**Modelet e procesit për zhvillimin e softuerit**

**Projekti** është grup i veprimeve të planifikuara dhe realizuara nga një grup i njerzëve për të arritur një qëllim të caktuar duke u kufizuar në kohë dhe në kosto.

Në menaxhimin e një projekti përfshihen : Përcaktimi i metodologjis, menagjimi i kufizimeve teknike, menagjimi i individëve të përfshirë dhe menxhimi i mjeteve të disponueshme.

**Metodologjia e zhvillimit softuerik** është kornizë që përdoret për të planifikuar, strukturuar dhe kontrolluar procesin e zhvillimit të sistemit softuerik.Ndryshe njihet si **Cikli i jetës për zhvillim të softuerik** dhe **Procesi i zhvillimit të softuerit.**

Procesi i zhvillimit të softuerit përfshin këto **aktivitete** të cilat kanë qëllim zhvillimin ose përmisimin e softuerit : *Specifikimi, Dizajnimi, Zhvillimi, Testimi, Evaluimi.*

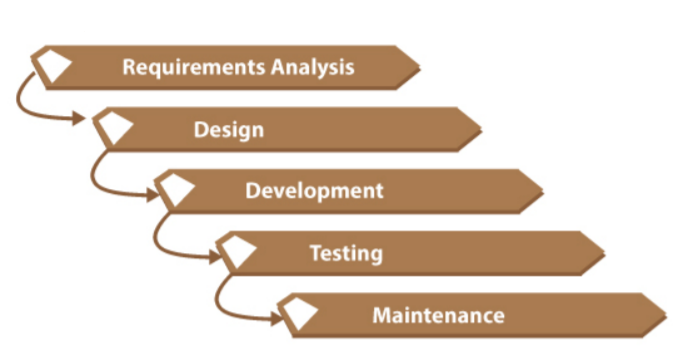
**Procesi i zhvillimit të softuerit** është grup i aktiviteteve të ndërlidhuar për të ndërtuar një softuer.

**Përshkrimi i procesit** është pjesë e procesit të zhvillimit të softuerit dhe përfshinë : **Produktet** – rezultatet pas secilit aktivetet. **Rolet** – rolet e indivdëve në proces. **Kushtet para dhe pas** – secilit aktivitet.

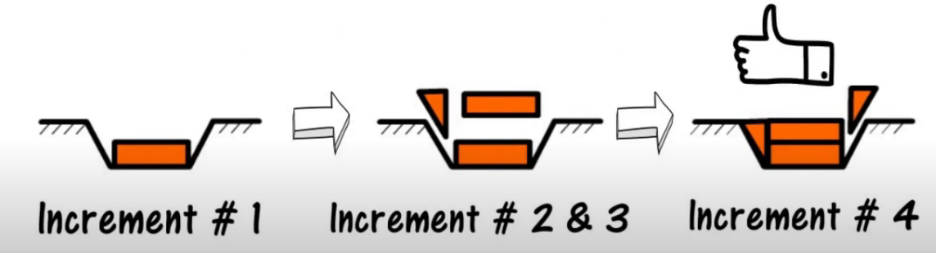
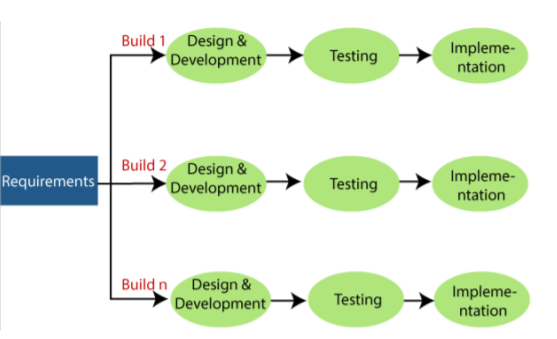
Cikli i jetesës së softuerit përfshinë të gjitha fazat që nga specifikimi i kërkesave deri në krijimin e softuerit. Kto faza organizohen në bazë të modelit që drejton.

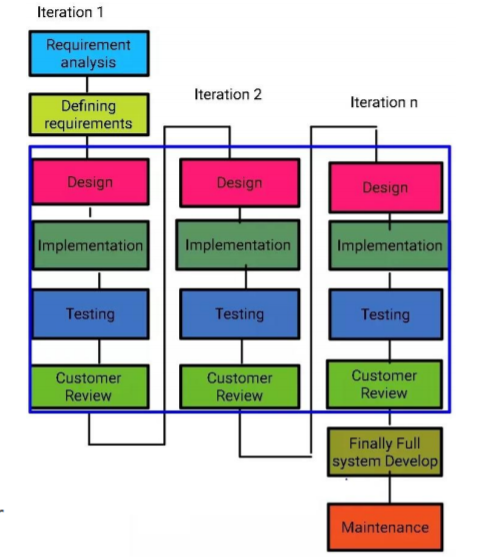
Disa nga modelet e zhvillimit të softuerit janë : **Linear, Iterativ, Rritës, Adaptues dhe Ekstrem**.

**Modeli Linear** : Fazat janë të varura nga njëra tjetra dhe ekzekutohen në mënyrë sekuenciale, nuk ofrohet mundësia për feedback, qëllimet, kërkesat dhe zgjidhjet janë mirë të definuara. Lejon shumë pak ndryshime në fushveprimin e produktit. Produkti ofrohet vetëm pas përfundimit të gjitha fazave. Ndryshe njihet si **waterfall**, secila fazë fillon vetëm pasi të ket përfunduar faza paraprake, nuk ka overlap të fazave. Punon vetëm nëse kërkesat janë të qarta qysh në fillim, nëse nuk janë ne nuk mund të vazhdojmë me asnjë fazë tjetër.

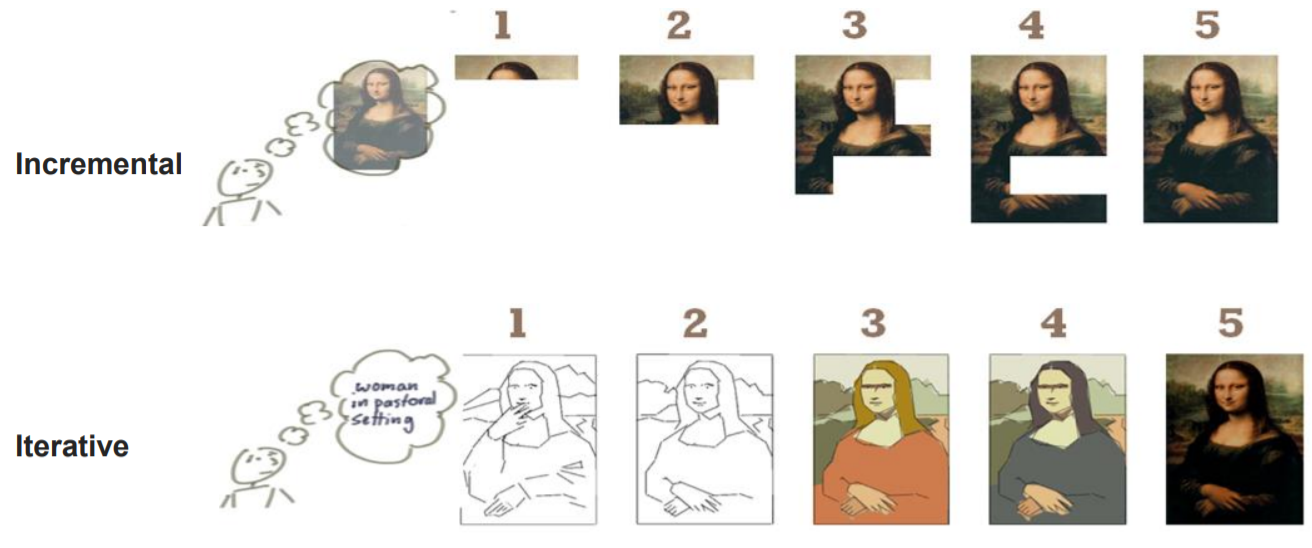
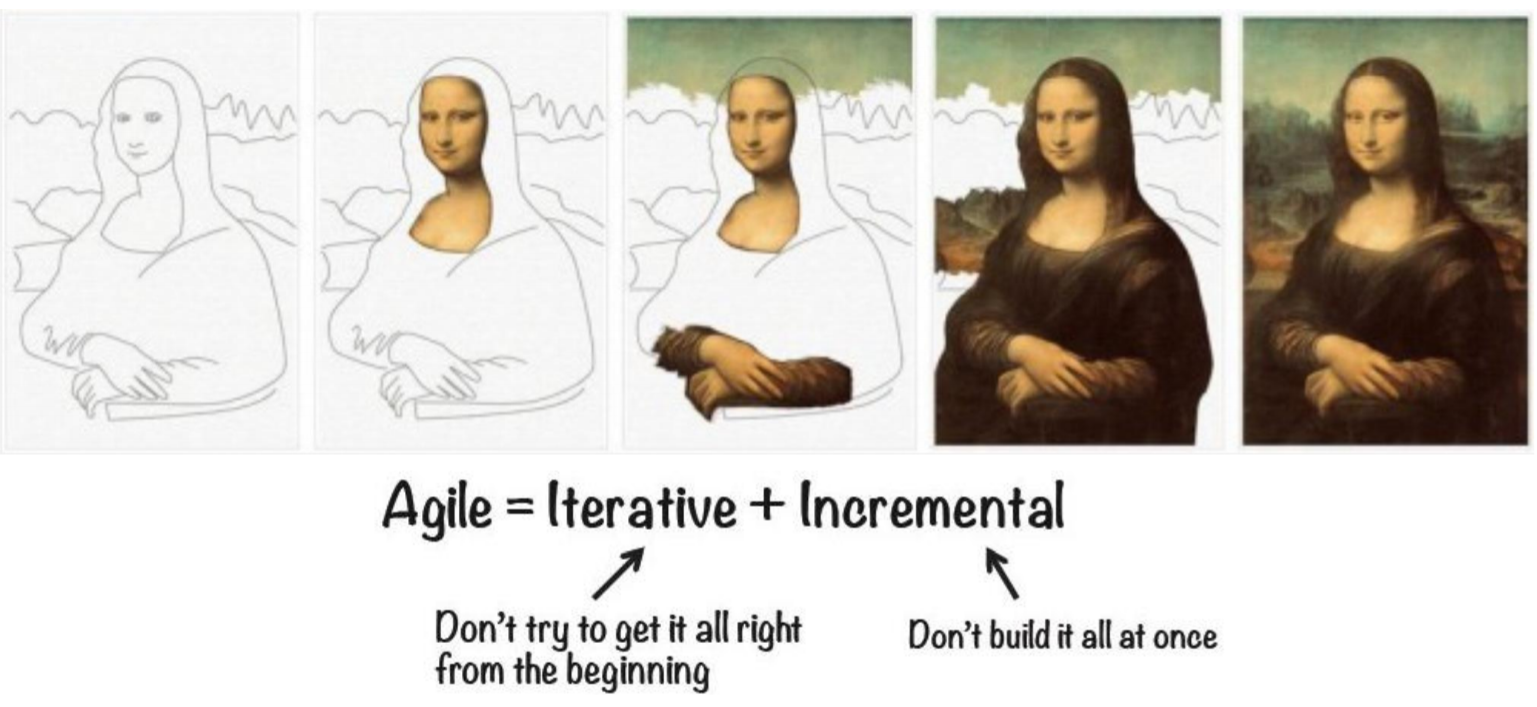


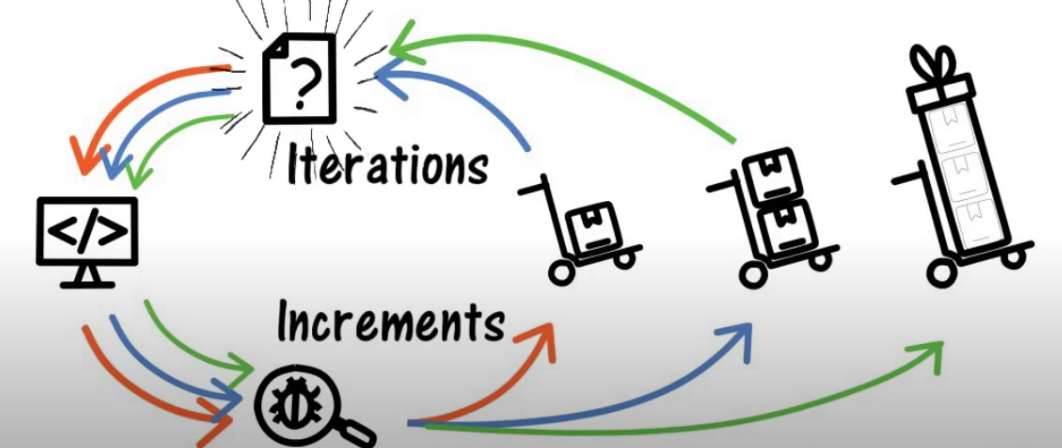
**Modeli Rritës** : Ngjason me atë linear, por për dallimi zhvillimi i softuerit bëhet me *inkremente*, pas gjdo *inkrementi* lancohet një zgjidhje e pjesëshme ashtu që të mirret feedback nga klientët se qfarë duhet shtuar(qfarë feature). Produkti final dorëzohet në formë të komponentave pasi të përfundojn të gjitha *inkrementet*. Ky model është i mirë kur kërkesat nuk janë mir të definuara dhe kemi të bëjm me projekte jo të vogla.



**Modeli Përsëritës** : Softueri zhvillohet në *iteracioni* ose përsëritje të plotë të fazave, pas gjdo *iteracioni* marin feedback nga klientët dhe në *iteracionin* tjetër bëjmë ndryshimet e nevojshme për plotësimin e kërkesave të marra nga feedback, përdor strategjin *learn-by-doing*. Ky model preferohet për projektet të cilat do pësojnë ndryshime shumë. 

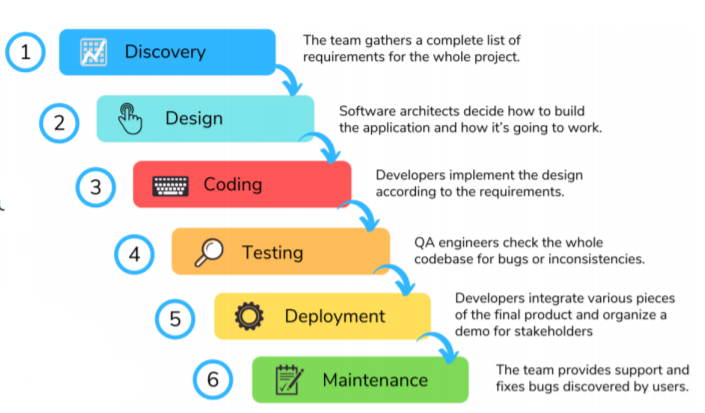
**Feedback** përdoret për të rishikuar prioritet e projektit dhe për të bër ndryshime në funksionalitet, kërkesa ose planifikime.

  
Kombinimi i modelit *inkremental* dhe ati *iterativ* na rezulton në një model të ri të quajtur *Agile*. 

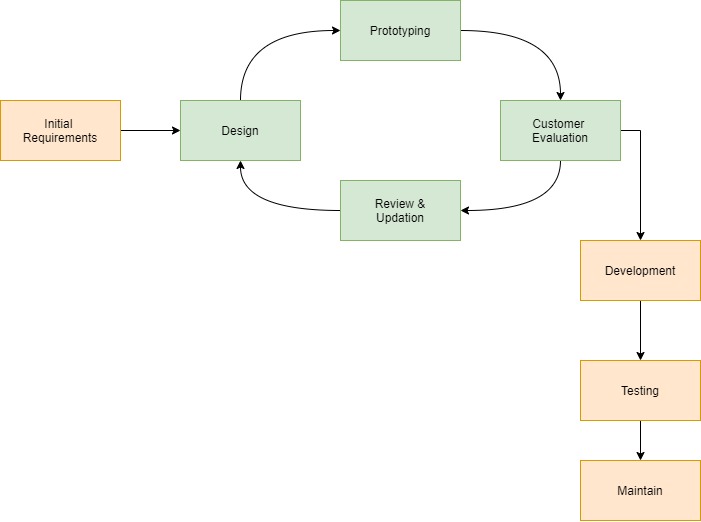


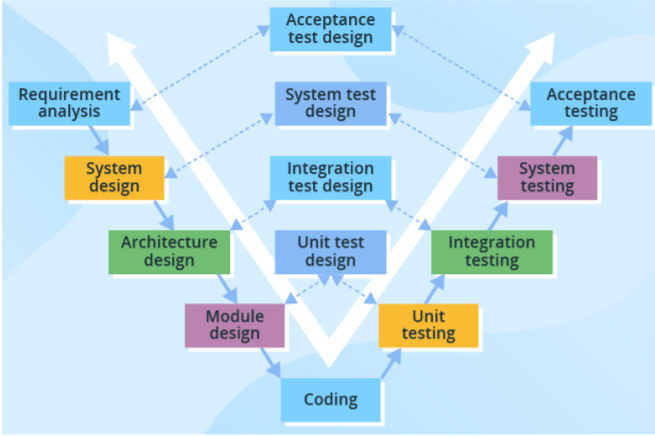
**Metodologjitë e zhvillimit të sofuterit janë** : Ad-hoc, Waterfall, Prototip, V-Model, Sprial, RAD, UP, Agile.

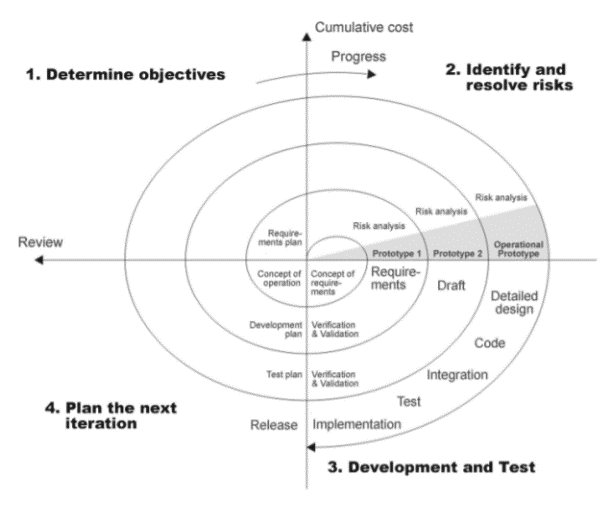
**Ad-hoc** : Nuk konsiderohet taman metodologji, klienti e shpreh kërkesën dhe zhvilluesi e bën punën. Prefrohet për projekte të vogla dhe të thjeshta, është e shpejtë dhe e drejpërdrejt. Mangësi e kti modeli është paraqitja e problemeve në rast se klienti nuk i shpreh drejt kërkesat zhvilluesi nuk do të di si të procedoj me implementimin dhe nuk lejon ndryshime në kërkesa.

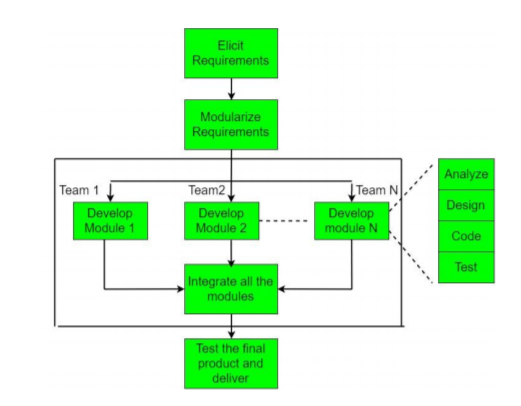
**Waterfall** : Testimet kryhen vetëm pas kodimit. Përparsi të ksaj metodologjie janë : Kërkesat të qarta para se të filloj zhvillimi, fazat kryhen në përiudha specifike kohore, secila faz rezulton me dokumentimin e duhur, është i lehtë për zbati. Mangësi të ksaj metodologjie janë : Shfaqja e problemeve gjatë kalimit nga një faz në fazën tjetër, produkti ofrohen vetëm pas përfundimit të të gjitha fazave, nuk lejon ndryshimin në kërkesa dhe nuk është fleksibil.

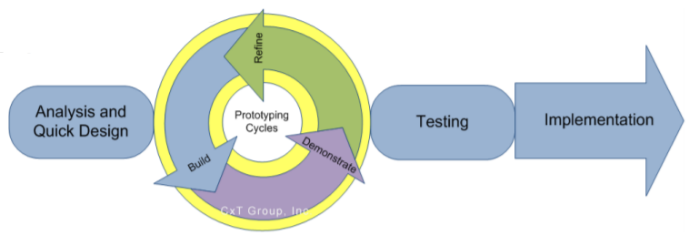
*Prototipi shfaqë funksionet kryesore të produktit final por nuk dmth që ka logjikën e njëjtë si të produktit final ose është vetë produkti final.*

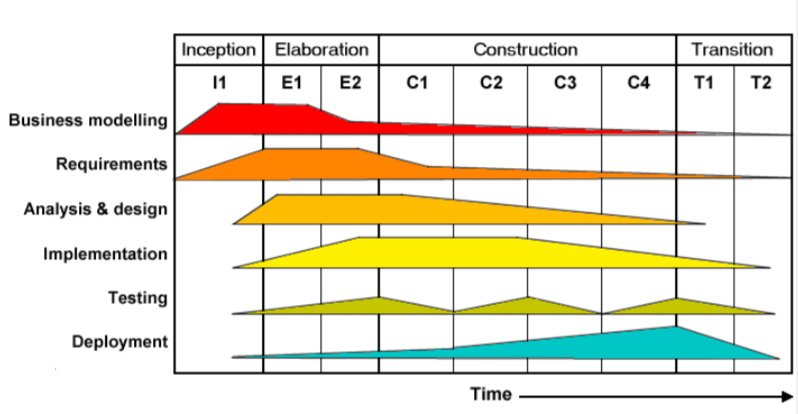
**Modeli i Prototipit** : Së pari mirren kërkesat për produktin që do të krijohet, pasi ti kemi siguruar kërkesat kalojmë në një fazë *iteruese* ku përfshihet *dizajni, prototipimi* dhe pastaj *evaluimi nga konsumatori*(vlerësimi) në këtë fazë konsumatori e bën vlersimin e prototipit nëse e vlerson negativisht ose dëshiron që të shtojë, përmisojë ndonjë kërkesë ne vazhdojmë me fazën e *rafinimit të kërkesave* duke u bazuar në sygjerimet e mara nga konsumatori. Ky itermin ndodhë deri sa konsumatori të jet i knaq me prototipin me qrastë kalohet në fazën e *implementimit*, *testimit, mirëmbajties*. *Përparësitë* e ktij modeli janë : Konsumatori është i përfshirë shumë në fazat para zhvillimit me qrastë ai e ka të qartë se qfarë do të ofrojë produkti final, na jep feedback i cili na ndihmon që të identifiokojmë herët funksionet e vështira dhe konfuze ose atë që mungojnë, gjithashtu ulet kostoja dhe humbja e kohës ngase defektet identifikohen herët. Ky model është shumë i përshtashëm nëse ka ndryshime të shumta në kërkesa. *Mangësit* e ktij modeli janë : Numri i iteracioneve mund të jetë shumë i madhë dhe ne nuk mund ta dim sa iteracione do kryhen ose sa kohë do të konsumojnë sepse kjo varet nga konsumatori. Projekti mund të rritet në komplekistet pasi që lejon që qëllimet/kërkesat të ndryshojmë, investimi në prototipe mund të jet i madhë nëse nuk kontrollohet si duhet.

**Modeli-V** : Është verison i modifikuar i waterfall, secila fazë është e pasqyruar, dhe për secilën fazë ka test raste të veqanta. Zhvilluesit punojnë në mënyr paralele me testuesit. Përparësi e ksaj metodologjie është : testimet e shumta. Mangësitë e ksaj metodologjie janë : ngurtësia, ndryshimet në ndonjë faz relfektojnë në ndryshime të dokumentacioneve të kërkesave dhe testimeve, nuk preferohet për projekte afat shkurte.

**Spirale** : Përfshin në vete modelet *waterfall*, *iterativ* dhe *prototip*, ndahet në katër faza kryesore : **Përcaktimi i objektivave, Analizimi i rrezikut, Inxhinieria**(zhvillimi dhe testimi)**, Planifikimi**. Kjo metodologji preferohet për projekte të mëdha, të ndërlikuara, me rrezik, kur nevojiten prototipet, kur priten ndryshime. Përparsi të ksaj metodologjie janë : Analiz e madhe e rrezikut, feedback i shpejtë, ofron zgjidhje para përfundimit. Mangësi të kësaj metodologjie janë : Kosto e lartë, nuk bën për projekte të vogla.

**RAD** : Zhvillimi i shpejt i aplikacionit në vete përfshinë modelet iterative, rritëse dhe adaptive, Puna mund të ndahet në module të cilat mund të ju caktohen në mënyr të pa varur ekipeve të ndryshme. Bëhen shumë pak planfikime dhe protipe të shpejta, kërkesat mblidhen nga fokus grupet, nga testimi i prototipeve, ripërdorimi i prototipeve dhe integrimet e vazhdueshme. RAD preferohet të përdoret kur : dihen kërkesat, projekti duhet të përfundoj shpejt(2-3muaj), në sistemet modulare, ska rrezik teknik, ka buxhet të mjaftushëm dhe klienti dëshiron të përfshihet në secilën faz të zhvillimit të softuerit.



**UP**(Procesi i unifikuar) : Ështe kombinim i iteratives me inrekemntales e cila është e orientuar dhe drejtuar ne Use Case dhe ne ariktekturen e sistemit. Ndahen ne 4 faza prej te cilave 3 jane itreative.

*Faza 1* : Inception - Gjate kesaj faze qellimi eshte te kuptohen kerkesat e perdoruesit per sistemin qe do te ndertohet, identifikohen funksionet kryesore dhe modelohen 20% te Use Case. Vetem nje Iterim. *Faza 2* : Elaboration - Kjo faze eshte e fokusuar ne krijim e nje arkitekture te ekzekutueshme.te sistemit, gjithashtu gjate kesaj faze kuptohen edhe kerkesat tjera te sistemit dhe modelohen 80% i Use case, gjithashtu percaktohen kerkesat jo - funksionale. 2 Iterime. *Faza 3* : Construction - Faza qe merr me se shumti kohe realizohet 4 Iterime. Gjate ksaj faze behet implementimi/kodimi i sistemit dhe testet. *Faza 4* : Tansition - Faza e fundit me dy iterime, ne kete faze behet lansimi i sistemit dhe behen trajnime per perdoruesit dhe mirembajtesit.

**Agile** : Përfshin në vete modelin iterativ dhe inkremental. Në agile kemi ekip vetë-organizuese, dorëzim gradual, zhvillim evulocionar, përqëndrim në rezultate, të orientuar në njerëz feedback të shpejtë dhe fleksibilitet ndaj ndryshimeve. Agile është më shumë kornizë konceptuale sesa një metodologji. Agile thekson 4 vlera dhe 12 parime për zhvillim të softuerit.

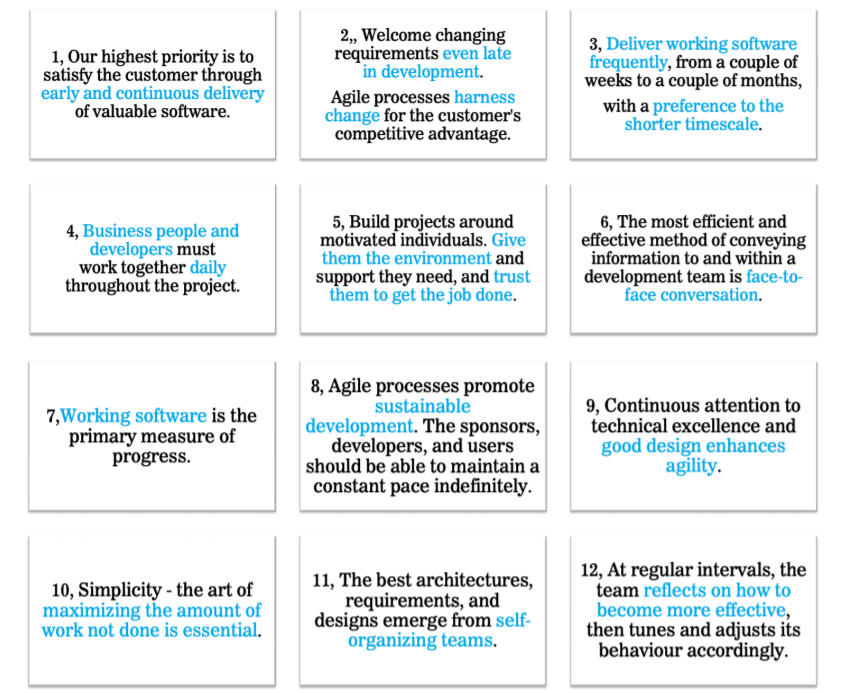
Vlerat e theksuara nga Agile janë : **Individuals and Interaction** over process and tools, **Working Software** over comprehensive documentation, **Costumer Collabration** over contract negotation, **Response to change** over following a plan.

**Individuals and Interaction** : Agile thekson se zgjedhia e indivdëve të duhur për të zhvilluar softuerin është esenciale për suksesin e projektit, gjithashtu ndërveprimi mes individëve ndihmonë në tejkalimin e problemeve që mund të shfaqen.

**Working Software** : Agile bën të qartë se në fokus të jet zhvillimi i softuerit që mund të përdoret nga klientët e jo krijimi i dokumentimeve.

**Costumer Collabration** : Agile thotë se klienti duhet të futet në një feedback loop deri në përfundim të projektit, në mënyr të produkti ti përmbush të gjitha kërkesat e klientit e jo të bëhet një gjë e tillë me kontrat e cila mund të rezultoj në mospërputhje të kërkesave.

**Response to change** : Sipas Agile nuk duhet të ndjekim një plan statik nga fillimi deri në fund, në të kundërtën duhet të ndërtojmë një plan dinamik i cili i mundëson klientit të bëjn ndryshime në fushveprim të projektit.

**Benifitet** e metodologjisë Agile : Fokusim në përdorues, fokusim në vlerat e biznesit, përmisim i cilësisë,dorzim i hershëm,transparencë, parashikim i kostos dhe orarit, lejon ndryshime, inkuardon palët e interesit.