
Greitas programų kūrimas

Tikslai

- Paaiškinti kaip iteratyvus, palaipsnis vystymo procesas leidžia greičiau pateikti naudingesnę programinę įrangą.
- Aptarti vikraus (Agile) vystymo metodų esmę.
- Paaiškinti ekstremalaus (extreme) programavimo principus ir praktinį naudojimą.
- Paaiškinti prototipų kūrimo vaidmenį programinės įrangos kūrimo procese.

Temos

- Agile metodai
- Extremalus programavimas
- Greitas programinės įrangos kūrimas
- Programinės įrangos prototipų kūrimas

Greitas programų kūrimas

- Greitai besikeičianti verslo aplinka verčia verslą reaguoti į naujas galimybes ir konkurenciją.
- Tam reikia greitai sukurti naują programinę įrangą. Kokybė tokiu atveju nėra svarbiausias programinės įrangos sistemų reikalavimas.
- Verslas gali pasitenkinti žemesnės kokybės programomis, jei galima greitai gauti naujas sistemos funkcijas.

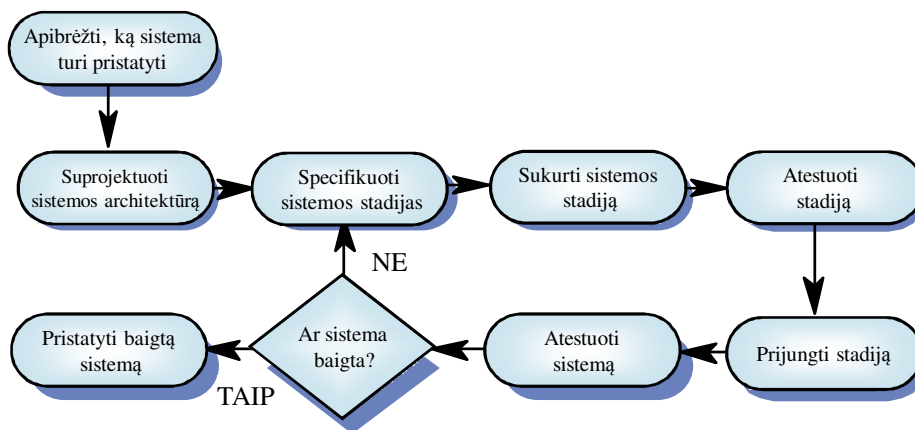
Reikalavimai

- Besikeičiančioje aplinkoje dažnai neįmanoma nustatyti stabilių sistemų reikalavimų.
- Krioklio kūrimo modelis tampa nepraktišku ir tik iteracinis specifikacijų bei programų kūrimo metodas leidžia greitai gauti programinę įrangą.

Greito kūrimo proceso charakteristikos

- Specifikavimo, projektavimo, realizavimo procesai vyksta lygiagrečiai. Nėra detalios specifikacijos ir projektinė dokumentacija yra minimali.
- Sistema yra kuriama per eilę stadijų. Galutiniai vartotojai įvertina kiekvieną stadiją ir teikia pasiūlymus vėlesnėms stadijoms.
- Sistemos vartotojo sąsaja paprastai kuriama naudojant interaktyvią kūrimo sistemą.

Iteracinis kūrimo procesas



©Ian Sommerville 2007

Software Engineering, 8th edition, Chapter 17

Slide 7

Palaipsninio kūrimo privalumai

- **Pagreitintas paslaugų pateikimas vartotojui.**
Kiekviena stadija vartotojui pateikia tą funkcionalumą, kurio prioritetą tuo metu yra didžiausias.
- **Vartotojų įtraukimas į sistemos kūrimą.**
Vartotojai turi būti įtraukti į sistemos kūrimą. Sistema labiau tenkins vartotojų reikalavimus ir vartotojai labiau pasitikės sistema.

©Ian Sommerville 2007

Software Engineering, 8th edition, Chapter 17

Slide 8

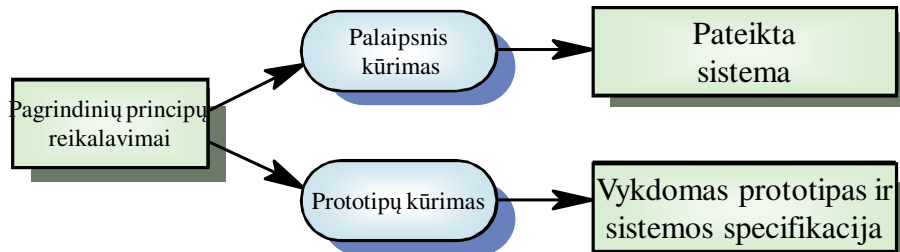
Palaipsninio kūrimo problemos

- **Valdymo problemos**
 - Sudėtinga spręsti apie pažangą ir sudėtinga rasti problemas, nes nėra dokumentacijos apie atliktus darbus.
- **Kontraktinės problemos**
 - Normalus kontraktas gali turėti specifikaciją, o be jos turi būti naudojamos kitos kontrakto formos.
- **Įteisinimo problemos**
 - Be specifikacijos neaišku, su kuria sistema reikia lyginti.
- **Palaikymo problemos**
 - Nuolatinė kaita pakenkia programos struktūrai ir tolimesni pakeitimai bei naujų reikalavimų įvertinimas yra žymiai brangesni.

Prototipų kūrimas

- Kai kurioms didelėms sistemoms palaipsninis iteracinis kūrimas gali būti nepraktiškas, ypač kai daug komandų dirba skirtingose vietose.
- Prototipų naudojimas, kai sukuriama eksperimentinė sistema kaip pagrindas reikalavimų suformavimui. Tokia sistema išmetama, kai susitariama dėl sistemos specifikacijų.

Palaipsninis ir Prototipų kūrimas



Prieštaringi tikslai

- Palaipsninio vystymo tikslas yra **pateikti veikiančią sistemą** jos galutiniams vartotojams. Kūrimas pradedamas nuo tų reikalavimų, kurie geriausiai suprantami.
- Išmetamų prototipų kūrimo tikslas yra **patvirtinti arba gauti sistemos reikalavimus**. Prototipų kūrimo procesas yra pradedamas nuo reikalavimų, kurie yra mažiausiai suprantami.

Temos

- Agile metodai
- Extremalus programavimas
- Greitas programinės įrangos kūrimas
- Prototipų naudojimas

Agile metodai

- Nepasitenkinimas projektavimo metodų pridėtinėm išlaidom iššaukė agile metodų sukūrimą. Šie metodai:
 - Daugiau orientuojasi į programos kodą negu į projektą;
 - Remiasi iteratyviu programinės įrangos kūrimo metodu;
 - Ketina greit pateikti veikiančias programas ir greit prisiderina prie besikeičiančių reikalavimų.
- Agile metodai labiausiai tinka mažoms bei vidutinėms verslo sistemoms ir personalinių kompiuterių produktams.

Agile metodų principai

Principai	Aprašymas
Vartotojo įtraukimas	Vartotojas turi būti artimai įtrauktas vystymo metu. Jų tikslas yra suformuoti naujus sistemų reikalavimus ir suteikti jiems prioritetus bei vertinti sistemos iteracijas.
Palaipsninis pateikimas	Programinė įranga vystoma palaipsniui vartotojui nustatant reikalavimus, kurie įtraukiami kiekvienoje stadijoje.
Žmonės ne procesas	Kūrimo komandos įgūdžiai turėtų būti pripažįstami ir naudojami. Komanda privačlo turėti teisę savarankiškai nuspręsti, kokius metodus naudos PĮ kūrimui.
Įtraukti pakeitimus	Nesitikėti, kad pakeitimų nebus, ir projektuoti sistemą taip, kad ją būtų galima nesunkiai pritaikyti naujiems pakeitimams.
Siekti paprastumo	Orientuotis į paprastumą: tiek kuriant PĮ, tiek taikant kūrimo procesus. Kiek įmanoma, aktyviai dirbti, kad panaikinti sistemos sudėtingumą.

Agile metodų problemos

- Gali būti sunku ilgą laiką išlaikyti vartotojų, įtrauktų į procesą, susidomėjimą.
- Komandos nariai gali būti nepratę prie tokio intensyvaus darbo ir nepakankamai bendrauti su tiktais komandos nariais.
- Sudėtinga priskirti prioritetus ten, kur yra keli kuriama PĮ suinteresuoti asmenys – kiekvienas iš jų norės, kad būtų realizuoti būtent jiems reikalingi reikalavimai.
- Paprastumo siekis reikalauja papildomo darbo.
- Gali būti problematiškas kontrakto sudarymas lyginant su kitais iteratyvaus kūrimo metodais.

Temos

- Agile metodai
- Ekstremalus programavimas
- Greitas programinės įrangos kūrimas
- Prototipų naudojimas

Ekstremalus programavimas

- Tikriausiai geriausiai žinomas ir plačiausiai naudojamas agile metodas.
- Ekstremalus programavimas naudoja 'ekstremalų' iteratyvaus vystymo būdą:
 - Naujos versijos gali būti sukuriamos kelis kartus per dieną;
 - Naujos stadijos pateikiamos vartotojams per dvi savaites;
 - Kiekviena konstrukcija (build) pilnai testuojama ir priimama ,jei testas sėkmingas.

Ekstremalaus programavimo praktika

Palaiapsnis planavimas	Reikalavimai yra surašomi į aprašų korteles (Story Card) ir įtraukiami į išleidžiamą versiją aprašai nustatomi pagal turimą laiką ir jų santykinį prioritetą. Kūrėjai suskaido aprašus į 'Užduotis'.
Mažos versijos	Pradžioje išvystomas minimalus reikalingas funkcionalumo rinkinys, turintis verslo vertę. Sistemos versijos yra dažnos ir pastoviai papildo pirmą versiją.
Paprastas projektavimas	Vykdomas projektavimas turi atitikti esamus reikalavimus ir ne daugiau.
Pirminio testo kūrimas	Automatinis testų šablonas naudojamas naujo funkcionalumo testų rašymui dar prieš tai, kai realizuojamos naujos funkcijos.
Pertvarkymas	Visi programuotojai iš karto nuolat pertvarko kodą, jeigu kode reikalingi pakeitimai. Dėl to kodas būna paprastas ir palaikomas.

Ekstremalaus programavimo praktika

Porinis programavimas	Kūrėjai dirba porose, tikrindami vieni kitų darbą bei teikdami palaikymą kokybiškų darbų atlikimui.
Kolektyvinė kodo nuosavybė	Kūrėjų poros dirba visose sistemos srityse ir visi kūrėjai valdo visą kodą. Kiekvienas gali pakeisti bet ką.
Nenutrūkstamas integravimas	Kai tam tikras darbas pabaigiamas, jis integruojamas į visą sistemą. Po kiekvienos tokios integracijos, sistemoje turi būti atlikti visi modulio (Unit) testai.
Normalus tempas	Dideli viršvalandžių kiekiai nėra priimtini, nes dažniausiai tai baigiasi prasta kodo kokybe ir vidutiniu kūrimo laiku (o reikia greitai)
Vartotojai vietoje	Sistemos galutinio vartotojo atstovas turi būti visada pasiekiamas kūrėjų komandai. Ekstremalaus programavimo procese vartotojas yra kūrimo komandos narys ir yra atsakingas sistemos reikalavimų pateikimui kūrėjų komandai.

Ekstremalaus ir Agile programavimo principai

- Palaipsnis kūrimas vyksta per smulkias ir dažnas sistemos versijas.
- Vartotojų įtraukimas suprantamas kaip darbas prie projekto visą darbo dieną.
- Svarbiausia yra žmonės, o ne procesas. Kadangi visi programuotojai gali pataisyti bet kurią programos dalį, tai kūrimo laikas sutrumpėja.
- Pakeitimai vykdomi dėka reguliarių sistemos versijų.
- Palaikymo paprastumas per pastovų kodo pertvarkymą.

Reikalavimų scenarijai

- Ekstremaliam programavime reikalavimai išreiškiami kaip vartotojo scenarijai ar vartotojų aprašai.
- Jie yra surašyti ant kortelių ir kūrimo komanda juos suskaido į vykdymo uždavinius. Pagal šiuos uždavinius sudaromi tvarkaraščiai ir įvertinama kaina.
- Pagal scenarijų prioritetus ir suplanuotą tvarkaraštį vartotojas parenka, kuriuos vartotojo scenarijus reikia įtraukti į kitą versiją.

Aprašo kortelė dokumentų perkėlimui

Straipsnio perkėlimas ir spausdinimas

Pradžioje jūs pasirenkate norimą straipsnį iš pateiktų sąrašė. Jūs turite pranešti sistemai, kaip už jį mokėsite: prenumeruodami, per firmos sąskaitą ar kreditine kortele.

Toliau iš sistemos gaunate ir pildote autorinių teisių formą ir, kai sutinkate su sąlygomis, pasirinktas straipsnis perkeliamas į jūsų kompiuterį.

Toliau pasirenkate spausdintuvą ir straipsnio kopija išspausdinama. Sistemai pranešate, ar spausdinimas buvo sėkmingas.

Jei straipsnis yra tik spausdinamas, jūs negalite laikyti PDF rinkmenos. Ji automatiškai ištrinama iš kompiuterio.

Ekstremalus programavimas ir pakeitimai

- Paprastai programų inžinerijoje siekiama projektuoti lengvai keičiamas programas. Tikslinga paskirti tam tikrą laiką ir pastangas numatant pakeitimus, nes tai vėliau sumažins išlaidas per programos gyvavimo ciklą.
- Tačiau ekstremaliame programavime kodo tobulinimas yra pastovus, nes reguliarus kodo pertvarkymas ir tobulinimas leidžia palengvinti pakeitimų realizavimą.

Testavimas ekstremaliame programavime

- Pirmiausia sukuriami testai.
- Palaipsninis testų vystymas pagal scenarijus.
- Vartotojų įtraukimas į testų kūrimą ir atestavimą.
- Naudojami automatizuoti testų įrankiai visų komponentų testavimui kiekvieną kartą kai tik sukurama nauja versija.

Uždavinių kortelės dokumentų perkėlimui

Task 1: Implement principal workflow

Task 2: Implement article catalog and selection

Task 3: Implement payment collection

Payment may be made in 3 different ways. The user selects which way they wish to pay. If the user has a library subscription, then they can input the subscriber key which should be checked by the system. Alternatively, they can input an organisational account number. If this is valid, a debit of the cost of the article is posted to this account. Finally, they may input a 16 digit credit card number and expiry date. This should be checked for validity and, if valid a debit is posted to that credit card account.

Testinių atvejų aprašymas

Test 4: Test credit card validity

Input:

A string representing the credit card number and two integers representing the month and year when the card expires

Tests:

Check that all bytes in the string are digits
Check that the month lies between 1 and 12 and the year is greater than or equal to the current year.
Using the first 4 digits of the credit card number, check that the card issuer is valid by looking up the card issuer table. Check credit card validity by submitting the card number and expiry date information to the card issuer

Output:

OK or error message indicating that the card is invalid

Programavimas poromis

- Ekstremaliai programavime programuotojai dirba poromis, kartu kurdami kodą.
- Tai padeda gauti bendros nuosavybės kodą ir dalintis žiniomis komandoje.
- Taip vykdomas neformalus peržiūros procesas ir kiekviena kodo eilutė peržiūrima daugiau kaip vieno asmens.
- Tai skatina kodo pertvarkymą, nes visa komanda gali turėti iš to naudos.
- Matavimai parodė, kad porų programavimo našumas yra panašus kaip dviejų žmonių dirbančių nepriklausomai.

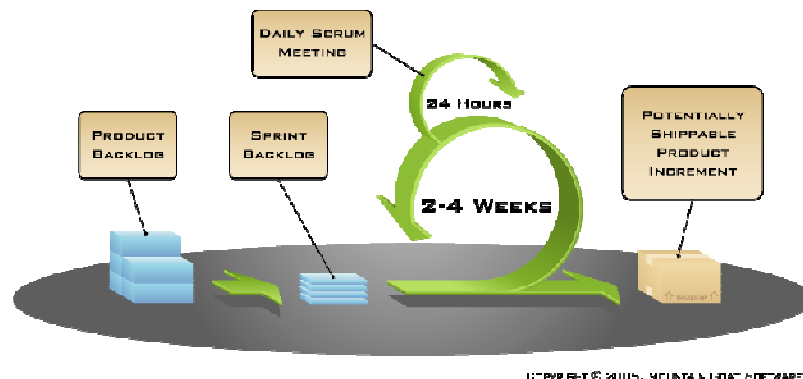
Temos

- Vikrūs metodai
- Extremalus programavimas
- Greitas programinės įrangos kūrimas
- Prototipų naudojimas

SCRUM

- Viena iš ekstremalaus programų kūrimo projektų valdymo metodikų:
 - <http://www.mountangoatsoftware.com/topics/scrum>
 - [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))

SRUM procesas



Kas yra kas...

- Product backlog?
- Sprint backlog?
- Sprint?
- Daily sprint meeting?
- Working increment of the software ?

Kas yra kas – Product backlog

- Tai pilnas reikalavimų sąrašas kuriamam produktui.
- Naudojant SCRUM metodiką, nenaudojami ilgi reikalavimų specifikacijos ar projektavimo dokumentai.
- Paprastai projekto komanda ir projekto vadovas išrašo tuos reikalavimus, kurie jų manymu yra paprasti ir suprantami.
- Sąrašas gaunasi daug didesnis, negu kad reikia vienai iteracijai.
- “Product backlog” yra nuolat pildomas, ir tai nėra jokia tragedija. Priešingai, pildyti šitą sąrašą skatinami ir komandos nariai, ir galutinio produkto vartotojai.

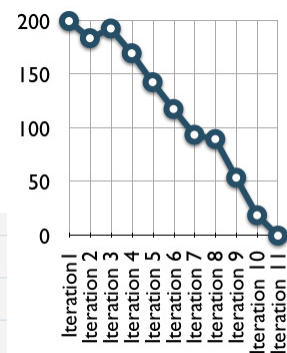
	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Kas yra kas – Sprint backlog

- Sprint backlog – tai darbų sąrašas, kuriuos SCRUM komanda ruošiasi padaryti per einamąją iteraciją, vadinamą “sprint”.
- Visi šie darbai (ir progresas) dažniausiai vaizduojami grafiškai.

Kas yra kas – Sprint backlog

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	



Kas yra kas – Sprint

- Tai einamoji iteracija, paprastai nuo 2 iki 5 savaičių
- Sprint susideda iš tokių etapų:
 - Sprint planavimo susitikimo
 - Kasdieninių susirinkimų, kurių metu sprendžiamos iškilusios problemos
 - Sprint peržiūros (review) susitikimo, kada klientams ir kt. asmenims parodome, ką padarėme
 - Sprint apžvalgos (retrospective) susitikimo užbaigus iteraciją, jame dalyvauja tik mūsų komanda.

SCRUM komanda

- SCRUM master:
 - tai nėra projekto vadovas, tai greičiau koordinatorius, kuris prižiūri, kad komanda dirbtų pagal taisykles, organizuoja dienišius susitikimus, padeda šalinti atsiradusias kliūtis.
- SCRUM komanda:
 - neįeina tokios tradicinės rolės, kaip programuotojas, testuotojas, architektas ir pan.
 - Visi dirba kartu prie vieno kodo gabalo.
 - Sukuriama “bendrums” nuotaika, kad ne kažkas vienas padarė, bet “mes visi” padarėme.
 - 5-9 žmonės optimaliausia

Daily SCRUM meeting

- Ką padarei vakar?
- Ką darysi rytoj?
- Ar yra kokių nors sunkumų, kurie tau kliudo/kliudys užbaigti planuotus darbus?
- Tipinės kliūtys:
 - Sugedo mano ir man reikia naujo.
 - Aš vis dar neturiu programinės įrangos (knygos, kompiuterio, ...), kurią užsaciau dar prieš mėnesį.
 - Man reikia pagalbos “debuginant” šią problemą.
 - Man nesiseka išsiaiškinti, kaip veikia Gal kas norėtų dirbti su manimi kartu?
 - Techninės pagalbos inžinierius man vėl neperskambino
 - Vadovybė paprašė manęs dvi-tris dienas dirbti kitą darbą

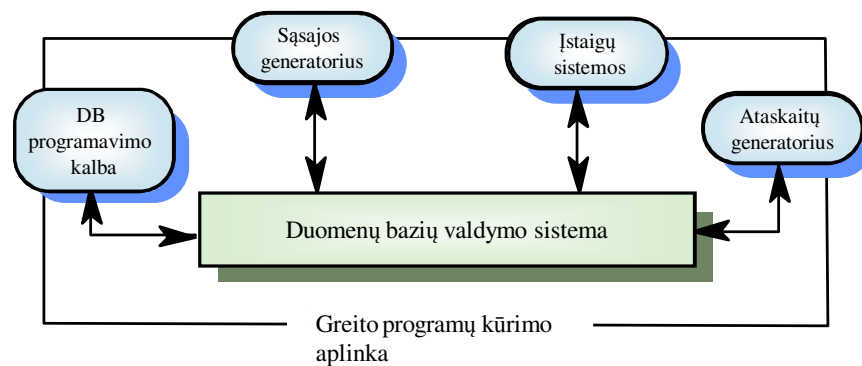
Greitas programinės įrangos kūrimas (Rapid Application Development)

- Agile metodams skiriamas didelis dėmesys, bet kiti greito kūrimo metodai buvo naudojami jau daug metų.
- Jie skirti intensyviai apdorojamų duomenų verslo sistemų kūrimui ir remiasi programavimu ir duomenų bazių informacijos atvaizdavimu.

Greito kūrimo aplinkos priemonės

- Duomenų bazių programavimo kalbos
- Sąsajų generatoriai
- Ryšiai su įstaigų programomis
- Ataskaitų generatoriai

Duomenų bazių programavimas



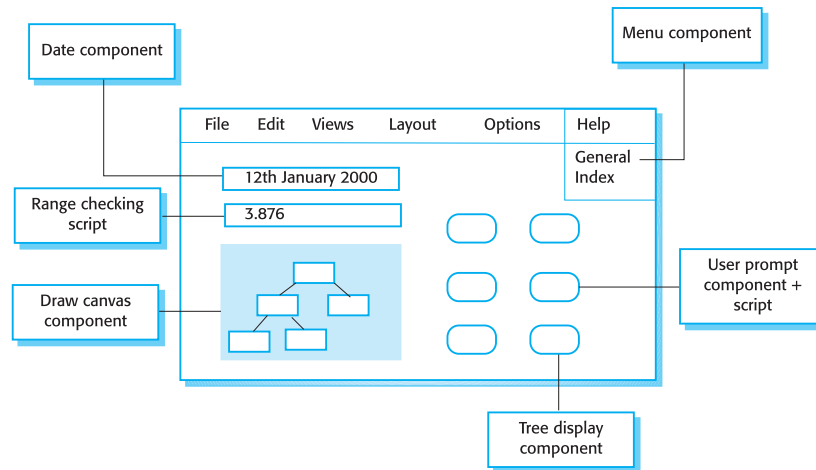
Sąsajų generavimas

- Daugelis taikymų yra pagrįsti kompleksinėmis formomis ir rankinis jų kūrimas yra varginanti veikla.
- Greito kūrimo aplinka apima ekrano vaizdo generavimą apimantį:
 - Interaktyvų formų apibrėžimą naudojant tempti ir mesti (drag and drop) metodus;
 - Formų jungimą, kur vaizduojamų formų seka yra apibrėžta;
 - Formų tikrinimą, kur formų laukų leistinos reikšmės yra apibrėžtos.

Vizualus programavimas

- Scenarijaus kalbos, tokios kaip Visual Basic, MSVC ir kt., palaiko vizualinį programavimą, kur prototipas sukuriamas kuriant vartotojo sąsają iš standartinių komponentų ir juos apjungiant.
- Didelės komponentų bibliotekos egzistuoja tam, kad palaikytų šį kūrimo tipą.
- Jos gali būti priderintos, kad atitiktų specifinius programos reikalavimus.

Vizualus programavimas su pakartotiniu panaudojimu



©Ian Sommerville 2007

Software Engineering, 8th edition, Chapter 17

Slide 45

Problemos dėl vizualaus programinės įrangos kūrimo

- Sunku koordinuoti komandinį (Team-based) kūrimą.
- Nėra aiškios sistemos architektūros.
- Sudėtingi ryšiai tarp programos dalių gali sukelti palaikymo problemas.

©Ian Sommerville 2007

Software Engineering, 8th edition, Chapter 17

Slide 46

Temos

- Lankstūs metodai
- Extremalus programavimas
- Greitas taikymų kūrimas
- Prototipų naudojimas

Sistemos prototipų naudojimo sritys

- Dažniausiai prototipai naudojami tam, kad klientas ir sistemos kūrėjas geriau suprastų sistemos reikalavimus.
 - ✓ Reikalavimų išgavimas. Vartotojas gali eksperimentuoti su prototipu, kad pamatytų kaip sistema atlieka reikalingą darbą.
 - ✓ Reikalavimų atestavimas. Prototipai gali aptikti klaidas ir praleistus reikalavimus.
- Prototipų kūrimas gali būti laikomas kaip riziką mažinanti veikla, kuri mažina su reikalavimais susijusias rizikas.

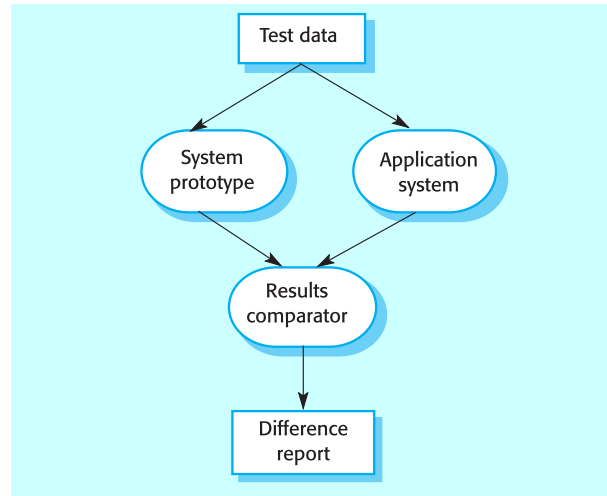
Prototipų naudojimo įtaka sistemai

- Pagerintas sistemos naudojamumas
- Sukuriama artimesnė vartotojo poreikiams sistema.
- Pagerinta projekto kokybė.
- Palengvintas palaikymo procesas.
- Sumažintos vystymo pastangos.

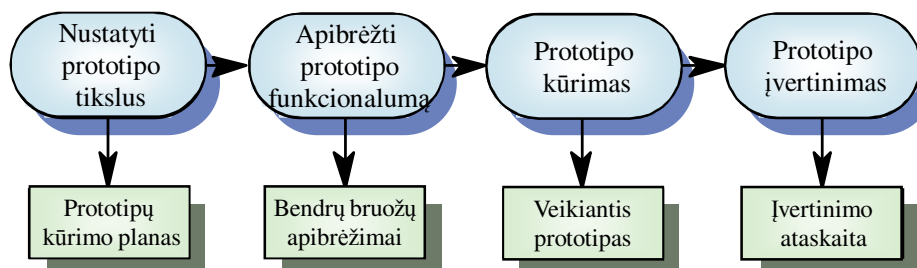
Prototipų kūrimo nauda

- Nesusipratimų tarp kliento ir programinės įrangos kūrėjo išaiškinimas.
- Trūkstamos ir neaiškios paslaugos gali būti aptiktos.
- Veikianti sistema yra prieinama jau kūrimo proceso pradžioje.
- Prototipai gali būti kaip pagrindas sistemos specifikacijų išgavimui.
- Gali palaikyti vartotojų apmokymą ir sistemos testavimą.

Prototipo naudojimas testavime



Prototipo kūrimo procesas



Išmetami prototipai

- Naudojamas sumažinti sistemos reikalavimų riziką.
- Prototipas pradamas kurti nuo pradinės specifikacijos, pateikiamas eksperimentavimui ir po to išmetamas.
- Išmetimo prototipas **negali** būti laikomas galutine sistema
 - ✓ kai kurios sistemos charakteristikos gali būti praleistos
 - ✓ nėra specifikacijų ilgalaikiam sistemos palaikymui
 - ✓ sistema bus prastos struktūros ir sunkiai palaikoma

Esminiai aspektai

- Iteratyvus programų kūrimo metodas leidžia greičiau pateikti programas.
- Agile metodai yra iteratyvūs kūrimo metodai, jų tikslas sumažinti kūrimo pridėtines išlaidas ir tuo pačiu greičiau pateikiant programas.
- Ekstremalus programavimas numato sistematinį testavimą, nuolatinį tobulinimą ir vartotojų įtraukimą.
- Vykdomų testų parašymas prieš programos sudarymą yra ypač svarbus ekstremaliam programavimui.

Esminiai aspektai

- Greitos programų kūrimo aplinkos naudoja duomenų bazių programavimo kalbas, formų generavimo priemones, ir ryšius su įstaigos programomis.
- Išmetimo prototipas naudojamas reikalavimų paieškai ir projektavimo pasirinkimui.
- Realizuojant išmetimo prototipą pradedama nuo mažiausiai suprantamų reikalavimų, o palaipsninio kūrimo metu nuo geriausiai suprantamų reikalavimų.