Mis on arendusprotsess?

Tarkvaraarendusprotsess on süstemaatiline lähenemisviis, mille abil loodakse, kohandatakse või parandatakse tarkvara. See protsess hõlmab mitmeid etappe ja tegevusi, mille eesmärk on tagada, et lõpptoode vastaks kindlaksmääratud nõuetele ja oleks kvaliteetne.

Tüüpiline tarkvaraarendusprotsess sisaldab järgmisi etappe:

1. Nõudemääratlus ja planeerimine:
   * Määratakse kindlaks tarkvaraprojekti eesmärgid, vajadused ja piirangud.
   * Koostatakse projekti plaan, mis sisaldab ajakava, ressursse, kulusid ja muid olulisi aspekte.
2. Süsteemidisain:
   * Arendatakse üksikasjalik süsteemiplaan, mis kirjeldab tarkvarasüsteemi arhitektuuri, komponente ja nende omavahelisi suhteid.
   * Määratakse kindlaks tarkvara moodulid ja nende funktsionaalsus.
3. Rakendamine (Programmeerimine):
   * Koodi kirjutamine vastavalt süsteemidisainile ja nõuetele.
   * Testitakse ja parandatakse koodi vastavalt vajadusele.
4. Testimine:
   * Hinnatakse tarkvara, et tagada selle vastavus nõuetele ja leida võimalikke vigu.
   * Testimine võib hõlmata erinevaid tasandeid, sealhulgas üksikmooduli testimine, integreerimistestimine ja süsteemitestimine.
5. Hooldus ja värskendamine:
   * Pärast tarkvara väljalaset võib olla vajalik hooldus ja värskendamine, et parandada vigu, lisada uusi funktsioone või kohanduda keskkonna muutustega.
   * Hooldus hõlmab tarkvara toimimise jälgimist ja vajalike muudatuste tegemist.

**Süsteemiarenduse protsess** hõlmab tervet rida tegevusi, mille eesmärk on luua või täiendada infosüsteemi vastavalt konkreetsetele nõuetele ja eesmärkidele. See on organiseeritud ja struktureeritud meetod, mis tagab süsteemi tõhusa jaotamise, kasutamise ja hooldamise. Siin on üldine süsteemiarenduse protsess ja mõned meetodid, mida selle raames kasutatakse:

### **Süsteemiarenduse Protsess:**

* Vajaduste Analüüs:
  + Määratleb süsteemi eesmärgid ja eesmärgid.
  + Kogub ja analüüsib kasutaja nõudeid.
* Süsteemi Planeerimine:
  + Määratleb projekti ulatust ja eesmärgid.
  + Planeerib ressursid, ajakava ja eelarve.
* Süsteemi Kavandamine:
  + Loob süsteemi arhitektuuri ja disaini.
  + Valib tehnoloogiad ja vahendid.
* Implementatsioon:
  + Koodi kirjutamine ja süsteemi ehitamine.
  + Rakenduste testimine.
* Integratsioon ja süsteemi testimine:
  + Integreerib kõik osad.
  + Testib süsteemi tervikuna.
* Süsteemi paigutus ja kasutuselevõtt:
  + Installeerib süsteem lõppkasutajatele.
  + Koolita kasutajaid ja käivita süsteem.
* Hooldus ja optimeerimine:
  + Jälgib süsteemi jõudlust ja käideldavust.
  + Teeb vajalikke parandusi ja täiustusi.

### **Süsteemiarenduse meetodid:**

**1) koskmudel (waterfall model);**

**2) spiraalmudel (spiral model);**

**3) inkrementaalmudel (incremental model);**

**4) prototüüpimine (prototyping).**

**Süsteemiarenduse elutsükkel** on protsess, mida järgitakse süsteemi loomisel, täiustamisel või hooldamisel. Elutsükkel hõlmab mitmeid faase, mille eesmärk on tagada süsteemi tõhus arendamine, testimine, kasutuselevõtt, hooldus ja optimeerimine. Kuigi erinevad meetodid võivad kasutada erinevaid terminoloogiaid ja lähenemisviise, on üldiselt aktsepteeritud, et süsteemiarenduse elutsükkel hõlmab järgmisi põhifaase:

* **Analüüs:**
  + Süsteemi eesmärkide ja nõuete kindlaksmääramine.
  + Kasutajate vajaduste ja ootuste analüüs.
  + Funktsionaalsete nõuete väljaselgitus
  + Mittefunktsionaalsete nõuete väljaselgitus
* **Projekteerimine (kavandamine):**
  + Süsteemi arhitektuuri kavandamine.
  + Tehniliste aspektide, nagu riistvara ja tarkvara määratlemine.
* **Realisatsiooni (teostuse) faas**:
  + Tarkvara koodi kirjutamine vastavalt kavandatud arhitektuurile.
  + Tarkvara testimine ja vigade tuvastamine.
* **Süsteemi Testimine:**
  + Süsteemi testimine vastavalt määratletud nõuetele.
  + Veaotsing ja parandamine.
* **Paigutus ja kasutuselevõtt:**
  + Süsteemi paigutamine tegelikule keskkonnale.
  + Koolitus kasutajatele ja süsteemi kasutuselevõtt.
* **Hooldus ja optimeerimine:**
  + Süsteemi jälgimine ja hooldus.
  + Täiustuste ja täienduste rakendamine vastavalt muutuvatele nõuetele.

**Võrdlus-Eelised**

| **Inkrementaal mudel** | **Prototüüpimise mudel** |
| --- | --- |
| Vajab hoolikat planeerimist ja projektijuhtimist, et tagada iteratsioonide edukas lõpuleviimine | Prototüüpide kaudu saavad kasutajad varakult osaleda ja anda väärtuslikku tagasisidet. |
| Iga iteratsioon toodab töötava osa tarkvarast, mis võimaldab kasutajatel seda varakult testida ja kasutada | Kiire prototüüpimine võimaldab kiiresti testida ideid ja kontseptsioone. |
| Kasutajate tagasiside saab integreerida arendusprotsessi, mis võimaldab paremini vastata nende vajadustele. | Enne täisfunktsionaalse toote arendamist saab prototüüpidega tuvastada probleeme ja riskitegureid. |
| Projekt saab alata väiksema ressursikuluga ja kulud suurenevad järk-järgult vastavalt iga iteratsiooni lisanduvatele funktsioonidele. |  |

**Võrdlus-Miinused**

| **Inkrementaal mudel** | **Prototüüpimise mudel** |
| --- | --- |
| Planeerimise keerukus, mis vajab hoolikat planeerimist ja projektijuhtimist, et tagada iteratsioonide edukas lõpuleviimine. | Võib olla ajamahukas, kui prototüüpimine muutub liiga keeruliseks, võib see olla ajamahukas ja kulukas. |
| Võimalikud kvaliteediprobleemid, kui igat iteratsiooni ei testita põhjalikult, võib see põhjustada kvaliteediprobleeme. | Kvaliteediprobleemid, kui prototüüpi ei testita põhjalikult, võib see põhjustada kvaliteediprobleeme lõpptootes. |
| Suurem ressursikulu pikas perspektiivis, kuigi algne investeering võib olla madalam, võib pikaajaline ressursikulu olla suurem kui teistes arendusmudelites. | Eeldab pidevat kasutajate kaasamist, missõltub kasutajate aktiivsest kaasamisest ja nende tagasiside õigeaegsest saamisest. |

**Inkrementaalmudel (Incremental Model)** on iteratiivne tarkvaraarenduse metoodika, mis jagab projekti väiksemateks osadeks ehk inkrementideks, võimaldades iga inkrementi eraldi arendada ja kasutusele võtta. Selle mudeli eesmärk on pakkuda tarkvaraarendusele paindlikumat ja haldatavamat lähenemist, jagades projekti väiksemateks osadeks. Iga inkrement põhineb eelneval ja arendusprotsess toimub järjestikuste iteratsioonide kaudu.

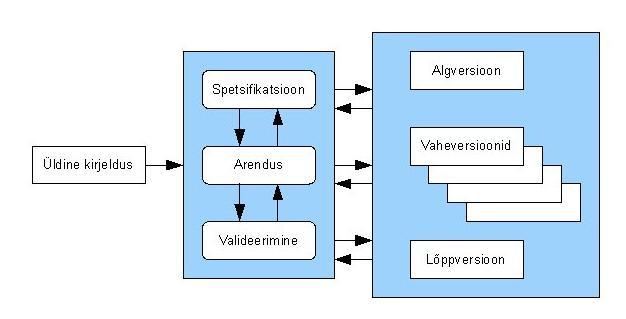
Inkrementaalmudeli peamised tunnusjooned hõlmavad järgmist:

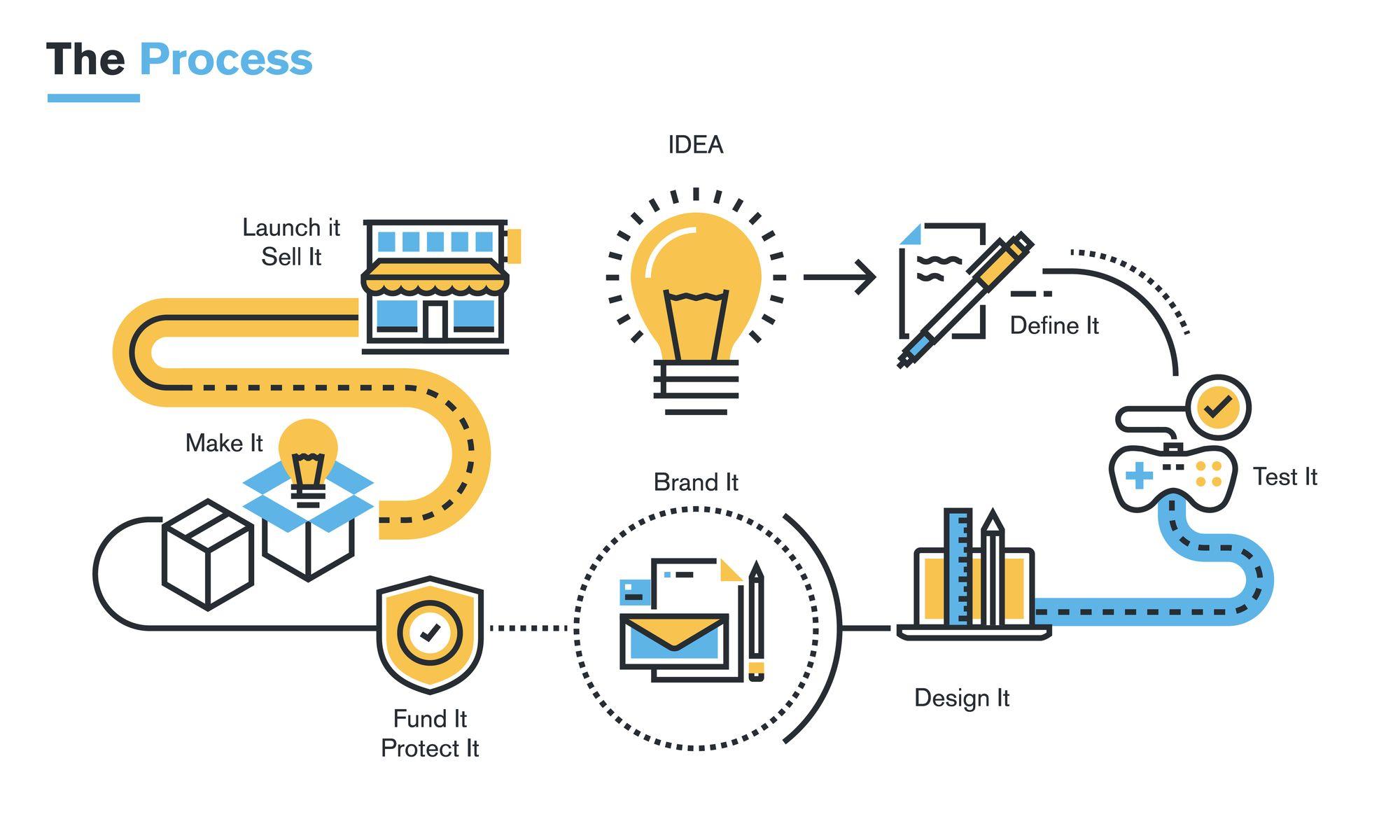
1. **Faasid:** Tarkvaraarendusprotsess jaguneb erinevateks faasideks, ja iga faas loob kasutuskõlbuliku ja potentsiaalselt tarnitava osa tarkvarast.
2. **Inkrementaalne arendus:** Arendamine toimub järk-järgult, iga inkrement lisab uusi funktsioone või parandab olemasolevaid.
3. **Paralleelne arendus:** Mõnel juhul saab erinevad inkremente arendada paralleelselt erinevad meeskonnad või arendajad, mis muudab protsessi ajaliselt efektiivsemaks.
4. **Tagasiside silmus:** Sidusrühmadel on võimalus anda tagasisidet arendusprotsessi käigus ja pärast iga inkrementi. See võimaldab kohandusi ja täiustusi kasutaja tagasiside põhjal.
5. **Integratsioon:** Uued inkremendid integreeritakse juba varem arendatud inkrementidega, tagades, et tarkvara jääb terviklikuks ja toimivaks tervikuks.
6. **Riskide haldamine:** Inkrementaalmudel aitab haldada projekti riske, pakkudes kasutuskõlblikke inkremente, mis saab varakult testida ja valideerida, vähendades suuremate probleemide tekkimise tõenäosust hilisemates arendusetappides.

**Prototüüpimine (prototyping)** on tarkvaraarenduse metoodika, mis keskendub kiirele ja iteratiivsele protsessile, et luua esialgset versiooni tarkvarast või süsteemist, mida saab kasutada kliendi või kasutajate tagasiside kogumiseks ja süsteemi disaini täiustamiseks. Prototüüpimine erineb traditsioonilistest meetoditest, nagu Waterfall, kus kogu süsteem on planeeritud ja arendatud enne lõpliku toote loomist.

Prototüüpimise protsess hõlmab järgmisi samme:

1. **Nõuete kogumine:** Alguses kogutakse kliendi või kasutaja nõuded ja eesmärgid. Kuigi need nõuded ei pea olema täielikult määratletud, on vajalik põhiline arusaam sellest, mida süsteem peaks tegema.
2. **Kiire prototüübi loomine:** Arendajad loovad kiiresti esialgse versiooni süsteemist või tarkvarast, kasutades lihtsaid tööriistu ja meetodeid. See prototüüp ei pea olema täielik ja funktsionaalne tarkvara, vaid see peab illustreerima põhilisi ideid ja funktsioone.
3. **Kasutaja tagasiside kogumine:** Prototüüpi kasutatakse kasutajate või klientidega suhtlemiseks, et saada nende tagasisidet ja arvamusi. See võimaldab tuvastada, kas prototüüp vastab nende ootustele ja vajadustele.
4. **Prototüübi täiustamine:** Kasutades saadud tagasisidet, täiustatakse ja kohandatakse prototüüpi vastavalt kasutajate soovidele ja nõuetele.
5. **Korduvad sammud:** Need sammud korduvad mitu korda, kuni prototüüp vastab lõpuks klientide ja kasutajate nõuetele ja ootustele.
6. **Täisfunktsionaalse tarkvara või süsteemi loomine:** Kui prototüüp on piisavalt täiustatud ja vastab nõuetele, võetakse prototüüp aluseks, et luua lõplik ja täisfunktsionaalne tarkvara või süsteem.

Inkrementaalmudel (Incremental Model)



Prototüüpimine (prototyping)

Kokkuvõtte:

Infotehnoloogia maailmas leidub palju ettevõtteid, kes kasutavad või kasutasid mõlemad Inkrementaalset mudelit ja prototüüpimist mudelit vastavalt oma vajadustele. Nendeks võivad olla:

Microsoft: Microsoft on kasutanud erinevaid arendusmetoodikaid, sealhulgas inkrementaalset lähenemist. Näiteks võib Microsofti tarkvaraarendusprojektidel olla pidev värskenduste ja täienduste tsükkel.

Google: Google on tuntud oma kiirete uuenduste ja uute toodete turuletoomise poolest, mis võivad nõuda inkrementaalset arendust ja prototüüpimist.

Apple: Apple on kuulus oma toodete disaini ja innovatsiooni poolest ning nende toodete arendamisel võib kasutada prototüüpimist ja iteratiivset lähenemist.

Spotify: Muusika voogedastusteenuse pakkujana on Spotify tuntud oma kiirete funktsiooniuuenduste ja kasutajakogemuse täienduste poolest, mis võivad toetuda inkrementaalsele arendusele.

Facebook: Suured sotsiaalmeedia platvormid, nagu Facebook, on kasutanud iteratiivset arendust, et kiiresti reageerida kasutajate vajadustele ja turumuutustele.

Amazon: Amazon on rakendanud paindlikke arendusmudeleid, et pidevalt parandada oma e-kaubanduse platvormi ja lisada uusi funktsioone.

On oluline mõista, et ettevõtted võivad kasutada erinevaid arendusmetoodikaid vastavalt konkreetsele projektile, meeskonnale ja ärikeskkonnale. Lisaks võivad ettevõtted kasutada mitmeid metoodikaid ja neid kohandada vastavalt oma vajadustele.

Mõlemad mudelid on suunatud muutustele ja võimaldavad paindlikku lähenemist, kuid erinevad oma fookuses ja rakendamises.

Inkrementaalmudel rõhutab järkjärgulist funktsioonide lisamist, samas kui prototüüpimismudel keskendub kiirele prototüüpide loomisele ja klientide osalusele. Valik mudeli kasutamiseks sõltub projekti nõuetest ja eesmärkidest.

Kasutatud allikad:

<https://eopearhiiv.edu.ee/e-kursused/eucip/arendus/12_ssteemiarenduse_elutskkel.html>

<https://eopearhiiv.edu.ee/e-kursused/eucip/arendus/1222_inkrementaalne_arendusmudel.html>

<https://eopearhiiv.edu.ee/e-kursused/eucip/arendus/1224_prototpimine.html>

<https://www.plutora.com/blog/incremental-model-what-and-how-to-implement-it>

<https://www.javatpoint.com/software-engineering-incremental-model>

<https://en.wikipedia.org/wiki/Incremental_build_model>

<https://en.wikipedia.org/wiki/Software_prototyping>

<https://selleo.com/blog/what-is-software-prototyping-and-why-do-you-need-it>

<https://www.tatvasoft.com/outsourcing/2022/05/what-is-software-prototyping.html>

<https://fullscale.io/blog/prototype-examples-from-top-tech-companies/>

<https://www.quora.com/Who-uses-incremental-models>

https://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/