

Interakcija čovek-računar ispitna pitanja 2017

1. Osnovna polazna očekivanja od interfejsa i merila upotrebljivosti interfejsa (12 - 17)

- Vrednost nekog interfejsa se može odrediti analizom k-ka nalik upotrebljivosti, univerzalnosti i korisnosti.
- Efikasni interfejsi među korisnicima stvaraju osećaj uspešnosti i jasnoće. Oni ne opterećuju korisnike jer oni tačno znaju kakav će biti odziv na svaku njihovu akciju.
- Za kriterijume dizajna ljudskog inženjeringa navedeni su sledeći ciljevi :
 1. obezbeđivanje performansi potrebnih operaterima;
 2. minimalizovanje neophodnog znanja i zahtevnosti;
 3. postizanje pouzdanosti na relaciji operator-softver;
 4. standardizacija dizajna u okviru jednog sistema i između više različitih sistema
- Prvi cilj u analizi zahteva je **procena potreba korisnika** - tj. zadataka koje će interfejs morati da izvrši.
- * Ukoliko funkcio. interfejsa nije zadovoljavajuća, onda je potpuno nebitno da li interfejs lepo izgleda.
- Sledeći cilj je **obezbeđivanje pouzdanosti**.
- * Akcije moraju da funk. onako kako je naznačeno, podaci prikazuju pravi sadržaj iz baze podataka...
- Treći cilj je **promosivanje odg. standardizacije, integracije, doslednosti i prenosivosti**.
- * Čak i neznatne razlike između interfejsa mogu da dovedu do neprijatnih i opasnih grešaka i duže vreme savlad.
- Četvrti cilj je **završetak projekta na vreme i u okviru budžeta**.
- Precizno određivanje ciljne grupe i plan. skupa zadataka predstavlja osnovu za **meru upotrebljivosti**. Za svakog korisnika i svaki zadatak se mogu definisati ciljevi koji služe kao smernice dizajnerima.
 1. *Trajanje obuke* - Vreme potrebno tipičnim predstavnicima grupe da nauče da koriste komande
 2. *Brzina rada* - Koliko je vremena potrebno za izvršavanje reprezentativnih zadataka?
 3. *Učestalost grešaka korisnika* - Koliko često korisnici greše prilikom rep. zad. i kakve su to greške?
 4. *Zadržavanje usvojenih znanja* - U kojoj meri zadržavaju znanje posle određenog perioda (*sat, dan*)?
 5. *Subjektivno zadovoljstvo* - Koliko se korisnicima sviđaju pojedini aspekti interfejsa?

2. Različnost motivacija za unapređivanje upotrebljivosti (17 - 24)*Odlike sistema od velike važnosti za očuvanje života*

- Strogo kontrolisan proces dizajna, intenzivna obuka, nema prostora za grešku. Nuk. elek.

Odlike integrisanih informacionih sistema

- Fokus na brzom radu preko prečica, manji fokus na zadovoljstvo radnika. Bank. sistem.

Odlike socio-tehničkih sistema

- Fokus na zadovoljstvo korisnika. Primer sajt javne uprave.

Odlike personalnih aplikacija za kancelariju i kuću

- Manje bitni stabilnost i brzina rada, često postoje osnovne i napredne funkcije. Video igre, veb pretraživač.

Odlike profesionalnih alata

- Brzina učenja nebitna, bitan obuhvat funkcionalnosti. Alat za obradu fotografija, modelovanje...

3. Univerzalna upotrebljivost (24 - 39)

- Razumevanje fizičkih, intelektualnih i personalnih razlika potencijalnih korisnika je od vitalnog značaja za proširivanje tržišta. Naš uspeh se meri time koliko dobro razumemo i ispunjavamo potrebe naših kor. Dobro dizajniran radni prostor povećava zadovoljstvo i brzinu u radu, a smanjuje broj grešaka.

- Koncipiranjem interfejsa za razne situacije najčešće dobija proizvod koji je dobar za sve. Posebne potrebe jedne grupe kao što je, na primer, obaranje ivičnjaka neophodno za hendikepirane ljude koji se kreću u kolicima, mogu da budu od koristi i za neku drugu grupu - roditelje sa malom decom u kolicima, bicikliste, putnike sa koferima...

- *Razlike u fizičkim sposobnostima i radnom prostoru* - veličina šake npr. je jedan od parametra koji se gledao pri dizajniranju tastature i miša, težilo se ka uvažavanju najvećeg mogućeg broja razlika u fizičkim sposobnostima ljudi.

- *Različitost saznavnih i opažajnih mogućnosti* - kratkoročna i dugo. memorija, razmišljanje, procena rizika, rešavanje zadatka, sticanje znanja...

- *Individualne razlike* - Neki ljudi vole računare, neki ih ne vole, neki ljudi vole nasilne igrice, neki ne - *Jung*.

- *Kulturna i etnička raznovrsnost* - bojom žalosti u zapadnoj civ. se smatra crna boja, dok u istočnim civ. je to bela boja.

- *Stariji korisnici* - Text-to-Speech, magnifier... Kod *dece* treba imati mogućnost povratka na pret. stanje.

- Pored prilagođavanja raznim klasama korisnika, dizajneri interfejsa treba da se prilagode i raznim hardverskim i softverskim platformama. *Da li će uređaj koji ima Win 98 i Dial up raditi isto kao Win 7 BB?*

4. Ciljevi bavljenja interakcijom čovek-računar (39 - 43)**1. Uticaj na akademske i industrijske istraživače**

- Psiholozi sve više istražuju način na koji ljudi rešavaju probleme i ispoljavaju kreativnost sa računarima kako bi u potpunosti razumeli saznejne procese. Istraživači iz oblasti informacionih nauka, poslovanja, obrazovanja, sociologije takođe doprinose i imaju korist u proučavanju ICR.

2. Obezbeđivanje alatki, tehnika i znanja za komercijalne razvijaoce

- Poslodavci su na upotrebljivost dugo gledali kao na drugorazrednu temu, ali danas u sve većoj meri angažuju dizajnere korisničkih interfejsa, arhitekta informacija, testere upotrebljivosti. Postaju svesni prednosti kvalitetnih interfejsa koji osim što privlače kupce i povećavaju brzinu rada njihovih radnika.

3. Podizanje svesti o računarima na viši nivo

- I dalje postoji ogroman broj korisnika kod kojih računari izazivaju nelagodnost. Plaše se da ne pogreše ili oštete opremu prilikom njenog korišćenja, osećaju se nekompetentnim i kao da je računar pametniji od njih. Jednim delom uzrok ovakvih stavova leži u lošem dizajnu sa složenim komandama, nerazumljivim porukama o greškama, neprirodnim redosledom radnji. Jedan od ciljeva ICR je podsticanje korisnika da svoje unutrašnje strahove pretvore u odlučne akcije - umesto da se osećaju krivim kada dobiju poruku tipa SYNTAX ERROR ili glupim što ne mogu da zapamte složenu seriju akcija, oni bi trebalo da se ljute na dizajnera interfejsa koji je bio toliko nerazuman i koji im nije omogućio podesniji mehanizam ili da jednostavno potraže novi proizvod.

5. Smernice za pravljenje korisničkog interfejsa (61 - 66)

Dokumenti sa smernicama razvijaju zajednički jezik i promovišu doslednost u terminologiji, izgledu i redosledu postupaka. Smernice služe kao pravila unutar neke kompanije ili tima koja treba poštovati. Smernice su takođe korisne jer promovišu diskusiju i samo njihovo poboljšanje među dizajnerima. Pr :

- **Navigacija kroz interfejs** - obzirom da navigacija mnogim korisnicima može biti teška, važno je da postoje jasna pravila, kao što su smanjivanje opterećenja korisnika, ili neprikazivanje nepotrebnih prozora i grafičkih elemenata korisniku. Primeri smernica su *koristiti jedinstvene i deskriptivne naslove, kod binarnih izbora uvesti polje za potvrdu, za pregled velikih slika preview...*
- **Organizovanje ekrana** - neki od ciljeva smernica bi mogli biti :
 1. *Doslednost prikazivanja podataka* - standardizacija terminologije, skraćenica, formata, boja...
 2. *Efik. asimilovanje informacija od str. korisnika* - format treba da bude poznat user-u (desno poravnanje celih br.).
 3. *Min. oslanjanje na pamćenje kor.* - ne očekivati da oni pamte informacije sa jednog ekrana da bi ih koristili na drugom.
 4. *Kompat. prikaza pdtk i njihovog unosa* - format u kome se inf. prikazuju treba da bude jasno povezan s formatom u kome se unose.
 5. *Flex. kor. kontrole nad prikazom pdtk* - kor. treba da mogu da izvuku inf. u najpogodnijem obl. za tren. zad.
- **Privlačenje pažnje korisnika** - s obzirom da se korisnici mogu suočiti sa velikom količinom informacija pri radu, izuzetne uslove i vremenski zavisne informacije treba predstaviti na način koji privlači pažnju korisnika. U narednim smernicama je def. nekoliko tehnika za privlačenje k. pažnje.
 1. *Intezitet* - koristiti samo 2 nivoa u gde će viši intezitet koristiti isključivo za privlačenje pažnje.
 2. *Markiranje* - podvlačenje, uokvirivanje, pokazivanje strelicom na željenu stavku i slično.
 3. *Veličina* (max. 4), *font* (3), *boje* (4), primena inverznih boja i zvukova.
 - Treba napomenuti da preteranom primenom ovih tehnika dobijamo prenatrpane ekrane.
- **Olakšavanje unosa podataka** - doslednost svih postupaka unosa podataka, minimalan rad korisnika prilikom unosa, min. oslanjanje na korisnikovo pamćenje, kompatibilnost unesenih podataka sa njihovim prikazom na ekranu, fleksibilna korisnička kontrola...

6. Principi pravljenja korisničkog interfejsa (66 - 82)

Za razliku od usko fokusiranih smernica, principi su fundamentalniji, primenjiviji i dugovečniji. Na primer, princip zadovoljavanja korisničkih osobnosti u sistemu svakom dizajneru zvuči logično, ali njegovo ispunjenje i nije tako jednostavno, u smislu zadovoljavanja različitih grupa nivou na osnovu njihovog znanja (npr. bibliotekar i predškolsko dete) i sl. Neki od fundamentalnih principa su :

- **Definisanje nivoa korisničkog znanja** - svaki dizajn počinje definisanjem ciljne grupe korisnika, uključujući pol, starost, fizičke i sazajne sposobnosti... Potrebno je obratiti pažnju i na korisnikovo poznavanje korisničkih interfejsa i domena same aplikacije, što možemo saznati odgovarajućim testovima. Proces upoznavanja korisnika nikada ne prestaje, jer se i oni konstantno menjaju, ali primera radi neka imamo opštu podelu korisnika na početnike, osrednje i profesionalce.
 - Početnicima (tipa bake i deke) treba ograničiti rečnik pojmova na jednostavne i poznate termine. Broj radnji takođe treba da bude što manji kako bi uspeali da završe svoj prvi zadatak i pobede strah, kao i da dobiju samopouzdanje i steknu pozitivan stav prema interfejsu.
 - Osrednji korisnici (tipa menadžeri koji koriste tekst editor za pravljenje šablona) već znaju o konceptima zadatka i interfejsa, ali npr. struktura menija ili pronalazak određenih f-ja im pravi problem. Dosledni redosled radnji, terminologija, smislene poruke i smernice za uobičajni način korišćenja mogu da pomognu korisnicima.
 - Eksperti su upoznati do detalja sa samim zadatkom i teže ka što bržem obavljanju posla. Oni zahtevaju brz odziv, sažete i diskretne povratne informacije i prečice kako bi se željene akcije mogle odraditi jednostavnim potezom tastature ili miša.
- Ukoliko isti sistem treba da koristi nekoliko klasa korisnika, teži se *višeslojnom* pristupu u učenju.
- **Prepoznavanje zadataka** - Nakon skiciranja profila korisnika, developeri treba da identifikuju zadatke koje je potrebno izvršiti. Složeni zadaci mogu da se razlože na više srednje teških radnji, koje dalje mogu da se razlože u atomske operacije koje korisnici mogu da izvrše jednom komandom tipa izborom iz menija, prečicama i sl.
- **Izbor stila interakcije** - Nakon analize zadataka dizajner bira jedan od sledećih stilova interakcije:
 1. **Direktno manipulisanje** - radna površina (Desktop), igrice i sl. Manipulisanje vizuelnim prezentacijama objekata i akcija omogućava korisnicima da brže završavaju poslove i prate rezultate svojih akcija (prevlačenje datoteke u korpu npr.).
 2. **Izbor iz menija** - korisnici čitaju spisak određenih stavki iz njega i biraju onu koja je najpodesnija za ono što žele da učine a zatim prate efekte svog postupka. Prednost ovog pristupa je jasna struktura procesa odlučivanja jer su svi izbori prikazani istovremeno.
 3. **Popunjavanje obrazaca** - Prilikom unosa podataka, oslanjanje samo na izbor iz menija nije efikasno i upravo tad se koristi popunjavanje obrazaca (upisivanje u prazna polja). Pojednostavljuje unos podataka i ne zahteva previše treninga. (ali treba znati opseg vred.i sl.)
 4. **Komandni jezik** - čestim korisnicima pruža jasan osećaj kontrole. Kada nauče sintaksu, oni mogu da formulišu i veoma složene zahteve, i to bez potrebe da čitaju odziv na ekranu što inače usporava rad. Ovde su greške česte, i obuka je neophodna.
 5. **Prirodni jezik** - i pored ograničenog uspeha, programeri se nadaju da bi računari jednog dana mogli da odgovaraju na proizvoljne rečenice i fraze na prirod. jezicima. *Siri*.

Moguće je i mešati više različitih stilova interakcije. Neki od 8 zlatnih pravila dizajna interfejsa su težnja doslednosti, univerzalnoj upotrebljivosti, sprečavanje grešaka, nuđenje povratnih informacija, mogućnost poništavanja akcija...

7. Korisnik-centrična naspram tehnoцентриčne organizacione paradigme (110 - 113)

- U prvim decenijama razvoja računara i softvera, programeri su pravili softvere za sebe te su kompleksni i napredni interfejsi štaviše bile i poželjni. Programeri su se prilagođavali mašinama. Današnji korisnici ne moraju biti posvećeni tehnologiji - tehnoцентриčni stil prošlosti lagano biva potiskivan željom developera za izlazak u susret korisnikovim ciljevima. Mašina se prilagođava korisnicima. Iterativna metoda dizajna koja omogućava revizije sistema na osnovu reakcija korisnika dovodi do visokokvalitetnih sistema.
- Inženjering upotrebljivosti postaje prepoznatljiva disciplina čiji je cilj testiranje i poboljšanje upotrebljivosti sistema. Kada su konkurentski proizvodi slični po funkciji, UI postaje presudan aspekt njihove prihvaćenosti. Poboljšavanjem upotrebljivosti kompanije povećavaju produktivnost i zadovoljstvo korisnika i omogućavaju sebi dugoročan profit.

8. Tri potporna stuba razvoja korisničkog interfejsa (114 - 117)

1. Dokumentovane smernice i procesi

- Jedna od najbitnijih komponenti uspeha Apple-a sa Macintosh-om bio je na vreme napisan dokument sa smernicama u kome su formulisani jasni principi kojih su se pridržavali mnogi programeri. Na taj način obezbeđen je sklad između proiz. razl. proizvođ. *pr. terminologija, skraćenice, fontovi, veličina fonta, boje, izbor iz menija, poravnanje...*
- Kada tim usvoji smernice, proces implementacije se nastavlja uz minimalne izmene.

2. Softverske alatke za korisničke interfejse

- Kupci i korisnici često ne znaju kako će sistem izgledati kad bude završen. Problem je kada se ovi sistemi jednom uvedu, njihovo suštinsko menjaje je teško, skupo i dugo traje. Ovaj problem se olakšava ako u prvim fazama projekta prikažemo korisnicima kako će dizajn finalnog proizvoda izgledati (prototip), *npr. napravimo 2 od 50 funkcija čisto da imaju ideju.*

3. Stručna mišljenja i testiranje upotrebljivosti

- Dizajnerima je jasno da pre nego što svoj dizajn ponude kupcima, najpre moraju da izvrše mnogo malih i nekoliko pilot testova komponenti. Osim raznih metoda pregleda, rade se i testovi sa izabranim korisnicima. Same procedure se bitno razlikuju i zavise od ciljeva, broja očekivanih korisnika, opasnosti koje mogu proisteći iz grešaka i budžeta projekta.

9. Metodologija logičkog korisnik-centričnog interaktivnog projektovanja (LUCID) (118 - 122)

- LUCID je opšte prihvaćena razvojna metodologija interaktivnog dizajna koja je koncentrisana na korisnika i obuhvata šest etapa :

1. **Vizija** - Usklađivanje obaveza svih učesnika, formiranje strategije i jasne i zajedničke vizije kakav proizvod treba da bude (*tj. form. (posl.) ciljeva*). U ovoj fazi razvija se koncept proizvoda.
2. **Otkrivanje** - Proučavanje korisnika u cilju definisanja njihovih potreba - kakve zadatke će izvršavati, koju terminologiju koriste, njihove sklonosti...
3. **Temelji dizajna** - Formiranje prototipa ključnog ekrana, zatim sledi testiranje upotrebljivosti dizajna, revizije i ponavljanje.
4. **Detaljni dizajn** - Realizacija dizajna visokog nivoa u zaokružene specifikacije, kompletiranje smernica za dizajn, pravljenje detaljnih skica svakog ekrana.
5. **Izrada** - Podrška proizvodnog procesa kroz kritička mišljenja i razrešavanja izmena u završnoj fazi.
6. **Objavljivanje** - Razvijanje plana implementacija proizvoda koji će podržati korisnike u prelasku na novi proizvod - lideri identifikuju eventualne prepreke za prihvatanje softvera i formulišu odgovarajuća ohrabrenja ili podsticaje; izvođenje finalnog testa upotrebljivosti, merenje zadovoljstva korisnika i dokumentovanje zaključaka.

10. Tehnika terenskog posmatranja (122 - 125)

- Početne faze većine dizajnerskih metodologija obuhvataju i posmatranje korisnika. Terensko istraživanje je inspirisano etnografskim istraživanjima. Etnografi se uključuju u radna, kućna, seoska i druga okruženja i posmatraju, osluškiju i ponekad zapitkuju. Cilj ovakvog posmatranja jeste prikupljanje podataka neophodnih za redizajn interfejsa. Posmatrač pre samog istraživanja se adekvatno pripremi (priprema eventualnih pitanja, upoznavanje sa procesima rada i istorijatom, pribavljanje dozvola...). Važno je napomenuti da posmatrač može uticati na tok izvršavanja radnji, čak i da toga ni posmatrač ni posmatrani nisu svesni (posmatrani ne žele da izgledaju loše npr.) Posmatrači trebaju da budu spemni na nepredviđene pojedinosti na koje naiđu. Nakon istraživanja vrše se analize nad podacima i pišu izveštaji (kratki sastavi iz kojih se preradom mogu izvući scenariji).

11. Participacija u dizajnu i razvoj scenarija (125 - 129)

- Prilikom terenskog posmatranja, lično prisustvo dizajnera im omogućava da razviju saradničke odnose sa krajnjim korisnicima sa kojima mogu da diskutuju o svojim idejama, a isto tako ti korisnici na ovaj način dobijaju priliku da aktivno učestvuju u procesu dizajna svojih novih interfejsa. Podstiče prihvaćenost krajnjeg rešenja i približava ga odgovarajućim potrebama. Sa druge strane, skupo je i produžava vreme implementacije, a može i stvoriti neprijatelje ili rezultirati trulim kompromisima (sa nekompetentnim učesnicima).

- U slabije definisanim projektima, mnogi dizajneri su pronašli scenarije iz svakodnevnice koji pomažu u opisu onoga što se događa kada korisnici izvršavaju tipične zadatke. Pisanje scenarija predstavlja jednostavan način za predstavljanje inovativnog sistema na samom početku, a ukoliko je moguće izvode se i u formi pozorišne predstave. Efikasnost ove metode posebno dolazi do izražaja kada više korisnika treba da ima saradničke odnose (kontrolne sobe, kabine aviona...). Svaki scenario može da se odnosi na uobičajene ili alarmantne radnje, čiji akteri mogu biti i početnici i eksperti.

12. Socijalne posledice i pravni aspekti razvoja korisničkog interfejsa (129 - 133)

- Interaktivni (vladini) sistemi često imaju priličnog uticaja na veliki broj korisnika. Da bi se smanjili rizici i motivisali produktivni predlozi u ranoj fazi razvoja u kojoj je sve još uvek lako promeniti, svim učesnicima se često upućuju smišljene izjave sa predviđenim posledicama. *Izjava o socijalnim posledicama* može da pomogne u promovisanju visokokvalitetnih sistema u aplikacijama za potrebe vlade. Pravovremena i sveobuhvatna diskusija može da otkrije probleme i omogući učesnicima da iskažu svoje stavove.

Naravno, uvek postoji opasnost da ovakve diskusije samo pojačaju strahove i prinude dizajnere na nerazumne kompromise, ali u projektima sa dobrim menadžmentom ove opasnosti su prihvatljive.

U izjavi se opisuje novi sistem i njegove prednosti i skreće se pažnja na moguće probleme i prepreke.

- Danas svaki developer treba da poznaje pravne aspekte koji bi mogli da utiču na dizajn ili implementaciju njegovog proizvoda.

* **Privatnost** je jedan od aspekata - medicinski, pravni, finansijski i drugi podaci često moraju da se zaštite kako bi se sprečio neovlašćen pristup, njihovo menjanje ili brisanje ili bilo kakva zloupotreba. Osim fizičke bezbednosti, u okviru interfejsa se mogu implementirati mehanizmi za pristup podacima putem lozinki i provere identiteta.

* Drugi aspekt se tiče **bezbednosti i pouzdanosti** - korisnički interfejsi aviona, automobila, vojnih sistema, medicinskih instrumenata nemaju prostora za konfuziju niti bilo šta što bi moglo da prouzrokuje fatalnu grešku kod korisnika.

* Treći aspekt je **zaštita autorskih prava softvera** - potpuno je normalno što developeri softvera žele da zaštite svoje softverske pakete nakon uloženog vremena i novca u njih.

* Četvrti aspekt se odnosi na **zaštitu autorskih prava onlajn informacija, slika i muzike** - da li korisnici nakon što pristupe nekom onlajn resursu imaju pravo da ga sačuvaju za svoje kasnije potrebe? Da li smeju to da šalju svojim kolegama? Da li se oni mogu smatrati vlasnicima tih informacija?

* Peti aspekt je **sloboda govora u elektronskim okruženjima** - da li korisnici imaju pravo da koriste elektronsku poštu za kontroverzne izjave? Da li su ovakve izjave zaštićene ustavima država? Da li su operateri odgovorni za eliminisanje neukusnih sala, priča i slika?

13. Evaluacija interfejsa stručnom kontrolom i revizijom (140 - 144)

- Iskusni dizajneri vremenom stiču dovoljno mudrosti da znaju da je detaljno testiranje neophodno. Ako povratne informacije predstavljaju „doručak šampiona”, onda je testiranje „večera bogova”. Neke od stavka koje trebaju da se nađu u planu evaluacije su *inovativnost projekta, broj korisnika, korišćenje interfejsa u kritičnim situacijama, cena proizvoda i sredstva namenjena za testiranje...* Do pre nekoliko godina fokus na upotrebljivost i njeno proveranje se smatralo plusom, dok je danas to praktično neophodno, ne samo zbog konkurencije već i zbog korisnika (može se završiti tužbom).

- Stručne kontrole obavljaju stručnjaci iz oblasti u kojoj će se interfejs koristiti i mogu se obaviti u početnoj ili finalnoj fazi proizvoda. Rezultati kontrole se mogu dati u obliku izveštaja u kojem su nabrojani problemi i sugestije izmena, a može se završiti i kao neformalan razgovor sa dizajnerima. Kontrola može da traje od polovine dana do jedne nedelje, a neke od metoda kontrole su :

* **Heuristička evaluacija** - kroz kritiku interfejsa ustanovljuje se u kojoj meri je on usklađen sa osnovnim dizajnerskim heuristikama (nalik 8 zlatnih pravila dizajna)

* **Pregled smernica** - provera usklađenosti interfejsa sa smernicama organizacije

* **Provera doslednosti** - provera doslednosti terminologije, fontova, šeme boje, materijala za obuku...

* **Kognitivni prolaz** - Simulira se korišćenje rešenja od strane korisnika, npr. osnovni unos podataka ili oporavak od greške.

- Stručnjaci se stavljaju u svakodnevne situa. pri radu sa interfejsom. Met. ptič. persp. - proučavanje svih odštam. ekrana na zidu.

14. Testiranje upotrebljivosti i laboratorije (144 - 150)

- Menadžeri su shvatili da postojanje testa upotrebljivosti u planu projekta predstavlja jak podsticaj za dovršavanje faze dizajniranja. Izveštaji sa testova su potvrda brzine kojom napreduje projekat, a njihov rezultat su konkretne preporuke za eventualne promene. Dizajneri su na ovaj način dobili pomoć u vidu povratnih informacija o vrednosti dizajna, dok menadžeri više ne doživljavaju velike drame u završnim fazama projekata. Pored ubrzanja realizacije projekata, test. upotrebljivosti je donelo i značajne uštede.

- Sve veća prihvaćenost TU dovela je do nastanka laboratorija za ispitivanje upotrebljivosti. Tipična laboratorija ima dve prostorije odvojene lažnim ogledalom: u jednoj prostoriji učesnici rade ono što je predviđeno da rade, dok se u drugoj nalaze testeri i posmatrači (dizajneri, menadžeri i mušterije). Vreme pripreme laboratorijskog proveravanja upotrebljivosti traje nedeljama i košta, a korisnik dok radi je često sniman kako bi se zabeležili delovi kada je razmišljao naglas.

- Veoma retko se proizvodi tačno naprave iz prvog pokušaja. Testiranje upotrebljivosti predstavlja inženjersko-psihološku metodu za pronalaženje mogućnosti poboljšanja upotrebljivosti proizvoda.

Nabroj par modeliteta testiranja upotrebljivosti? - Kompetetivno, pripremno, pilot, univerzalno testiranje, testiranje na daljinu...

- **Kompetetivno testiranje?** - Poređenje sa prethodnim verzijama ili sličnim proizvodima. Važan je redosled izlaganja.
- **Pripremno testiranje?** - Ima samo par učesnika, najbolje je za evaluaciju testa upotrebljivosti koji je u fazi pripreme.
- **Pilot testiranje?** - Daje okvirne vrednosti.
- **Univerzalno testiranje?** - Mnogo učesnika, vrši se u raznim hard./soft./mrežnim okruž., npr. na raznim OS-ovima i pretraživačima.
- **Testiranje na daljinu?** - Veb i mobilne aplikacije mogu testirati učesnici širom sveta.

15. Instrumenti za anketiranje i upitnik o satisfakciji korisnika interakcijom (QUIS) (150 - 162)

- Ankete predstavljaju jeftinu i efikasnu dopunu testova upotrebljivosti. Jednostavno se savladavaju, a veliki broj korisnika ove metode joj daje i kredibilitet. Anketni listići se prvo pripremaju i testiraju na malom uzorku korisnika pre nego što se pristupi pravom ispitivanju. U anketama se od korisnika može tražiti da iznesu svoje utiske o konkretnim aspektima interfejsa kao što su *radnje za obavljanje konkretnih poslova, akcije u okviru interfejsa, sintaksa ulaza i dizajn ekrana*, ali isto tako među ostalim ciljevima je procena nekih korisničkih aspekata kao što su *profil (starost, pol, obrazovanje...), iskustvo sa računarima, karakteristika ličnosti...*

- U anketama se često koriste skupovi bipolarnih semantički povezanih parova uz pomoću kojih korisnici opisuju svoje reakcije. Imamo skalu na kojoj dva kraja izražavaju potpuno suprotne stavove (npr. neprijateljske/prijateljske, nejasne/konkretno...), a korisnici rangiraju poruke koristeći ocene sa skala.

- *Upitnik o satisfakciji korisnika tokom interakcije (QUIS)* se fokusira na detalje interfejsa kao što su čitkost znakova i koncepcija ekrana, interfejsni objekti (npr. značenje ikona) i akcije (prečice), poslovni aspekti (npr. odgovarajuća terminologija) i redosled ekrana.

1. Sveukupna reakcija na rešenje
 - Užasno / Divno
 - Teško / Lako
 - Frustrirajuće / Zadovoljavaće
 - Dosadno / Stimulišuće
 - Rigidno / Fleksibilno
2. Ekran
 - Čitanje karaktera sa ekrana
 - Teško / Lako
 - Organizacija informacija
 - Zbunjujuća / Veoma uredna
3. Terminologija i sistemske informacije
 - Korišćenje termina kroz sistem
 - Nedosledno / Dosledno
 - Terminologija je intuitivna
 - Nikad / Uvek
 - Položaj poruka na ekranu
 - Nedosledan / Dosledan
 - Dijalog prozori za unos podataka
 - Zbunjujući / Razumljivi
 - Poruke greške
 - Ne pomažu / Pomažu
4. Učenje
 - Težina upravljanja sistemom
 - Teško / Lako
 - Izvršavanje zadataka je jednostavno
 - Nikad / Uvek

16. Testovi prihvatljivosti i evaluacija tokom aktivnog korišćenja (162 - 167)

- Najčešće se za softver definišu testovi da bi se saznalo vreme odziva, interval između grešaka i sl. Ukoliko finalni proizvod ne ispuni ove kriterijume prihvatljivosti, sistem mora da se prerađuje sve dok se na testovima ne postignu ciljni rezultati. Testovi prihvatljivosti se takođe koriste i kod ljudskih interfejsa. Prilikom dokumentacije zahteva koje interfejs mora da zadovolji izriču se konkretni kriterijumi prihvatljivosti koji se tiču stvari tipa *vreme potrebno za savladavanje osnovnih funkcija, brzina izvršavanja zadataka, učestalost grešaka, subjektivno zadovoljstvo korisnika*. Razlika između testova prihvatljivosti i testova upotrebljivosti je ta što centralni cilj TP nije uočavanje grešaka, već provera da li su određeni zahtevi ispunjeni.

- Evaluaciju interfejsa tokom aktivnog korišćenja mogu obavljati stručnjaci zaduženi za pružanje korisničke podrške ili sami korisnici proizvoda. Ona se zasniva na kontinuiranom evidentiranju podataka o performansama korisnika, na primer:

- *Šta je pogrešno i treba da se ispravi?*
- *Šta se često unosi?*
- *Šta se od funkcionalnosti nikada ne koristi? (zbog čega?)*

Napomena : treba izbegavati prikupljanje podataka o samim korisnicima, na primer preko veb logova, ili biti u njemu transparentan, jer u suprotnom može doći do problema.

- Osim evidentiranjem poziva za podršku i analizama logova, u evaluaciji interfejsa korisni su i intervjui i diskusije u fokus grupama (vrlo korisne, ali i skupe), sanduče za onlajn predloge, kao i forumi, diskusione grupe i društvene mreže, gde nastaju zajednice čije je jezgro naš proizvod.

17. Metode za izradu specifikacije interfejsa (174 - 183)

Uspešni arhitekti korisničkih interfejsa znaju da je, pre početka konkretnih radova, dobro imati kompletan dizajn, bez obzira na to što će tokom procesa izgradnje doći do određenih izmena. Postoji nekoliko metoda za zapisivanje specifikacije, ali najpopularnije su *strukture menija* i metoda *dijagrama tranzicije*.

- **Stablo menija** ima jednostavnu strukturu, veoma je podesno kao alatka za izradu specifikacije jer pokazuje kompletnu i detaljnu sliku sistema. Na stablu se vide odnosi visokog nivoa i detalji niskog. Najvažnija činjenica je da možemo videti čitavu strukturu sistema kako bismo mogli da proverimo njegovu doslednost, kompletnost, kao i eventualno postojanje dvosmislenosti i suvišnih stavki.
- Nedostatak stabla menija je što ne pokazuje čitavu strukturu mogućih akcija kao što su npr. vraćanje u prethodni meni, prelazak u početni meni i sl. Ove akcije su namerno izostavljene jer bi njihovo stavljanje potpuno prekrilo čistu strukturu stabla. Ipak, u nekim procesima dizajna neophodno je definisati sve moguće akcije i stanja sistema. Prilikom njihovog definisanja koristi se notacija dizajna zvana **dijagram tranzicije** - sastoji se od skupa *čvorova* koji predstavljaju sistemska stanja i skupa *veza* između tih čvorova koji označavaju moguće tranzicije.

18. Alati za pravljenje modela interfejsa i alati za softverski inženjering interfejsa (183 - 192)

Dizajn na papiru je dobar za početak, ali detaljna specifikacija kompletnog korisničkog interfejsa zahteva korišćenje softverskih alatki. Ove alatke omogućavaju vizuelno uređivanje, što znači da dizajneri momentalno mogu da provere „izgled” sistema i izmene njegove boje, fontove i koncepciju. Prednost ovih alata potiče iz *nezavisnosti korisničkog interfejsa*, odnosno razdvajanja dizajna korisničkog interfejsa od složenosti programiranja.

Developeri interfejsa znaju koliko je važno u ranoj fazi dizajna imati skice koje omogućavaju istraživanje raznih alternativa, kao i komunikaciju u okviru tima a mogu i služiti za pokazivanje izgleda budućeg interfejsa klijentima. Za pravljenje modela korisničkih interfejsa dovoljni su papir i olovka, ali veštiji dizajneri svoje prototipove prave pomoću programa nalik Microsoft PowerPoint i Visio, Apple Keynote i sl. koji između ostalog kasnije mogu da se distribuiraju preko Interneta. Prototip možda neće imati punu bazu podataka, pomoć ili neke druge funkcije, ali će omogućiti kretanje pažljivo izabranom putanjom koja daje realnu predstavu o tome šta će budući interfejs sve moći.

Iako iskusni programeri nekada korisničke interfejse prave u programskim jezicima opšte namene kao C++, C# ili Java, danas je sve češći pristup gde se koriste alatke za pravljenje korisničkih interfejsa koje predstavljaju generatore interfejsa - praktično sve komponente aplikacije mogu da se naprave veoma brzo korišćenjem ovih vizuelnih alatki. *Microsoft Access, Visual Basic i Eclipse* su samo neke od mnogih. Postoji četiri softverska sloja za pravljenje korisničkih interfejsa (1. Sistem prozora 2. Komplet alatki za GUI 3. Radni okvir aplikacije 4. Aplikativni).

19. Inženjering interfejsa u slojevima prozorskih sistema i kompleta alata za GUI (192 - 198)

Neke platforme su veoma mlade, te za njih još ne postoje razvojne alatke višeg nivoa, što primorava softverske inženjere da rade na nižem nivou - ovde mogu da se menjaju neki grafički elementi nalik ikona, slika, kursora ili fontova. Svaka interaktivna aplikacija sadrži *while(true)* petlju, tzv. *glavnu petlju* koja upravlja tranzicijama stanja koje iniciriraju događaji koji stižu iz prozorskog sistema ili operativnog sistema (npr. *OnLoad, OnClick* i sl.). Programiranje na ovom nivou se radi ukoliko ne postoje alatke višeg nivoa ili kada su memorija i performanse od ključnog značaja.

U okviru većine proizvoda postoje *kompleti alata za GUI*, programske biblioteke za korisnički interfejs u koje spadaju uobičajne interfejsne komponente tipa prozora, traka za pomeranje sadržaja prozora, padajući meni, polja za unos podataka, dugmad i okviri za dijalog. Ove biblioteke omogućavaju ogromnu fleksibilnost, ali često su potrebni meseci učenja kako bi se savladale alatke koje ih koriste (*Windows Forms, Apple Macintosh Toolkit*). Njihova osnovna prednost je ta što programeri imaju apsolutnu kontrolu i veliki fleksibilnost pri pravljenju interfejsa.

20. Inženjering interfejsa u radnom okruženju aplikacije i specijalizovanom jezičkom sloju (198 - 202)

Čak i uz kompletne alati i odgovarajuće okruženje sa vizuelnim editorima, programiranje je i dalje dug i težak posao. Programiranje UI je do sada pokušano da se pojednostavi na dva načina - radnim okruženjem aplikacije i specijalizovanim jezičkim slojem.

Radno okruženje aplikacije u osnovi ima objektno orijentisano programiranje. Ideja na kojoj se zasnivaju radna okruženja je ta da svi programi sa korisničkim interfejsima imaju sličnu strukturu, koja se prevodi u klase, objekte i metode. Ovo dovodi do toga da aplikacija ne mora da se pravi od nule, već može da se upotrebi neko pripremljeno radno okruženje. Aplikacijska radna okruženja omogućavaju efikasno i brzo pravljenje naprednih UI, ali zahtevaju dugotrajnu obuku jer su veoma apstraktna i zahtevaju od programera da razume veliki broj koncepata i načine na koje se klase i metode povezuju.

Umesto proširivanja biblioteka postojećih jezika, drugi pristup koji se nameće je pisanje konkretnog programskog jezika za pravljenje korisničkih interfejsa i lakše programiranje. Tako je nastao jezik scenarija **Tcl**, kao i prateći komplet alati **Tk**. Jezik scenarija je programski jezik koji je napravljen tako da se ugrađuje u aplikacije kako bi im dodao neke nove funkcije ili povezao u jednu celinu. Ovi jezici nemaju vizuelni editor, ali sam potencijal jezika obesnažuje većinu mogućih kritika. Neki od scenarijskih jezika koji potiču od *Tcl/Tk* programskog jezika su *Visual Basic*, *Python*, *Javascript*. Ovakav inženjering se pokazao veoma pogodnim za pravljenje prototipova korisničkih interfejsa.

21. Alati za evaluaciju i kritički pregled interfejsa (203 - 205)

Prilikom dizajniranja korisničkih interfejsa, čak i jednostavne alatke za proveru tačnosti pisanja ili doslednost termina značajno pomažu dizajnerima interfejsa. Sistemi za merenja koji prijavljuju broj ekrana, interfejsnih komponenti i sl. mogu da pomognu u određivanju veličine projekta. Isto tako dizajneri mogu videti da li je stablo menija previše duboko ili sadrži suvišne elemente, da li su oznake korišćene dosledno, da li su za svu dugmad vezane odgovarajuće tranzicije.

Sa druge strane je *softver za evidentiranje u toku izvršavanja* koji beleži ponašanja korisnika - prave se jednostavni izveštaji koji se odnose na učestalost pojedinih poruka i grešci, izvore iz menija, pozivanje pomoći i sl. Istraživači takođe evidentiraju podatke o performansama koji im dalje služe kao smernice za neke odluke.

Uglavnom alati vrše analizu interfejsa - merenja nalik broja komponenti po okviru za dijalog, njihova gustina, razmera ekrana i njegovo poravnanje, lista upotrebljenih boja i stilova fontova, ali postoje i alati koji se bave proverom pravopisa i dostizanja doslednosti. Nakon analiza programi izbacuju savete koji mogu da igraju ulogu smernica za dizajnere.

22. Koncept i primeri direktne manipulacije (214 - 231)

Okosnicu interfejsa koji izazivaju zadovoljstvo i pozitivna osećanja kod korisnika čine *interfejsi sa direktnom manipulacijom* - njega čine uočljivi objekti i zastupljenost radnji koje se i traže (brze, inkrementalne i reverzibilne (undo)), a ukucavanje komande je zamenjeno pokazivanjem na objekat od interfejsa. Prevlačenje datoteke do kante predstavlja dobro poznat primer direktne manipulacije. Direktna manipulacija je pristupa od video igara pa sve do mobilnih uređaja i veb stranica.

Omiljeni primer direktne manipulacije predstavlja vožnja automobila. Prizor je direktno vidljiv kroz šoferšajbnu, a izvođenje radnji kao što su kočenje ili promena pravca postale su uobičajeno znanje u našoj kulturi. Da bi skrenuo levo, vozač jednostavno okreće volan na levu stranu. Odgovor na tu radnju je momentalan i prizor ispred njega se menja, dajući mu povratne rezultate tog skretanja.

Odlike direktne manipulacije su : kratko vreme za obuku za korišćenje, brz odziv tj. prikaz u realnom vremenu, dosta se koriste pokazivački uređaji i pokazivački tasteri na tastaturi, a manje ostali tasteri.

Primeri direktne manipulacije :

- Ikonice - slikovne reprezentacije funkcionalnosti, sličice su metafore radnji (*kanta za smeće nije bukvalno kanta za smeće, već označava radnju koja nalikuje bacanju u kantu za smeće*).
- U prošlosti korisnici prilikom uređivanja teksta su bili u stanju da vide samo jedan red. Prilikom evolucije interfejsa, korisnici su u mogućnosti da vide pun prikaz teksta i mogu da vrše promene upotrebom „backspace” karaktera ili direktnim ukucavanjem teksta.
- Osim raznih **editora**, primer direktne manipulacije su i **video igrice** gde korisnici direktno vrše interakciju sa igrom preko pokazivačkih uređaja i dobijaju istovremene rezultate.
- **Teledirigovanje** - kontrola udaljenog fizičkog procesa direktnom manipulacijom (vođenje bespilotne letelice, telemedicina...)
- **Virtuelna stvarnost** - stvaranje potpunog osećaja prisustva na željenom mestu (simulator leta, komandnog mosta...). HTC Vive i Oculus Rift.
- **Proširena stvarnost** - Google naočare.

23. Problemi kod direktne manipulacije (231 - 236)

- Prostorna ili vizuelna prezentacija nemaju nužno prednost nad tekstom, posebno za slepe i slabovide korisnike kojima je potreban specijalan softver.
- Rešenja sa direktnom manipulacijom zauzimaju vredan prostor na ekranu i stoga potiskuju vredne informacije sa njega, zahtevajući pomeranje sadržaja prozora ili skup više radnji da bi se do njih ponovo došlo.
- Korisnici moraju učiti značenje vizuelne postavke - iako dizajner možda razume ikonicu iz prve, korisniku će trebati određeno vreme da nauči njihovo značenje, što ne bi bio isti slučaj sa rečju.
- Vizuelna prezentacija može odvesti korisnika na stranputicu - korisnici mogu brzo shvatiti analognu predstavu, ali mogu izvesti pogrešne zaključke u vezi s dozvoljenim radnjama.
- Postoje ljudi kojima smeta vreme potrebno da se ruka premesti sa tastature na miš pa nazad umesto direktnog ukucavanja naredbe (*na primer, navikli su na korišćenje prečica*).

24. Vizuelno razmišljanje i ikonice (236 - 241)

Sve vizuelnija priroda računarskih interfejsa može ponekad biti izazov ili čak i pretnja racionalnim programerima kojima vlada logika i kojima je bliži linearni, na tekst orijentisani interfejs. Sa druge strane, računar pruža izuzetno vizuelno okruženje za otkrivanje strukture, prikazivanje odnosa i interaktivnost koje je privlačilo korisnike sa umetničkim sklonostima i intuitivne osobe.

Kao što postoje različiti ukusi sladoleda, tako postoji i više stilova interfejsa. Konflikt između teksta i grafike dostigao je vrhunac kada se pojavilo pitanje ikona - slikovne reprezentacije funkcionalnosti malih dimenzija, sličice su metafore radnji (*kanta za smeće nije bukvalno kanta za smeće, već označava radnju koja nalikuje bacanju u kantu za smeće*) ili predstavljaju neki objekat.

Manje ikone se često koriste radi očuvanja prostora ili integracije sa drugim objektima, pr. ikone u alatu za obradu slika predstavljaju alate ili radnje, dok su u programima za obradu teksta češći tekstualni meniji. Postoje i prilike gde su moguće kombinacije vizuelnih ikona i tekstualnih stavki - osmougaoni STOP znak. Odgovor na dilemu između teksta i ikone zavisi ne samo od korisnika, već i od konkretnog posla koji se obavlja posredstvom interfejsa.

Neke od smernica pri dizajniranju ikona su: *predstavljanje objekta ili radnji treba da bude izvedeno na poznat način, broj različitih ikona ograničiti, ikona treba da bude upadljiva, ikone treba da se razlikuju međusobno*. Ikone takođe mogu biti dinamične u odnosu na akciju koja se dešava na objekat koji predstavljaju (*kristić pri brisanju npr.*).