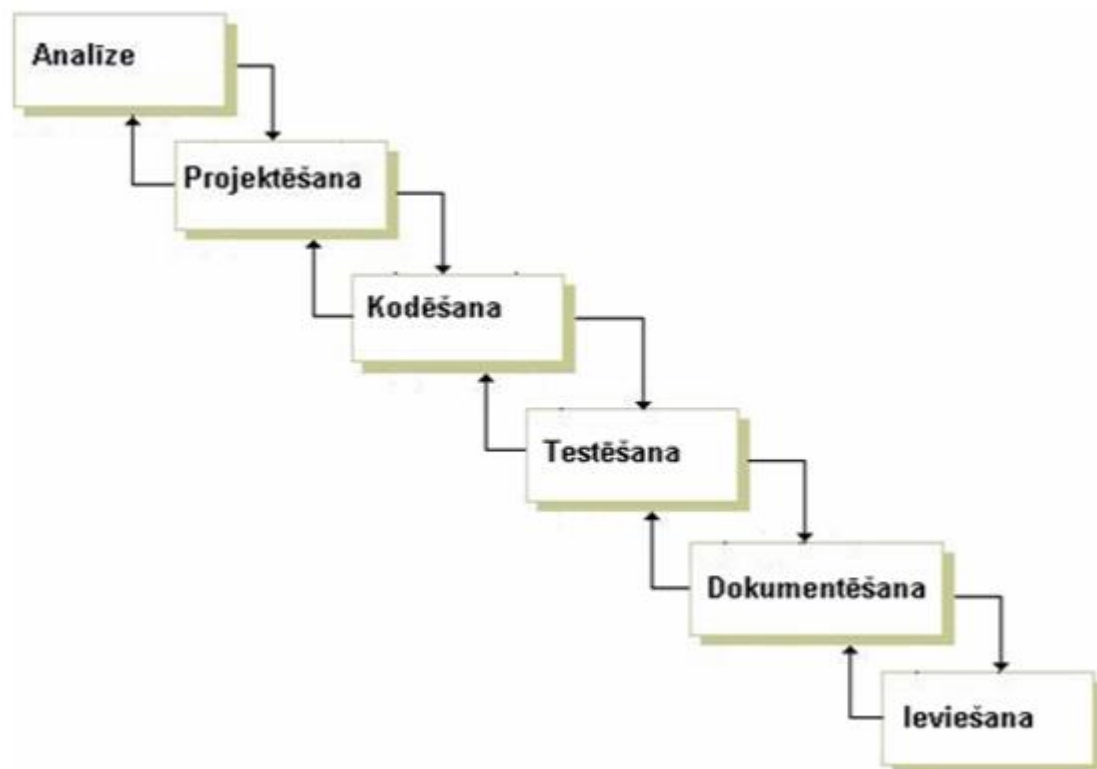
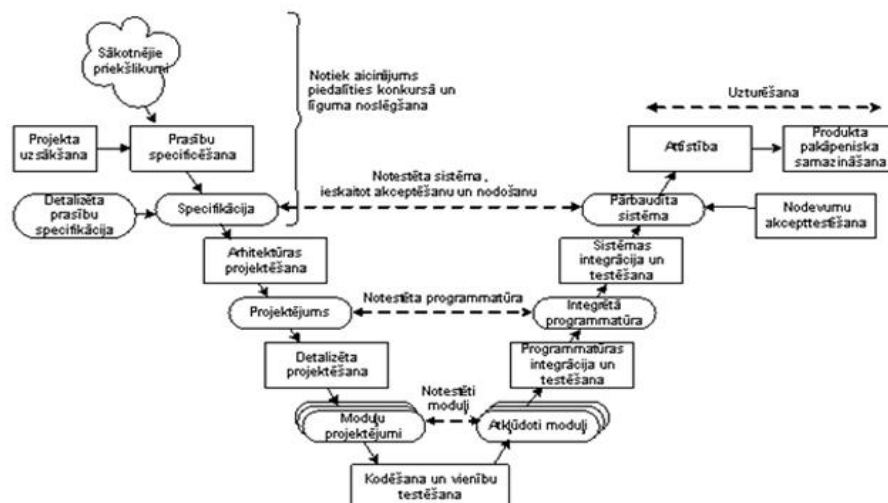


Ūdenskrituma modelis



Ūdenskrituma modelī sistēmas izstrāde ir sadalīta vairākās secīgās aktivitātēs vai fāzēs, kuras katras aktivitātes rezultāts tiek izmantots par ieejas informāciju nākamajai aktivitātei. Mūsdienās par ūdenskrituma modeli sauc jebkuru izstrādes metodoloģiju ar secīgām aktivitātēm neatkarībā no fāžu skaita un specifikas. Šim modelim ir virkne pozitīvu īpašību. Katras aktivitātes beigās tiek pielietota saražoto produktu verifikācija un validācija, kas palīdz sasniegt zināmu kvalitāti. Gadījumos, kad pasūtītāja prasības ir labi zināmas un saprotamas, ūdenskrituma modelis ir īstajā vietā.

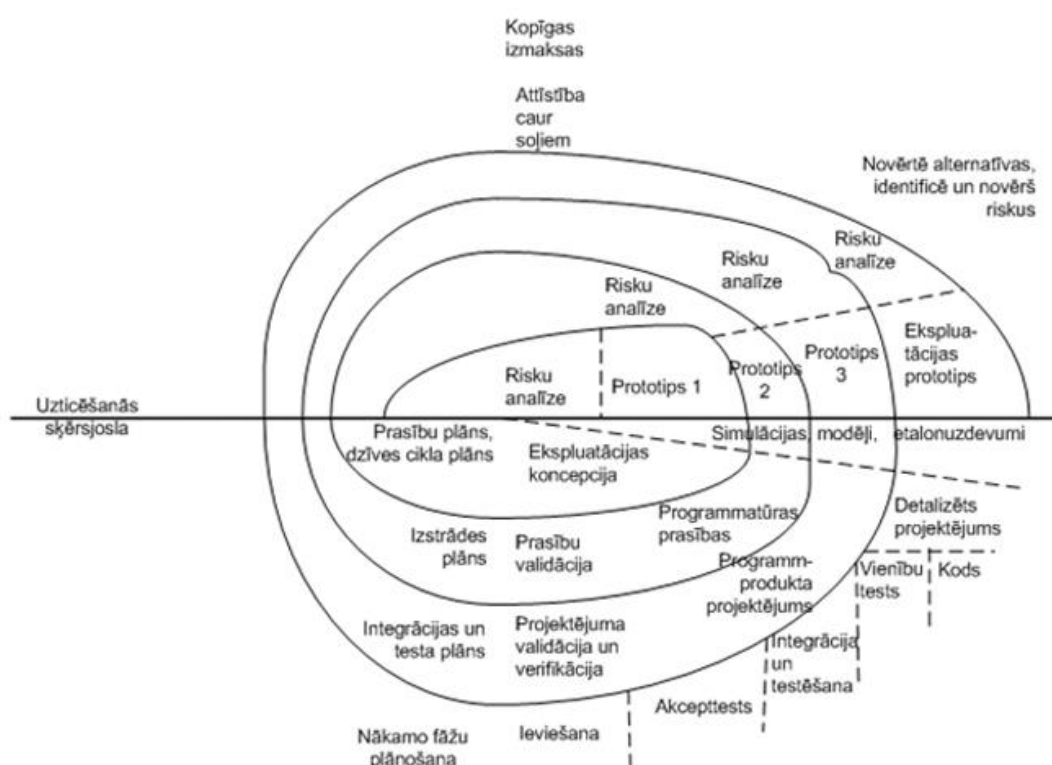
“V”modelis



Šajā diagrammā kreisais V zars parāda progresu sākot ar analīzi un projektēšanu līdz kodēšanai un pieaugošam sistēmas sadalījumam komponentēs. Labais V zars attēlo progresējošu sastāvdaļu salikšanu un testēšanu, beidzoties ar programmatūras produkta piegādi. Šī modeļa svarīgā īpašība ir dažādu fāžu savstarpējas atbilstības parādīšana. Piemēram, patstāvīgas programmas vai moduļi tiek testēti pret patstāvīgo moduļu projektējumiem, integrētā sistēmai tiek veikta sistēm-testēšana pret sistēmas projektējumu un visbeidzot gala sistēmai gala lietotāji veic akcepttestēšanu pret sākotnējām prasībām, noformētām prasību specifikācijā. Šajā modelī ar šīs atbilstības palīdzību, parādās arī kvalitātes nodrošinošie elementi. 'V' modelis ir interesants projekta pārvaldniekam arī cita iemesla dēļ. Gadījumos, kad projektā tiek pieaicināti ārējie apakškontraktori, šis modelis nodrošina skaidru izstrādes darba uzdevumu definēšanu un nodevumu pārbaudi.

Spirāles modelis

Spirāles modelis atšķiras no ūdenskrituma modeļa, piedāvājot evolūcijas vai iteratīvu pieeju sistēmas izstrādei. Ūdenskrituma modelis fokusējas uz fāze-pēc-fāzes procesu ar gala produktu no iepriekšējās fāzes, nonākot nākamajā. Šis piegājiens strādā labi gadījumos, kad prasības ir labi saprotamas un definētas, kā arī izstrādes vide ir pietiekami stabila. Ļoti bieži gadās situācijas, kad prasības nav zināmas gala lietotājiem, nav skaidri definējamas vai nav skaidrības par sistēmas pielietojumu praksē. Šajās situācijās var pielietot evolūcijas pieeju. Evolūcijas modelis paredz vairāku soļu ciklisku atkārtotāšanu, sniedzot arvien detalizētākas un skaidrākas prasības, kā arī atkārtotot dzīves ciklu vairākas reizes.



Projekts sākas ar spirāles centru un attīstās uz ārpusi. Atrdoties centrā, prasības ir grūti saprotams, bet tiek arvien vairāk papildinātas, virzoties uz priekšu pa spirāli. Kopējās

projekta izmaksas pieaug ar katru nākamo soli pa spirāli. Modelis ir sadalīts četros kvadrātos. Kreisajā augšējā kvadrātā tiek noteikti mērķi un identificētas alternatīvas un ierobežojumi. Augšējā labajā kvadrātā alternatīvas ir novērtētas un riski identificēti un izskatīti. Apakšējā labajā kvadrātā notiek izstrādes uzdevumu risināšana. Apakšējā kreisajā kvadrātā tiek plānota nākamā fāze vai iterācija. Bohema spirāles modelis piedāvā svarīgu koncepciju, kurā mērķu noteikšana, risku pārvaldība un plānošana ir ieļauti vispārējā ciklā.

Ar ko atšķiras Agile un ūdenskrituma metodes

Agile – spēja, jeb iteratīva izstrāde. Radies no vārda “agile”, jeb dzīvīgs. Agile uzskata, ka IT sistēma ir kā organisms, kas nevar tikt saprasts pirms nav uztaisīts. Līdz ar to sistēma tiek taisīta pa daļām – “saprātu/uztaisīju”. Komandā ir iekļauts pasūtītājs, un tai tiek piešķirtas netipiski lielas privilēģijas izlemt kāda sistēma izskatīsies.

Ūdenskritums – secīga izstrāde. Radies no projekta diagrammas, kurā darbi parasti parādās secībā kā kāpnes (prasības =>specifikācija=>kods=>testēšana utt.). Izpilde atgādina iešanu uz leju kā pa ūdenskritumu ar daudziem posmiem – no šejienes arī nosaukums. Ūdenskrituma metode uzskata, ka IT sistēmu ir jāizstrādā, to arvien detalizētāk specificējot, pirms tam katru detalizācijas līmeni, saskaņojot ar Pasūtītāju.

Galvenās atšķirības:

Īpašība	Agile	Ūdenskritums
Fokuss:	Izpildes termiņš	Piegādājama apjoms
Lēmumu pieņemšana	Komanda	Projekta vadītājs
Pasūtītāja loma	Piedalās detalizēti apjoma definēšanā visu laiku.	Katrā posmā akceptē arvien detalizētāku visas sistēmas aprakstu.
Piegādes laiks	Īsi posmi (1-4 nedēļas).	Garš izstrādes termiņš (tipiski 6-12 mēneši).
Kontrole par gala rezultātu	Pēc katra posma ir “gatavs un instalējams” sistēmas gabals.	Tipiski īsi pirms projekta beigām.
Galvenie riski	Var neievērot atkarības vai apstākļus, kuru dēļ sistēma var nonākt strupceļā, kā dēļ sistēma ir jāpārtaisa, radot papildus izmaksas.	Var nepamanīt būtiskas detaļas, kuru dēļ sistēmas apjoms var izrādīties būtiski lielāks nekā sākumā likās, tādējādi radot papildus izmaksas.
Ieguvumi	Ātri un visu laiku redzams kurp virzās projekts.	Savlaicīgi sadala visu apjomu un visus resursus uz visu projekta izpildes laiku.