

Značajke, scenariji i priče

Programsko inženjerstvo

Softverski produkti

- Tri su faktora koja potiču razvoj softverskih produkata:
 - Poslovne i korisničke potrebe,
 - Nezadovoljstvo poslovnim ili korisničkim softverom koji se trenutno koristi,
 - Tehnološke promjene omogućavaju korištenje potpuno novog softvera.
- U ranim fazama razvoja softvera pokušava se shvatiti koje značajke produkta će biti korisne korisnicima, te što im se sviđa ili ne u produktima koji trenutno koriste.

Značajke softvera

- Značajka (*engl. feature*) je djelić funkcionalnosti (npr. značajka za ispis, značajka promjene pozadine, značajka dodavanja novog dokumenta, ...).
- Prije nego se krene s razvojem produkta trebalo bi napraviti listu značajki koje će se razviti u sklopu projekta.
 - Lista značajki bi trebala biti početna točka za dizajn i razvoj produkta.

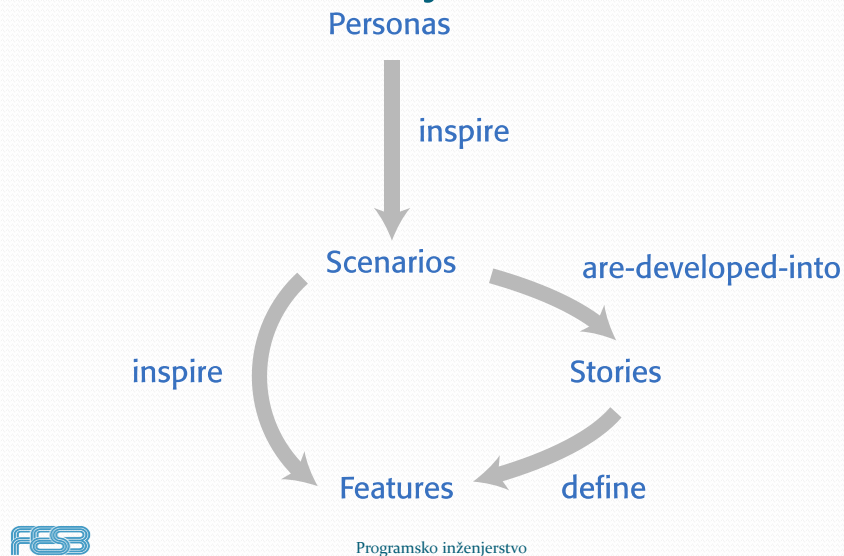


Razumijevanje korisničkih potreba

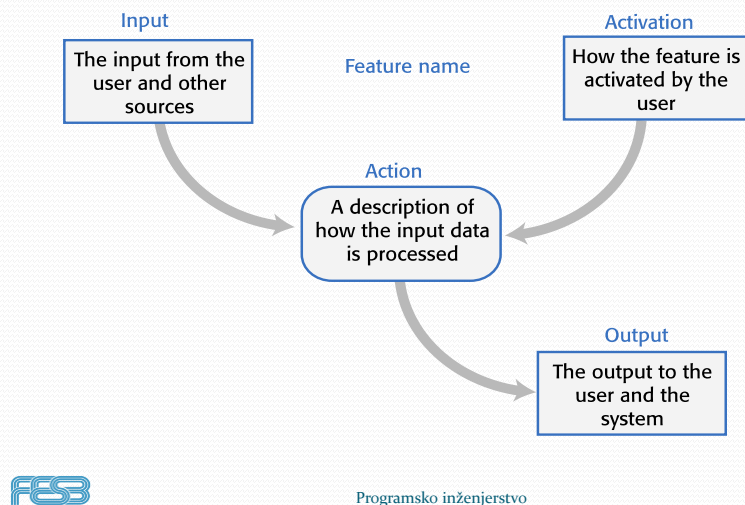
- Kod razvoja bilo kakvih produkata potrebno je izdvojiti vrijeme i razumjeti potrebe potencijalnih korisnika i kupaca tog produkta.
- Postoji čitav niz tehnika koje pomažu u razumijevanju načina na koji ljudi rade i koriste softver.
 - To mogu biti intervjui, ankete, etnografija i analize zadataka.
 - Neke od tehnika su skupe i nemaju smisla za male kompanije.
- Neformalna analiza i diskusija za vrijeme koje se postavljaju pitanja korisniku o njegovom radu, softveru koji koristi, njegovim prednostima i nedostacima je vrlo jeftina, a daje jako dobre rezultate.



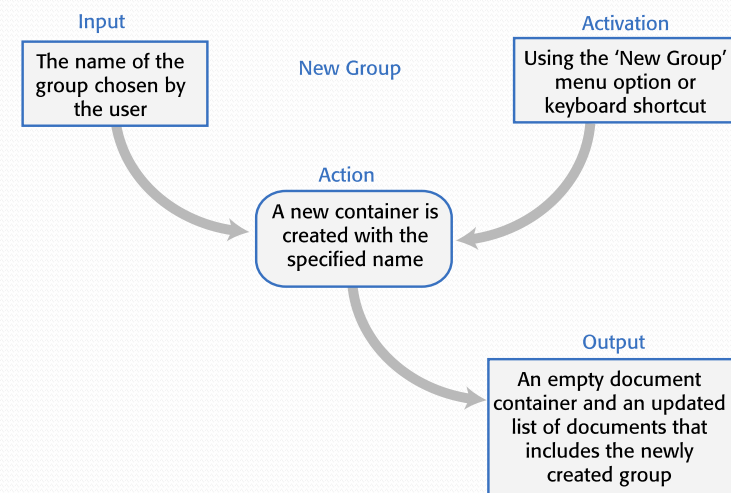
Od osoba do značajki



Opis značajki



Primjer 'New group' značajke



Programsko inženjerstvo

7

Osobe

- Razvojni tim treba biti svjestan svih mogućih potencijalnih korisnika softvera kako bi im dizajnirao korisne značajke s privlačnim sučeljem.
- Osobe (*engl. personas*) su imaginarni korisnici koji predstavljaju primjere pojedinih tipova korisnika koji će koristiti softver.
 - Npr. Ako se razvija produkt za stomatološku ordinaciju, tipovi korisnika bi bili: stomatolog, sestra na prijemnom šalteru i pacijent.
- Osobe različitih tipova korisnika olakšava programerima shvatiti što određeni tipovi korisnika žele od softvera, te na koje načine bi ga koristili.



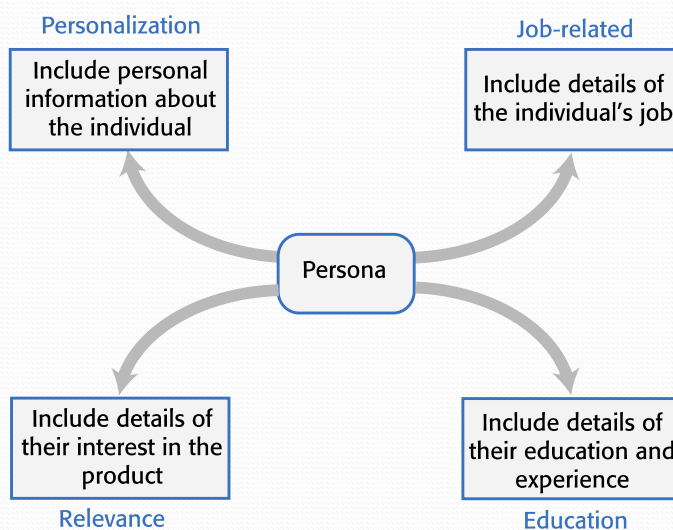
Programsko inženjerstvo

8

Opis osobe

- Osoba opisuje osnovni profil nekog tipa korisnika. Opis bi trebao biti kratak i jednostavan.
- Opis bi trebao sadržavati:
 - Pozadinu korisnika i zašto oni žele koristiti produkt,
 - Treba sadržavati razinu edukacije i potencijalna tehnička znanja i sposobnosti.
- Ovo bi trebalo pomoći u procjeni hoće li ili ne neka značajka biti korisna, razumljiva i hoće li je moći koristiti tipični korisnici.

Opis osobe



Opis osobe

- **Personalizacija**

Osobe trebaju imati ime i treba iznijeti neke od njihovih osobnih okolnosti. To je važno jer se o osobama ne bi trebalo misliti kao o ulozi nego kao o individui. Ponekad pomaže i predstavljanje osobe slikom. Neka istraživanja su pokazala da projektni timovi koriste osobe mnogo efikasnije.

- **Povezanost s poslom**

Treba navesti što ta osoba radi i što sve uključuje njen posao. Međutim ukoliko se radi o nekom poslu s kojim su svi upoznati (npr. učitelj) onda to i nije neophodno.

- **Razina edukacije**

Potrebno je opisati razinu edukacije i razinu tehničkih sposobnosti i iskustva te osobe. To je važno, pogotovo za dizajn sučelja.

- **Relevantnost**

Ukoliko je moguće, treba reći zašto bi taj korisnik bio zainteresiran za korištenje produkta i što želi raditi s tim produktom.



Primjer: Jack, a primary school teacher

- **Jack, a primary school teacher**

Jack, age 32, is a primary school (elementary school) teacher in Ullapool, a large coastal village in the Scottish Highlands. He teaches children from ages 9-12. He was born in a fishing community north of Ullapool, where his father runs a marine fuels supply business and his mother is a community nurse. He has a degree in English from Glasgow University and retrained as a teacher after several years working as a web content author for a large leisure group.

Jack's experience as a web developer means that he is confident in all aspects of digital technology. He passionately believes that the effective use of digital technologies, blended with face to face teaching, can enhance the learning experience for children. He is particularly interested in using the iLearn system for project-based teaching, where students work together across subject areas on a challenging topic.



Primjer: Emma, učiteljica povijesti

- **Emma, a history teacher**

Emma, age 41, is a history teacher in a secondary school (high school) in Edinburgh. She teaches students from ages 12 to 18. She was born in Cardiff in Wales where both her father and her mother were teachers. After completing a degree in history from Newcastle University, she moved to Edinburgh to be with her partner and trained as a teacher. She has two children, aged 6 and 8, who both attend the local primary school. She likes to get home as early as she can to see her children, so often does lesson preparation, administration and marking from home.

Emma uses social media and the usual productivity applications to prepare her lessons, but is not particularly interested in digital technologies. She hates the virtual learning environment that is currently used in her school and avoids using it if she can. She believes that face-to-face teaching is most effective. She might use the iLearn system for administration and access to historic films and documents. However, she is not interested in a blended digital/face-to-face approach to teaching.



Programsko inženjerstvo

13

Primjer: Elena, IT tehničar

- **Elena, a school IT technician**

Elena, age 28, is a senior IT technician in a large secondary school (high school) in Glasgow with over 2000 students. Originally from Poland, she has a diploma in electronics from Potsdam University. She moved to Scotland in 2011 after being unemployed for a year after graduation. She has a Scottish partner, no children, and hopes to develop her career in Scotland. She was originally appointed as a junior technician but was promoted, in 2014, to a senior post responsible for all the school computers.

Although not involved directly in teaching, Elena is often called on to help in computer science classes. She is a competent Python programmer and is a 'power user' of digital technologies. She has a long-term career goal of becoming a technical expert in digital learning technologies and being involved in their development. She wants to become an expert in the iLearn system and sees it as an experimental platform for supporting new uses for digital learning.



Programsko inženjerstvo

14

Prednosti definiranja osoba

- Glavna prednost definiranja osoba što one pomažu programerima da dobiju osjećaj tko će koristiti sustav.
- Na ovaj način članovi razvojnog tima mogu promatrati sustav iz perspektive korisnika.
 - Umjesto da programeri razmišljaju što bi oni napravili u nekoj situaciji, mogu razmišljati što bi ta osoba napravila u tom trenutku.
- Osobe olakšavaju provjeru ideja kako se ne bi uključile značajke koje nitko neće koristiti.



Scenariji

- Scenarij je tekst koji opisuje na koji način korisnik ili grupa korisnika bi mogla koristiti sustav koji se razvija.
 - To je jednostavan opis situacije u kojoj korisnik koristi neku od značajki sustava kako bi obavio neku radnju.
- Nema potrebe uključiti sve u scenarij, to nije specifikacija sustava.
- Opisi scenarija mogu biti različite dužine, od 1-2 paragrafa do cijele stranice.



Primjer: Jackov scenarij korištenja iLearn sustava za razredni projekt

Fishing in Ullapool

Jack is a primary school teacher in Ullapool, teaching 6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history archive site) to access newspaper archives and photographs. However, Jack also needs a photo-sharing site as he wants students to take and comment on each others' photos and to upload scans of old photographs that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack sends an email to a primary school teachers' group to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.

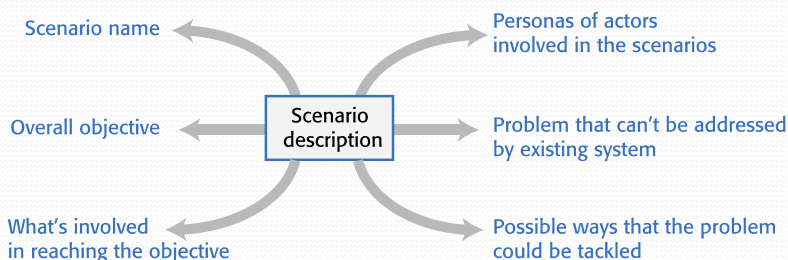
He uses the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that, when they log in, they can immediately use the system to upload photos from their phones and class computers.



Programsko inženjerstvo

18

Elementi scenarija



Programsko inženjerstvo

19

Elementi scenarija

- Naziv scenarija
- Kratka izjava o općim ciljevima:
 - U Jackovom scenariju – podržati projekt o ribarskoj industriji.
- Reference na uključene osobe (Jack) kako bi se mogle razumjeti sposobnosti i motivacija korisnika.
- Informacije o tome što je sve uključeno u obavljanje aktivnosti:
 - U Jackovom scenariju to uključuje prikupljanje informacija poput sjećanja rođaka, pristupa novinskim arhivama, ...
- Opis problema kojeg trenutni sustav ne može riješiti:
 - Mala djeca ne razumiju autorska prava i privatnost, pa dijeljenje fotografija zahtjeva da učitelj može upravljati objavom kao bi bio siguran da je sve u skladu sa zakonom.
- Opis bar jednog načina na koji se problem može riješiti:
 - U Jackovom scenariju je opisano da bi najbolji način bilo korištenje vanjskog alata koji je dizajniran za rad s djecom.



Pisanje scenarija

- Scenariji bi trebali uvijek biti pisani iz korisničke perspektive i bazirani na identificiranim osobama ili stvarnim korisnicima.
- Početna točka pisanja scenarija je stvorena osoba i trebalo bi zamisliti nekoliko scenarija za svaku od osoba.
- U idealnom slučaju scenarij bi trebao biti napisan općenito i ne bi trebao sadržavati implementacijske detalje.
 - Međutim, opis implementacije ine najčešće najlakši način za objasniti kako nešto radi.
- Kada se opisuje sustav važno je identificirati sve grupe potencijalnih korisnika.



Uključenost korisnika

- S obzirom da su scenariji pisani na jednostavan način trebalo bi korisnika uključiti u njihov razvoj.
- Najbolji pristup da članovi tima razviju imaginarni scenarij na osnovu vlastitog shvaćanja kako bi sustav se mogao koristiti i tek onda pitati korisnika je li to dobro shvaćeno ili ne.
- Korisnika se može pitati o nedovoljno jasnim stvarima i korisnik može sugerirati kako proširiti scenarij kako bi bio što realističniji.
- Pokazalo se da korisnici nisu dobri u pisanju scenarija:
 - Scenarije pišu na osnovu procedura koje se koriste u tom trenutku u radu;
 - Scenariji mogu biti predetaljni i nije moguće ih generalizirati.



Korisničke priče

- Scenariji su priče visokog nivoa o korištenju sustava. Oni bi trebali opisati niz interakcija sa sustavom, ali ne sadrže detalje tih interakcija.
- Korisničke priče su detaljniji opisi koji daju puno više detalja o jednoj stvari koju korisnik želi od sustava.
- Standardan format pisanja korisničkih priča:
 - **As a** <role>, I <want | need> **to** <do something>
 - Kao nastavnik, ja želim reći svim članovima grupe kada su dostupne nove informacije.
- Varijanta ovog načina je još i dodatak razloga zbog čega se nešto želi A variant of this standard format adds a justification for the action:
 - **As a** <role> I <want | need> **to** <do something> **so that** <reason>
 - Kao nastavnik, ja moram imati mogućnost praćenja prisutnosti učenika na nastavi kako bi se znalo jesu li bili prisutni na neophodnom broju predavanja.



Korištenje korisničkih priča u planiranju

- Korisničke priče se koriste kod planiranja.
 - U većini slučajeva *product backlog* u Scrum metodi se sastoji od korisničkih priča.
- Korisničke priče bi se trebale fokusirati na jasno definirane značajke koje se mogu implementirati unutar jednog sprinta.
- Ukoliko je priča o složenijim značajkama za koje treba nekoliko sprintova za implementaciju onda se naziva **epic**.
 - Npr. Kao osoba zadužena za održavanje sustava potrebna mi je mogućnost stvaranja sigurnosne kopije (engl. *backup*) čitavog sustava, te mogućnost vraćanja pojedinačnih aplikacija, datoteka, direktorija ili čitavog sustava.
 - Ova korisnička priča bi se trebala razbiti na više jednostavnih priča, od kojih je svaka fokusirana na određeni dio *backup* sustava.



Primjer: Korisničke priče iz Emminog scenarija

As a teacher, I want to be able to log in to my iLearn account from home using my Google credentials so that I don't have to remember another login id and password.

As a teacher, I want to access the apps that I use for class management and administration.

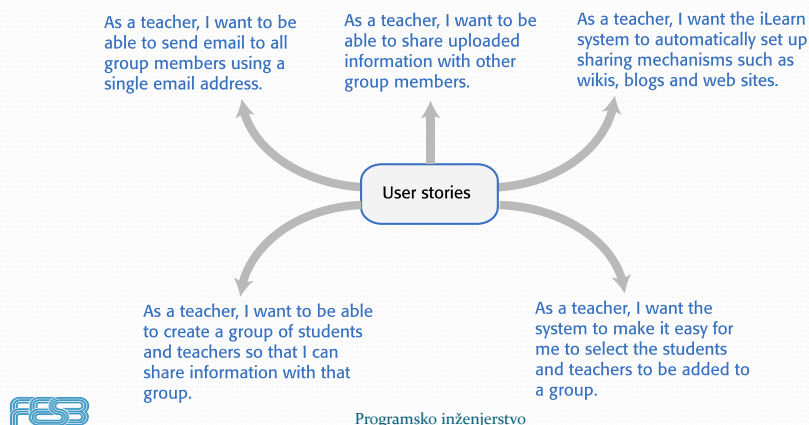
User stories

As a teacher and parent, I want to be able to select the appropriate iLearn account so that I don't have to have separate credentials for each account.



Opisivanje značajki korištenjem korisničkih priča

- Priče se mogu koristiti za opisivanje značajki produkta koji se razvija, pri tome svaka značajka može imati više priča koje opisuju kako se koristi.

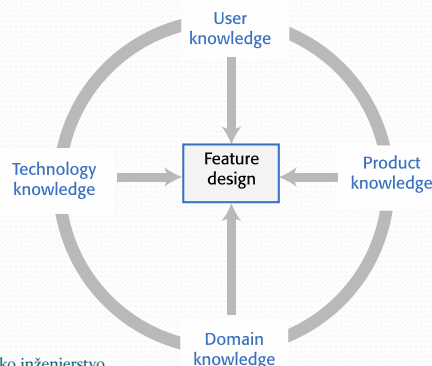


Priče i scenariji

- S obzirom da se sve funkcionalnosti opisane u scenariju predstavljaju kroz korisničke priče, pitanje je treba li uopće pisati scenarije?
- Scenariji su prirodniji i korisniji iz sljedećih razloga:
 - Scenariji su prirodniji jer opisuju što korisnik sustava zbilja radi s tim sustavom, zbog čega je ljudima to lakše razumjeti nego jednostavne izjave što netko želi ili treba.
 - Ukoliko se priča sa stvarnim korisnikom lakše je pratiti informacije zapisane u obliku scenarija jer ljudi najčešće ne koriste način koji se koristi za pisanje priča.
 - Scenariji daju više informacija o kontekstu korištenja sustava. Priče bi trebale biti kratke izjave, pa ako se doda informacija o korištenju značajke sustava prestaju biti kratke.

Identifikacija i dizajn značajki

- U početnoj fazi dizajna produkta cilj je stvoriti listu značajki koje definiraju produkt.
- Značajka bi trebala omogućiti korisnicima pristup i korištenje funkcionalnosti sustava, tako da lista značajki definira sve funkcionalnosti sustava.



Programsko inženjerstvo

32

Znanje potrebno za dizajn značajki

- **Znanje korisnika**
Mogu se koristiti scenariji i korisničke priče kao bi se tim informirao o tome što korisnici žele i kako mogu koristiti značajke sustava.
- **Znanje o produktu**
Ukoliko ne postoji iskustvo rada sa sličnim produktima možda bi trebalo provesti istraživanje što takvi produkti generalno nude. Ponekad treba replicirati postojeće značajke jer pružaju osnovnu funkcionalnost koja je uvijek potrebna.
- **Znanje o domeni**
To je znanje o području tj. domeni za koju se razvija produkt (financije, rezervacija događaja, ...). Poznavanjem domene lakše je osmisлити nove inovativne ideje kako pomoći korisnicima.
- **Znanje o tehnologiji**
Ukoliko je tim upoznat s najnovijim tehnološkim dostignućima moguće je razviti značajke koje koriste prednosti tih tehnoloških dostignuća.



Programsko inženjerstvo

34

Faktori u dizajnu seta značajki

- Jednostavnost
- Funkcionalnost
- Kontrola
- Automatizacija
- Poznate značajke
- Nove značajke



Programsko inženjerstvo

35

Balansiranje između različitih faktora

- **Jednostavnost i funkcionalnost**
 - Potrebno je uspostaviti ravnotežu između razvoja jednostavnog sustava, lakog za korištenje i pri tome uključiti dovoljno funkcionalnosti kako bi zadovoljio korisnike s različitim potrebama.
- **Poznate i nove značajke**
 - Korisnici vole da novi sustav podržava sve ono na što su do sada navikli, ali istovremeno treba sadržavati i nove značajke kako bi korisnici napustili stari prešli na novi sustav.
- **Automatizacija i kontrola**
 - Neki korisnici da softver automatski obavlja posao za njih, drugi preferiraju kontrolu. Stoga je potrebno dobro promisliti što automatizirati, na koji način i na koji način korisnici mogu prilagoditi automatizaciju da više odgovara njihovim potrebama.



Programsko inženjerstvo

36

Feature creep

- *Feature creep* se dešava kada se na zahtjev korisnika dodaju nove značajke bez da se razmisli trebaju li one stvarno i mogu li se realizirati na druge načine.
- Previše značajki otežava razumijevanje i korištenje produkta.
- *Feature creep* se dešava iz tri razloga:
 - Product menadžeri ne žele reći 'ne' korisnicima kada traže određene značajke.
 - Programeri kopiraju značajke iz sličnih produkata.
 - Produkt uključuje značajke za iskusne i neiskusne korisnike.



Programsko inženjerstvo

37

Kako izbjeći *feature creep*

Does this feature really add anything new or is it simply an alternative way of doing something that is already supported?

Is this feature likely to be important to and used by most software users?

Feature questions

Can this feature be implemented by extending an existing feature rather than adding another feature to the system?

Does this feature provide general functionality or is it a very specific feature?



Programsko inženjerstvo

38

Identifikacija značajki

- Značajke se mogu identificirati direktno iz projektne ideje ili iz scenarija.
- Jednostavno se mogu naglasiti fraze u opisu kako bi se identificirale značajke koje će se uključiti u softver.
 - Označavaju se značajke koje su potrebne da bi podržale korisnikove akcije, a identificiraju se aktivnim glagolima poput koristiti i odabrati.



Primjer: Ideja projekta iLearn sustava

- **FOR** teachers and educators **WHO** need a way to help students use web-based learning resources and applications, **THE** iLearn system is an open learning environment **THAT** allows the set of resources used by classes and students to be easily configured for these students and classes by teachers themselves.
- **UNLIKE** Virtual Learning Environments, such as Moodle, the focus of iLearn is the learning process itself, rather than the administration and management of materials, assessments and coursework. **OUR** product enables teachers to create subject and age-specific environments for their students using any web-based resources, such as videos, simulations and written materials that are appropriate



Značajke iz ideje produkta

- 1) Značajka koja omogućava korisnicima pristup i korištenje postojećih web resursa;
- 2) Značajka koja dozvoljava da istovremeno postoji više različitih instanci sustava;
- 3) Značajka koja omogućava korisnicima konfiguriranje instance prema vlastitim potrebama.



Programsko inženjerstvo

41

Primjer: Jackov scenarij

Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. *Students use an iLearn wiki to gather together fishing stories and SCRAN (a history archive) to access newspaper archives and photographs.* However, Jack also needs a photo-sharing site as he wants *pupils to take and comment on each others' photos and to upload scans of old photographs* that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack *sends an email to a primary school teachers' group*, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics *is not integrated with the iLearn authentication service*, he sets up a teacher and a class account with KidsTakePics.

He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that when they log in, they can immediately use the system to upload photos from their phones and class computers.



Programsko inženjerstvo

42

Primjer značajki iz Jackovog scenarija

- 1) Wiki grupa za pisanje;
- 2) Pristup SCARN povijesnoj arhivi
- 3) Značajku za kreiranje i pristup e-mail grupe.
- 4) Značajku za integraciju aplikacija s iLearn autentifikacijskim servisom.



Lista značajki

- Krajnji rezultat procesa identifikacije značajki je lista značajki koja će se koristiti za dizajn i implementaciju produkta.
- Ne postoji potreba za velikom količinom detalja u ovoj fazi. Detalji se dodaju u trenutku kada se krene u implementaciju značajke.
- Značajke se mogu opisati korištenjem strukturiranog opisa ili korisničkih priča.



Primjer: iLearn značajka za autentifikaciju

Description

Authentication is used to identify users to the system and is currently based on a login id/password system. Users may authenticate themselves using their national user id and a personal password or may use their Google or Facebook credentials.

iLearn
authentication

Constraints

All users must have a national user id and system password that they use for initial system authentication. They may then link their account with their Google/Facebook account for future authentication sessions.

Comments

Future authentication mechanisms may be based on biometrics and this should be considered in the design of the system.



Programsko inženjerstvo

45

Kraj

Programsko inženjerstvo

48