някаква цел, използвайки някакво средство. Задачата се довежда докрай чрез извършването на действия. **Познавателния анализ оценява, че някои от тези действия са физически (натискане на бутони, говор), но някои са умствени или познавателни операции**. Задачи, като например кой бутон да бъде натиснат припомняйки предишни знания или сравняващи два обекта, са по-скоро познавателни отколкото физически операции.

ПРЕГЛЕД НА НЯКОИ ТЕХНИКИ ЗА ПАЗ:

От различните познавателни модели исторически най-важен е **модела на човешкия "процесор"**, който описва психологическия модел на хората като състоящ се от 3 взаимодействащи системи: вечната, моторната и познавателната системи, всяка от които има собствена памет, знание и процесор. Този модел доведе до методите **ЦДМИ (правила за Цели, Действия, Методи и Избор)** и **ТПС (Теория на Познавателната Сложност)** в Анализа на задачи. По-естествен метод за представяне на ЦДМИ е НЦДМИ. Теорията на Джонсън за **Структури за Познаване на Задачите (СПЗ)** допуска, че хората, изучавайки и изпълнявайки задачи, развиват структури на познанието. Метод, известен като **Научен Анализ на Задачите (НАЗ)** е използван "да идентифицира елементите на познанието, представени чрез САЗ". Можем да видим дали различните техники са приложими чрез отнасянето им към **общия модел на Анализ на Задачата**, който се състои от три нива на описание: **Цели, Задачи и Действия**. Тук се спираме на 2 основни нива на познавателна дейност, с които трябва да се занимаем: **представянето и проектирането на връзката "задача - действие"** и **представянето и проектирането на връзката "цел - задача"**.

СТИЛОВЕ НА ВЗАИМОДЕЙСТВИЕ

ВЪВЕЖДАНЕ НА КОМАНДИ

Практически, ако човек не знае протокола, който трябва да се следва, никакъв диалог не би бил възможен. Например символът ">" показва, че системата е в очакване на команда, която да й каже какво да прави по нататък, например "DIR" - да покаже съержанието на директорията.

МЕНЮТА И НАПРАВЛЕНИЕ

Менюто е набор от възможности, показани върху екрана, при който изборът и изпълнението на една (или повече) от тях води до промяна в състоянието на интерфейса. За да бъдат менютата ефективни, избраните **имена** или **иконки** трябва да показват собственото си предназначение.

В миналото един от проблемите при използването на менюта беше, че те заемат много място на екрана. Две решения на този проблем са pull-down (спускащото се надолу) и pop-up (изскачащото) меню. **pull-down менюто** се изтегля надолу от една единствена заглавна част в най-горната част на екрана, избира се елемент от него и менюто автоматично се прибира обратно към първоначалната си заглавна част. **Pop-up менютата** се появяват, когато се кликне с мишката върху определена част от екрана, обикновено обозначена чрез иконка, и менюто стои, докато потребителят му укаже да се прибере отново, обикновено като кликне върху каре за затваряне ("close box") по границата на прозореца на менюто.

ПОПЪЛВАНЕ НА ФОРМУЛЯРИ И ЕЛЕКТРОННИ ТАБЛИЦИ

Когато в системата чрез клавиатура се въвеждат няколко различни категории данни, често е полезно екранът да се оформи във вид на формуляр (**form-fill**), особено ако трябва да се въвежда многократно един и същи тип информация, както се случва при продажбите на дребно (тип, номер, цена, наличност, доставка) и досиетата на хора. Предимството на формулярите е в това, че те **помагат на потребителите да въведат данните на правилното място**, като по този начин намаляват необходимостта да се

наблюдава екрана прекалено внимателно. Обикновено, когато се попълни една част от формуляра, **курсорът сам се придвижва** директно до мястото, където трябва да се въведе следващото звено от данни. Формулярите трябва да бъдат оформени така, че потребителят да знае **какъв тип данни са допустими във всяко отделно поле**. Също така, за потребителя трябва да е очевидно как може да направи **корекция**. Един от начините да се направят формулярите лесни за използване е да се оформят така, че да приличат на добре оформените печатни формуляри, в начина по който изглеждат и се попълват.

ДИАЛОГ ВЪПРОС - ОТГОВОР

Терминът "диалог въпрос-отговор" може да се отнесе към елементарен вид меню система, при която всяко взаимодействие състоящо се от въпрос, набор от възможности и отговор от потребителя обикновено **заема един или два реда от екрана**. **Въпросите се задават един по един и следващият въпрос може да зависи от получения преди това отговор**. Този тип комуникация може **да се използва, когато очакваният вход е ограничен**. Във всяка своя форма, диалозите "въпрос-отговор" са системно - зависими диалози, които предпазват потребителя от всякаво колебание относно насоката на действие. Поради тази причина те са подходящи за **начинаещите потребители**, но могат да бъдат много изнервящи за опитните потребители, които знаят какво искат да направят.

ДИАЛОГ НА ЕСТЕСТВЕН ЕЗИК

Системата трябва да може да се справя със същата неяснота, двусмислие и неправилни граматични структури, които обикновено се свързват с разпознаването на реч. Въвеждането чрез клавиатура премахва проблема с акцента и интонацията, но поставя проблема за различията в правописа, правописните грешки и **грешките при въвеждане**. Въпреки че засега не е възможно да се разработи система, разпознаваща естествен език, въвеждан от клавиатурата, бяха създадени няколко вида експертни системи и интелигентни обучаващи системи, използващи **структурирано подмножество на естествения език**. Потребителят трябва да се научи как да използва такъв ограничен език недвусмислено и да съставя изреченията по начин, разбираем за съветната система. Друг проблем на взаимодействието чрез естествен език е многото писане.

ДИРЕКТНО МАНИПУЛИРАНЕ

Терминът директно манипулиране определя системи, притежаващи следните характеристики:

- видимост на обектите, които ни интересуват
- бързи, обратими, инкрементални действия
- замяна на сложния синтаксис на командния език с директно манипулиране на интересувания ни обект

Най-често системите с директно манипулиране използват **иконки**, **представящи обекти, които могат да бъдат движени върху екрана** и да бъдат управлявани чрез **движенето на курсор с мишката**. *Apple Macintosh* беше първата като цяло успешно наложила се търговска система с общо предназначение, която притежаваше черти от директното манипулиране. Тя се базира на аналогия с повърхността на бюро, в която се използват иконки, представящи обекти, обикновено също свързани с такива 'бюра'. Например файловете се представят чрез малки иконки.

2.ПРОЕКТИРАНЕ НА ГРАФИЧЕН ИНТЕРФЕЙС: ИНТЕРАКТИВНИ СТИЛОВЕ И ТЕХНИКИ, ОТЧИТАНЕ НА ПСИХОЛОГИЧНИТЕ ОСОБЕНОСТИ НА ПОТРЕБИТЕЛИТЕ, КОНЦЕПТУАЛНИ МОДЕЛИ И МЕТАФОРИ, МЕТОДИ И СРЕДСТВА ЗА РЕАЛИЗАЦИЯ.

ИНТЕРАКТИВНО ПРОЕКТИРАНЕ НА ИНТЕРФЕЙСИ С АКТИВНО УЧАСТИЕ НА ПОТРЕБИТЕЛИТЕ

Основен подход при проектирането на интерфейс е т.нар. интерактивно проектиране. При този вид проектиране на интерфейси в центъра на процеса се поставя **потребителя**. Като се отчита факта, че ролята на интерфейса е да помага на хората в ежедневието и работата им, се формулират и основните **цели** на този подход: създаване на лесни за използване, ефективни и приятни за човека интерфейси. За да се постигнат тези цели, се привличат **потребителите в процеса на проектиране** на интерфейса. Това става основно чрез разбиране и отчитане на това **кой са потребителите, какви са основните им дейности, и в какъв точно контекст се налага те да използват планираните интерфейси**. За да можем да оптимизираме начина, по който потребителите ще използват интерфейса, трябва да отчитаме в кои дейности те са добри и в кои не са, какво може да им помогне да подобрят работата си, да знаем какво искат и да ги накараме да се чувстват въввлечени в процеса на проектиране на интерфейса, като използваме добре изучени и проверени в практиката методи за тази цел.

В процеса на отчитане и използване на потребителите при проектирането на интерфейса ние преследваме две основни групи цели: **използваемост** на интерфейса и неговото **възприятие** от потребителя.

Използваемостта на интерфейса включва **резултатност** (доколко интерфейса помага на потребителя да извърши необходимите дейности), **експедитивност** (извършват се с **минимални усилия и лесно**), **безопасност** (**предпазва потребителя от грешки** и му позволява лесно да се поправи), **полезност** (каква част от дейностите се поддържат от интерфейса), **леснота** (лесен за изучаване и използване), **възприемчивост** (**лесен за запомняне** и повтаряне).

ФОКУСИРАНЕ НА ВНИМАНИЕТО

СТРУКТУРИРАНЕ НА ИНФОРМАЦИЯТА

Един начин, по който интерфейсите могат да бъдат замислени да помагат на потребителите да откриват информацията, от която се нуждаят е да структурират интерфейса така, че лесно да се оправят. Първо, тези заявки представяйки не толкова много **информация** и не толкова малко на **екрана**, като и в двата случая потребителя ще прекара значително време разглеждайки изцяло или безреден екран, или многобройни екрани от информация. Второ, вместо произволно представяне на данни на екрана това ще бъде **группирано** и разположено в многозначни части. От улавяването на възприетите закони за групиране на информацията може да бъде пълнозначно структурирана така, че да е по-лесно да се възприемат и да може да насочва вниманието лесно да определя информацията.

ДРУГИ ТЕХНОЛОГИИ ЗА НАСОЧВАНЕ НА ВНИМАНИЕТО

Други технологии за представяне на информацията на интерфейси за водещо внимание включва използването на:

- пространствени и временни знаци
- цвят
- сигнални техники като проблясване на екрана и обратни видео и слухови предупреждения

Прозорците са друг използван начин за разделяне на компютърния екран в предпазливи или застъпващи се участъци, даващи възможност различните видове информация да бъде разделена. Използвайки различните методи ще отбележим, че:

- Важна информация, която се нуждае от незабавно представяне, винаги ще бъде показана на забележимо място
- По-малко неотложната информация ще бъде разположена на по-малко забележимо място, но на специфичните места от екрана, такива че потребителя ще знае къде да гледа

- Информация, която не е нужна много често, няма да бъде показвана, но ще бъде на разположение при заявка.

ИНТЕРФЕЙСНИ МЕТАФОРИ И КОНЦЕПТУАЛНИ МОДЕЛИ

Опитвайки се да отгатнем какви операции предлага дадената система, ние ще се обединим не само в необходимостта, но и в нуждата от употребата на метафори.

ВИДОВЕ МЕТАФОРИ

ЕЗИКОВИ МЕТАФОРИ

Когато се сблъскаме с проблема да си служим с някое новосъздадено откритие, често сме изправени пред задачата да "разгадаем" начина за използването му. Много важно е потребителите да могат да направят **сравнение между новата технология и тази, която вече са използвали**.

ВИРТУАЛНИ ИНТЕРФЕЙСНИ МЕТАФОРИ

Xerox е една от първите компютърни компании, осъзнала огромния потенциал от проектирането на **интерфейс, наподобяващ реалния свят**, с който хората са свикнали. Вместо да създават езикови метафори, проектантите направили една крачка по-напред, като създали "interface metaphor", който се базира на действителния офис. Същността на "interface metaphor" била да се създадат електронни съответствия на реални обекти в офиса. Това включвало представянето на управляващи обекти, като **иконите** на екрана, както и листи, папки, картотеки, входни и изходни кутии. Всичко отразено на екрана представлявало desktop, наподобяващо бюро за работа.

СЪСТАВНИ МЕТАФОРИ И МНОЖЕСТВЕНИ МИСЛОВНИ МОДЕЛИ

Това са допълнителни функционалности, които не са част от интерфейсната метафора, но позволяват на компютърното приложение да бъде по – мощно. Така се стига до идеята за съставни метафори и множествени мисловни модели.

Пример за такава метафора е **scrollbar-a** - такъв обект не съществува в реалния живот, но той е измислен, за да наблегне на основната идея за развиването, а именно документи, които са навити, а трябва да бъдат развити, за да могат да се прочетат.

Други примери за интерфейсни метафори, базиращи се на обекти, комбинирани с desktop метафори, са **менютата и прозорците**. Те имат своя метафорична основа, която се отличава от идеята за офис или бюро. Въпреки това, направени достатъчно ясно така, че човек със самото заставане пред екрана вижда пред себе си всичко, което има системата така, както ако влиза в офиса си. Вторият вид използвани метафори в менютата са основани на **прототипни действия**, свързани с професионалната практика. Като пример **"cut" и "paste"** са действия, използвани за преместване и копиране на текст и заместващи идеи от оформянето и печатането на страници в реалния живот.

РАЗРАБОТКА НА ИЗПОЛЗВАЕМ ГРАФИЧЕН ИНТЕРФЕЙС: ТЕХНИКИ БАЗИРАНИ НА ЕКСПЕРИМЕНТИ.

Експериментите използвани в ПЧМИ имат тясна сфера на действие и обикновено са адресирани към специфични аспекти на ПЧМИ дизайна. Най-важните особености на **експерименталните студии** са, че изследвателя може да манипулира множеството от фактори асоциирани с дизайна, както и да изучава ефектът на различни аспекти от потребителското изпълнение. **Преди тестване е нужно да се определи изискваното ниво на потребителския опит и конструкцията на експеримента**. Обикновено само **един или ограничен брой фактори са манипулирани**, така че случайната връзка между манипулациите и потребителското изпълнение може да бъде установена. Експерименталните студии с добра конструкция

обикновено имат ясен **хипотезис**, който **предсказва очакваните ефекти** от изпълнението, асоциирани с експериментите и заключаващи със статистическият анализ на събраните данни.

Такъв вид експерименти **дават информации, но като цяло не са обособени да се правят като част от дизайнерския процес**, защото количеството ресурси е недостатъчно. При все това, когато много специфична информация се нуждае от преценка експериментите могат да бъдат потребни.

ТРАДИЦИОННИ ЕКСПЕРИМЕНТИ

Принципи на експериментирането

Нужно е когато планирате един експеримент да помислите за три негови **аспекта**:

1. **Цел на експеримента** - кое да бъде променяно, кое да бъде константа и кое да бъде измерено.
2. **Хипотезис на експеримента** - необходимо е да бъде поставен по такъв начин, че да може да се тества.
3. Какви **статистически тестове** трябва да приложите към данните, които сте събрали и защо.

Променливи - **Независимите променливи** са тези, с които експериментатора манипулира. Променливите, които са зависими от тях са **зависимите променливи**. В психологичните експерименти независимите променливи почти никога не са засегнати от участието на хората в експеримента. Обратното, зависимите променливи почти винаги са "отговорни променливи" за участието на хората в експеримента (заетото време за отговор, дължината на отговора, броя на отговорите и т.н.); така че, като цяло те формират "**външността**" на експеримента. "

Селекция на субекти - Необходимо е да селектирате Вашите субекти внимателно, така че да избегнете всякакви предубеждения.

Експериментален замисъл - Начинът по който е конструиран експеримента е също така важен. Има три добре познати експериментални **конструкции**: с **независими субекти**, с **подходящи субекти** и с **повтарящи се оценки**.

Конструкция с независими субекти - В това положение групата от субекти е получена за експеримента като цяло, след което субектите се алокират **произволно** в едно от **двете експериментални условия**.

Конструкция с подходящи субекти - В това положение субектите са **съчетани по двойки** (често това са мъж и жена, така че всяко предубеждение в резултат на пола да бъде елиминирано), след което двойките се алокират произволно в **двете условия**.

Конструкция с повтарящи се оценки - В тази конструкция **всички** субекти се поставят в **двете експериментални условия**, така че разделяйки ги на две, половината от субектите се нуждаят от сравнение с останалите две конструкции. Въпреки че, в тази конструкция няма проблеми произтичащи от алокацията на субектите, има проблеми за разрешаване свързани с реда, в който субектите работят върху задачите.

Конструкция на простите субекти - Четвъртият тип конструкция е конструкция на простите субекти, в която по-задълбочените експерименти са представени **само с един субект**. Това може би е най-важният фактор за образуване на скала, но е възможно да възникнат и проблеми. Вероятно е да не успеете да валидирате резултатите статистически, както и всяко срещащо се обучение може да направи резултатите пристрастни.

Всяка **промяна в зависимите променливи**, причинена от промяна в **независимите променливи**, наричаме **експериментален ефект**. Възможно е да има **промени в зависимите променливи, които са предизвикани от променливи, различни от независимите променливи**. Както вече споменахме в конструкцията с повтарящи се оценки, възможно е редът да има ефект върху работата на субектите. Субектите може да

са нервни, уморени или отегчени от задачите. Съществуват и много други фактори, които също влияят върху представянето на субектите.

КРИТИЧНА РЕЦЕНЗИЯ НА ЕКСПЕРИМЕНТАЛНАТА ПРОЦЕДУРА

В критичната рецензия на един експеримент е важно да разгледаме дали той прави точно това, което би трябвало всъщност да прави и приложимостта му към дизайна. Трябва да обмислим четири ключови извода:

1. **Потребителска подготовка** - дали инструкциите, дадени на потребителите са адекватни и дали количеството практика допуснато преди стартирането на експеримента е било достатъчно.
2. **Влияние на променливите** - какви промени в независимите променливи са от значение за потребителите, когато те се заемат с експерименталните задачи.
3. **Структура на задачите** - дали задачите са комплекс, достатъчен да позволи пълна употреба на удобствата на интерфейса и дали потребителите разбират целите на задачите.
4. **Заето време** - дали дължината на продължителността от задачи произвежда в потребителя някаква умора или отегчение.

КРИТИЧНА РЕЦЕНЗИЯ НА ЕКСПЕРИМЕНТАЛНИТЕ РЕЗУЛТАТИ

Експерименталните резултати се нуждаят от критична рецензия, за да се установи точно какво трябва да бъде "улавяно в грешка", доколко приложимо е това и дали практическата значимост е такава, каквато и теоритичната. Трябва да обмислим четири главни пункта:

1. **Големина на ефекта** - абсолютния размер на разликите намерени в зависимите променливи. Той е важен при оценяването на резултата.
2. **Алтернативни интерпретации** - Полезно е да се обмисли дали има някакви алтернативни интерпретации на резултатите, *вероятно базирани на други променливи, които не могат да бъдат контролирани в експеримента.*
3. **Постоянство между зависимите променливи** - когато няколко зависими променливи се използват в един експеримент, връзката между тях трябва да бъде изучена. В някои случаи може да има непостоянство сред променливите.
4. **Генерализация на резултатите** - зависи от природата на експеримента. Неговите резултати могат да не бъдат генерализирани за различни задачи, потребители или работни среди.

4. РАЗРАБОТКА НА МУЛТИМЕДИЕН ГРАФИЧЕН ИНТЕРФЕЙС: ПРОЕКТИРАНЕ НА ЦВЕТОВЕ, ЗВУЦИ, ТЕКСТ, ГРАФИКА, АНИМАЦИЯ.

Златни правила на проектирането на потребителския интерфейс:

- Да се преследва **съвместимост** и еднообразие
- Да се предвиждат кратки пътища за чести посетители
- Да се дава **ясна** и информативна обратна информация
- Така да се проектира диалога, че да води до успешен край
- Да се използва прост механизъм за обработка на грешките
- Да се дава възможност за лесно повтаряне на действия
- Да се поддържа вътрешно място за управление
- Да се ограничи натоварването на кратковременната памет на потребителя

Правила за изобразяване на данни

- **Съвместимост** и еднообразие
- **Ефективно усвояване на информацията** от потребителя

- Минимално натоварване на паметта на потребителя
- Съвместимост между начините за визуализиране и въвеждане на данни
- Гъвкаво управление за потребителя относно начините за показване на данни

Правила за привличане на вниманието:

- Интензивност (до две нива)
- Отбелязване (например подчертаване)
- Чрез различни шрифтове (до 3)
- Инвертиран образ
- Мигане (2 до 4 херца)
- Цвят (до 4 различни стандартни цвята)
- Мигане на цветовете
- Звук

Правила за въвеждане на данни

- Съвместимост и еднообразие
- Минимален брой действия от потребителя
- Минимално натоварване на паметта на потребителя
- Съвместимост между начините за визуализиране и въвеждане на данни
- Гъвкаво управление за потребителя относно начините за въвеждане на данни

5. ОСОБЕНОСТИ ПРИ СЪЗДАВАНЕ НА ИНТЕГРИРАН ИНТЕРФЕЙС: МЕТОДИ ЗА МОДЕЛИРАНЕ, НАСОЧЕНИ КЪМ КРАЙНИЯ ПОТРЕБИТЕЛ, ЕКРАНЕН ДИЗАЙН, ОБРАБОТКА НА ВЗАИМОДЕЙСТВИЯТА, ИНТЕРАКТИВНИ МЕТОДИ ЗА ПРОЕКТИРАНЕ, ПРОТОТИПИРАНЕ.

Софтуерното прототипиране е динамично симулиране. Потребителите могат да изпробват операции чрез взаимодействие със системата в "реално време", но обикновено функционалността е само частична или симулативна.

Софтуерния прототип е система, която:

- **всъщност работи**, това не е просто идея или скица
- няма обобщен "живот": от една страна може да бъде изхвърлен веднага след използването му, от друга страна може да бъде развит до крайна система
- може да обслужва много различни предназначения
- трябва да е създаден **бързо и евтино**
- е съществена част от *итеративния потребителски-централизиран проект* във всяко развитие (виж модул 4) и следващ изменението на проекта

Пълното прототипиране - както подсказва името му, има функционалността на завършения продукт, но с по-ниски показатели.

Хоризонталното прототипиране представлява потребителския интерфейс, но без функционалност.

Вертикално прототипиране притежава цялата функционалност от ниско и високо ниво що се отнася до една ограничена част от системата.

Прототипиране с висока точност се отнася до прототипирането чрез опосредстване, като например чрез **видео**, което прилича максимално на окончателния интерфейс. Прототипи с висока точност чрез видео са доста популярни в търговските организации, защото те представят продукта по-лъскав и естетически приятен.

Прототипиране с ниска точност - използва материали, които са доста далеч от крайната версия, но са доста по-евтини и позволяват бързо разработване. Например, софтуерна версия на интерфейса с орязана функционалност ще има по-голяма точност от даден сториборд.

Шофираното прототипиране включва потребителя, който наблюдава друг човек, обикновено **член на разработващия колектив как управлява системата**. Това е начин да се тества дали интерфейса отговаря на нуждите на потребителя, без това да изисква потребителя да извършва действия със системата на ниско ниво.

Прототипирането "Магьосникът от Оз" също включва трета страна, но в този случай потребителя не знае за това. Потребителят комуникира с екрана, но вместо софтуер му **отговаря човек от разработващия екип**, който стои зад друг екран, отговаряйки на въпросите на истинския потребител. Този тип прототипиране обикновено се използва в началните етапи на прототипирането, за да се постигнат очакванията на потребителя. Съществува и допълнително предимство за разработващия екип от използването на шофираното прототипиране и прототипирането "Магьосникът то Оз". То се състои в подробното разбиране на потребителя при близката работа с него.

Ползата от различни типове прототипиране в различни етапи на проекта показват две фази на итеративния дизайн.

1. **В първата фаза** прототипите се разработват да доставят **различна информация** и **силно различаващи се възможности** могат да бъдат проверени едновременно. Тази фаза за прототипиране е разклоняваща се и предлага **алтернативни дизайни**.
2. В един момент тази фаза на прототипиране приключва с предложение за единствен пълен първоначален дизайн. След това се взема едно решение чрез проектиране, кодиране и тестови цикли. Всякакви големи промени са необичайни, тъй като са много скъпи. Тази фаза може да се разглежда като **сходящи, дълговременни цикли за фина настройка**.