

Projektiranje informacijskih sustava

SDLC faza dizajna – odabir strategije

Ak. god. 2011/2012

Dizajn

- U fazi dizajna bi sve informacije o sustavu prikupljene u fazi projektiranja i analize trebalo pretočiti u nacrt (*blueprint*) budućeg sustava koji omogućava “jednostavnu” implementaciju sustava.
- U fazi dizajna se definira na koji način (*how?*) će se izraditi sustav koji je identificiran u fazi analize (*what?*).

Dizajn

- U fazi dizajna se poslovni zahtjevi (ili korisnički zahtjevi) identificirani u fazi analize prevode u sistemske zahtjeve (*system requirements*).
- Sistemske zahtjeve tehnički opisuju sustav koji se izrađuje.

Dizajn

- Dizajn se sastoji od niza aktivnosti, svaka aktivnost rezultira određenim dokumentima.
- Konačni rezultat ove faze je specifikacija sustava (*system specification*) koja se sastoji od cijelog niza dokumenata (dizajn sučelja, fizički model podataka, ... koji su rezultat pojedinih aktivnosti tijekom dizajna) i koja točno i detaljno opisuje implementaciju sustava.

Dizajn

Koraci u fazi dizajna	Rezultati
Odabire strategije izrade sustava	Alternativna matrica
Dizajn arhitekture sustava	Dokumentacija arhitekture
Odabir hardvera i softvera	Specifikacija hardvera i softvera
Dizajn korisničke navigacije, ulaza i izlaza	Dizajn sučelja
Prebacivanje logičkih modela procesa u fizičke	Fizički modeli procesa
Dizajn programa koji izvode procese	Specifikacije dizajna programa
Prebacivanje logičkih modela podataka u fizičke	Fizički modeli podataka
Revizija CRUD matrice	CRUD matrica
Dizajn pohrane podataka	Dizajn pohrane podataka



specifikacija sustava

Dizajn

- Prvi korak u ovoj fazi je odabiranje strategije izrade sustava između tri moguće strategije:
 1. Razvoj vlastite aplikacije (*custom application in-house*)
 2. Kupovina gotove (*packaged*) aplikacije i prilagodba klijentovim zahtjevima
 3. Razvoj aplikacije izvan kompanije (*outsourcing*)

Razvoj vlastite aplikacije

- Razvoj vlastite aplikacije podrazumijeva razvoj aplikacije unutar vlastite kompanije.
- Prednosti razvoja aplikaciju unutar kompanije su:
 - potpuni nadzor nad aplikacijom što omogućava fleksibilnost kod razvoja aplikacije
 - lako povezivanje sa drugim vlastitim aplikacijama unutar kompanije
 - razvoj vlastitog kadra unutar kompanije

Razvoj vlastite aplikacije

- Nedostatci razvoja aplikaciju unutar kompanije su:
 - dugo trajanje dok se sustav ne počme koristiti i veliki trud za njegov razvoj (effort)
 - nedostatak poznavanja novih tehnologija
 - teže ograničavanje dodatnih korisničkih zahtjeva
 - veća cijena i rizik

Kupovina gotove aplikacije

- Prednosti kod kupovine gotovih aplikacija su:
 - danas su dostupne za većinu uobičajnih poslovnih potreba (knjigovodstvo, skladištenje, ...), čak do ERP (*Enterprise resource planning*) aplikacija za automatizaciju cjelokupnog poslovanja poput SAP, Oracle i sličnih sustava.
 - gotove aplikacije su dobro testirane, pouzdane.
 - brza isporuka i obično manja cijena no što bi koštao razvoj takve aplikacije
- Nedostatak gotovih aplikacija je što rijetko u potpunosti odgovara potrebama klijenta.

Kupovina gotove aplikacije

- Gotove aplikacije su obično napravljene tako da ih je moguće prilagođavati potrebama korisnika.
- Obično je moguće postavljati različite parametre sustava prema zahtjevima korisnika i sl.
- Ukoliko aplikacija nema takve mogućnosti, mogu se napraviti i dodatni dijelovi aplikacije u obliku dodatnih komponenti, add-on-ova i sl. Ti se dodatni dijelovi nazivaju “workaround”.
- Može postojati problem da li će prodavatelja aplikacije htjeti podržavati u budućim verzijama aplikacija ta naša pomoćna rješenja.

Kupovina gotove aplikacije

- Često je potrebno gotovu aplikaciju povezati sa postojećim sustavim.
- Izrada sustava se tada svodi na integraciju gotovih sustava i vlastitih (*legacy*) sustava (*system integration*).
- Povezivanje sustava se obično radi na način da izlazni podaci iz jednog sustava predstavljaju ulazne podatke u drugi sustav.
- Ponekad je potrebno preformatirati podatke iz jednog u drugi sustav.

Razvoj aplikacije izvan kompanije

- Razvoj aplikacija izvan kompanije (*outsourcing*) se svodi na pronalaženje kompanije koja može razviti sustav prema specifikacijama definiranim u prethodnim fazama SDLC-a.
- Danas se često koriste i tzv. pružatelji usluga (ASP - *application service provider*) koji aplikaciju ili servis potreban kompaniji pružaju preko Interneta.
- Npr. vrlo često se koristi *outsourcing* za web hosting.

Razvoj aplikacije izvan kompanije

- Mogući rizici razvoj aplikacija izvan kompanije:
 - Gubitak povjerljivih informacija
 - Gubitak kontrole nad daljnjim razvojem
 - Gubitak mogućnosti učenja

Razvoj aplikacije izvan kompanije

- Postoje tri tipa *outsourcing* ugovora:
 1. Vrijeme i raspored (*time and arrangement deal*) nisu strogo određeni, već je ugovorom predviđeno da se plati onoliko utrošenog vremena koliko *outsourcing* kompanija navede da je potrošila.
 2. Ugovor sa fiksnom cijenom (*fixed-price contract*) na početku definira cijenu koštanja proizvoda koji se isporučuje.
 3. Ugovor sa dodanom vrijednošću (*value-added contract*) definira isplatu *outsourcing* kompaniji kroz dio profita iz sustava koji je kompanija razvila.

Razvoj aplikacije izvan kompanije

- Kod *outsourcinga* potrebno je cijelo vrijeme biti u toku sa razvojem aplikacije.
- Smjernice za dobar *outsourcing* su:

- Keep the lines of communication open between you and your outsourcer.
- Define and stabilize requirements before signing a contract.
- View the outsourcing relationship as a partnership.
- Select the vendor, developer, or service provider carefully.
- Assign a person to manage the relationship.
- Don't outsource what you don't understand.
- Emphasize flexible requirements, long-term relationships, and short-term contracts.

ODABIR STRATEGIJE DIZAJNA

- Prilikom odabira strategije potrebno je uzeti u obzir:
 1. Poslovne potrebe – da li već postoji gotova aplikacija? Ako ne jedini način je razvoj nove aplikacije.
 2. “In-house” iskustvo – da li unutar kompanije već postoji iskustvo potrebno za razvoj sustava. Npr. da li je već razvijana web aplikacija?
 3. Znanje – da li unutar kompanije postoji znanje potrebno za razvoj sustava i da li je to znanje kompaniji uopće potrebno. Npr. uzeti *outsourcing* sistemca.
 4. Upravljanje projektima – da li postoji znanje i iskustvo u upravljanju projektima.
 5. Vremenski okvir – koliko brzo sustav treba biti isporučen.

ODABIR STRATEGIJE DIZAJNA



	When to Use Custom Development	When to Use a Packaged System	When to Use Outsourcing
Business need	The business need is unique	The business need is common	The business need is not core to the business
In-house experience	In-house functional and technical experience exists	In-house functional experience exists	In-house functional or technical experience does not exist
Project skills	There is a desire to build in-house skills	The skills are not strategic	The decision to outsource is a strategic decision
Project management	The project has a highly skilled project manager and a proven methodology	The project has a project manager who can coordinate vendor's efforts	The project has a highly skilled project manager at the level of the organization that matches the scope of the outsourcing deal
Time frame	The time frame is flexible	The time frame is short	The time frame is short or flexible

Razvoj alternativne matrice

- Ako se odlučite za vlastiti razvoj aplikacije koje alate i tehnologije je potrebno koristiti za vlastiti razvojni projekt?
- Ako se odlučite za kupnju gotove aplikacije kako odabrati onu aplikaciju ili isporučitelja aplikacije koji će zadovoljavati potrebe projekta?
- Kada se aplikacija razvija preko vanjskih suradnika koje kompanije tj. pružatelji usluga mogu napraviti aplikaciju koja će zadovoljavati potrebe projekta?

Razvoj alternativne matrice

- Odgovor na prethodna pitanja može se dobiti izradom alternativne matrice (*alternative matrix*).
- Alternativna matrica daje kombinaciju nekoliko analiza izvedivosti (*feasibility analysis*) u jednoj matrici za svako od mogućih rješenja tj. za vlastiti razvoj, za kupnju gotove aplikacije i za razvoja aplikacije izvan kompanije.

Razvoj alternativne matrice

- Potrebno je uključiti tehničku, financijsku i organizacijsku izvedivost.
- Pojedinačnim kriterijama izvedivosti se dodjeljuje značaj odnosno težinski faktor koji će indicirati relativnu važnost kriterija.
- Nakon toga se rješenjima dodjeljuje jakost prema navednim kriterijima kako bi se omogućio proračun “jakosti” alternativnih rješenja.

Razvoj alternativne matrice

[Primjer](#)

Evaluation Criteria	Relative Importance (Weight)	Alternative 1: Custom Application using VB.NET	Score (1-5)*	Weighted Score	Alternative 2: Custom Application using Java	Score (1-5)*	Weighted Score	Alternative 3: Packaged Software Product ABC	Score (1-5)*	Weighted Score
Technical Issues:		↑			↑			↑		
Criterion 1	20		5	100		3	60		3	60
Criterion 2	10		3	30		3	30		5	50
Criterion 3	10		2	20		1	10		3	30
Economic Issues:										
Criterion 4	25	Supporting	3	75	Supporting	3	75	Supporting	5	125
Criterion 5	10	Information	3	30	Information	1	10	Information	5	50
Organizational Issues		↓			↓			↓		
Criterion 6	10		5	50		5	50		3	30
Criterion 7	10		3	30		3	30		1	10
Criterion 8	5		3	15		1	5		1	5
TOTAL	100	↓		350	↓		270	↓		360

Razvoj alternativne matrice

- Obično se uzima da je ukupna vrijednost svih elemenata 100, a onda se pojedinim pristupima dodjeljuje faktor između 1-5 (1 najgori pristup, 5 najbolji pristup) koji se množi sa vrijednošću elementa i na taj način se izračuna kvantitativna vrijednost svakog pristupa.
- Problem je što je dodjeljivanje vrijednosti pojedinom elementu iz analize izvedivosti kao i faktora pojedinom pristup izuzetno subjektivno. Zato se često kombiniraju alternativne matrice koje su izradili različiti sistem analitičari.