8. OPTIMIZACIJA PROCESA

Optimizacija pomeni razumevanje trenutnega procesa in njegovo spreminjanje z namenom povečanja kvalitete produktov, zmanjšanja stroškov in časa razvoja. Za optimizacijo se priporoča uporaba zrelostnih ali agilnih pristopov (Sommerville 2010, 706).

*Agilni pristop* je fokusiran na iterativni razvoj s krčenjem presežkov v razvojnem procesu. Vendar moramo upoštevati temeljne karakteristike agilnih metod, katere so hitre izdaje funkcionalnosti in visoka odzivnost na spremembe (Sommerville 2010, 706).

*Zrelostni pristop* je fokusiran na izboljšave procesov, vodenje projektov in implementacijo boljših inženirskih praks v organizacijo. Nivo zrelosti odraža obseg adaptacije tehničnih in vodstvenih praks v proces razvoja programske opreme. Cilj pristopa je izboljšanje kvalitete produktov in predvidljivost procesa (Sommerville 2010, 706). Uporaba zrelostnih pristopov je postala osrednja aktivnost za izboljšanje procesov v organizacijah. V ta namen so se definirali določeni standardi ocenjevanja kot so: CMM[[1]](#footnote-1), CMMI[[2]](#footnote-2), BOOTSTRAP, SPICE[[3]](#footnote-3) in družina ISO-9000 standardov. Predlagani pristopi pokrivajo le procesno perspektivo razvoja. V ta namen Aleem in kolegi predlagajo DGMM[[4]](#footnote-4) z upoštevanjem faktorjev, ki temeljijo na razvojni, potrošniški in poslovni perspektivi (Aleem, Capretz, in Ahmed 2016, 57–58).

Za večje projekte in kompleksne sisteme je priporočena uporaba zrelostnih pristopov, medtem, ko je velika verjetnost, da bo izbira agilnega pristopa manjšim in srednje velikim projektom predstavljala najboljšo strategijo optimizacije procesa (Sommerville 2010, 706). Za ilustracijo bomo v našem primeru uporabili oba pristopa optimizacije procesov.

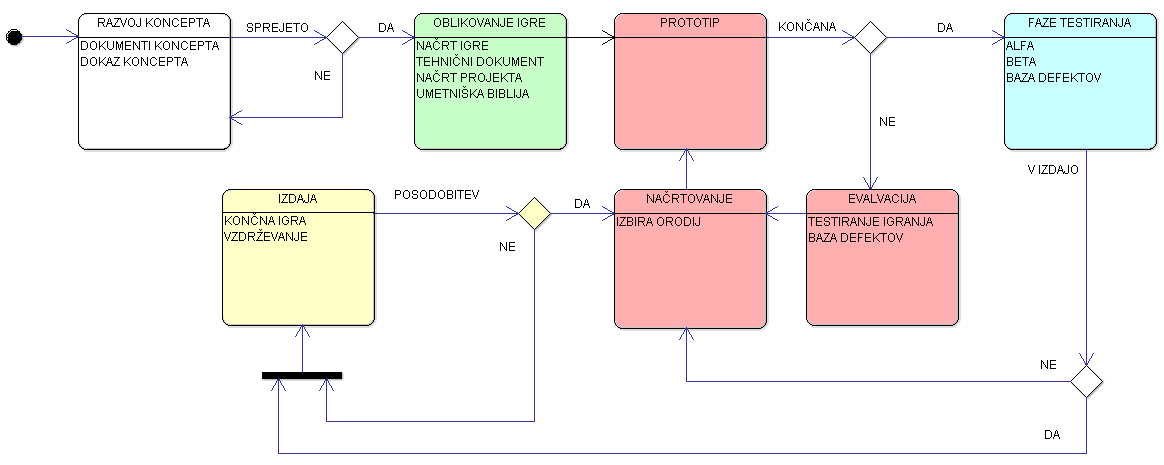
8.1 ZRELOSTNI PRISTOP

Avtorji DGMM so v prvem koraku identificirali glavne perspektive discipline, ki se delijo na: razvijalce, potrošnike in poslovni svet (Aleem, Capretz, in Ahmed 2016, 58). Nato so za posamezno perspektivo identificirali ključne faktorje, ki vplivajo na delovanje igre in razvojni proces. Izmed vseh treh perspektiv so identificirali 5 dimenzij in 18 različnih faktorjev (PRILOGA H) katere imenujejo procesne aktivnosti razvoja video iger (GDPA)[[5]](#footnote-5) (Aleem, Capretz, in Ahmed 2016, 59). Zrelost procesa razdelijo na pet stopenj (naraščajoče): ad-hoc, priložnosten, konsistenten, organiziran in optimiziran (Aleem, Capretz, in Ahmed 2016, 61). Vsako stopnjo zrelosti se preverja z namenskim vprašalnikom, katerega izjave se nanašajo na izvajanje GDPA. Izjavam podajamo odgovore ali aktivnost izvajamo ali ne. Na koncu seštejemo aktivnosti in s tem dobimo število aktivnosti[[6]](#footnote-6), ki se izvajajo. V kolikor je število izvajanih aktivnosti večje od določenega praga[[7]](#footnote-7) pomeni, da je procesni model dosegel zrelost za določeno stopnjo zrelosti. Za doseganje praga zrelosti velja, da se v procesu izvaja 80% aktivnosti glede na vse izjave, ki so podane za določeno stopnjo zrelosti (PRILOGA J).

8.2 AGILNI PRISTOP

Naš proces bomo optimizirali z obračanjem na temeljne prakse AM. Opazimo, da lahko apliciramo prakso enostavnega opisa modela. Pri modelu smo zmanjšali število korakov (balonov) in izenačili velikosti prikazov stanj. Poenostavili smo tranzicije pri čemer smo se izognili križanju in odstranili nepotrebne opise tranzicij in stanj. Slika 8.1 prikazuje optimiziran model.

Slika 8.1 optimiziran procesni model na podlagi praks AM



Vir: lasten

9. ZAKLJUČEK

V delu smo sestavili proces (sl. 6.6), ki bi bil primeren za optimizacijo z DGMM. Na podlagi vprašalnika (PRILOGA I) za ocenitev optimizirane zrelosti razvojnega procesa lahko preverimo ali smo sestavili proces, ki podpira izvajanje vseh GDPA.

Ugotovimo, da proces podpira GDD in njegovo konstantno posodabljanje, če je to potrebno. Konfiguracija in upravljanje se v procesu izvajata z orodjem, ki ga ekipa izbere ob začetku razvojnega procesa. V poglavju 7.5 smo predstavili tudi postopek AHP za uspešno izbiro programskega paketa, ki lahko zajema komunikacijske in organizacijske aspekte načrtovanja. Modeliranje zahtev se izvaja z nadgrajevanjem dokumenta oblikovanja igre, ki vsebuje tudi definiranje igralnega pogona. GDD postane vodilo zahtev iz katerih se spreminjata tudi umetnikova biblija in TDD. Skozi življenjski cikel razvoja mora GDD vedno predstavljati trenutno sliko igralske izkušnje (Bates 2004, 208).

Aktivnosti vezane na prototip se izvajajo v fazah dokazovanja koncepta in inkrementalnega razvoja prototipa dokler ta ne postane končna igra. V fazi evaluacije se izvaja testiranje igranja in s tem balansiranje igranja, kar omogoča izvajanje zadnje izjave vezane na prototip.

Tveganja so zmanjšana, saj proces omogoča fazo koncepta, ki pa vsebuje navadno tudi analizo tveganja. Analiza podaja vse faktorje, kateri bi lahko negativno vplivali na projekt (Novak 2012, 389). Poleg tega smo predstavili tudi orodja za spremljanje izvorne kode, ki onemogočajo, da bi nastala nepopravljiva škoda v kodi ali odšel nadzor nad ostalimi sredstvi (zvok, slika, umetnine).

Uporaba Scrumban pristopa pokrije večji del GDPA. Dnevni sestanki omogočajo dnevno komunikacijo z zaposlenimi, podajanje težav in njihovo reševanje, kar vodi v konstantno optimizacijo dela. Poleg tega zadovoljimo tudi zadnjo izdajo vprašalnika, saj je komunikacija z upniki vzpostavljena preko Scrum vodje. Zavoljo Kanban table in funkcije WiP pridobimo možnost izračunanja ocene dela in s tem napoved zaključka nalog in projekta.

Analiza igralnosti se izvaja po vsaki končani iteraciji. V procesu spada ta aktivnost pod evaluacijo (Novak 2012, 367). Tukaj se testira igranje pri katerem se ne išče defektov ampak dejavnike, ki zmanjšujejo zabavo (Levy in Novak 2009, 33).

Proces omogoča testni protokol, ki se izvaja po fazah (Alfa, Beta) in vključuje strategijo pridobivanja povratnih informacij preden gre igra v izdajo. To podpira Beta faza v kateri lahko igro izdamo v javnost ali pa najamemo Beta testerje (Novak 2012, 360).

Proces omogoča tudi podporo uporabnikom, saj se proces razvoja nikoli ne konča. V kolikor se pojavi potreba po posodobitvi zaradi defekta, boljše vsebine ali na podlagi povratnih informacij se ponovno začne iteracija produkcije v fazi načrtovanja posodobitve. S tem proces omogoča dolgoročno vzdrževanje in ohranjanje življenjske dobe igre, kar pozitivno vpliva na donosnost naložb.

Proces je definiran tako, da podaja smernice dobrih praks v razvoju video iger. Prav tako uporabnikom ne predstavlja ovir pri izvajanju svojih odločitev, urejanju urnikov in določanju nalog, saj omogoča apliciranje poljubnih praks in orodij. Glede na to, da podporni proces vsebuje Scrum prakse razumemo, da je proces usmerjen v samoorganizacijo posameznikov. Proces je tako usmerjen na ljudi od katerih se pričakuje samoiniciativnost, poznavanje discipline in doprinos znanja, ki lahko služi kot vodilo izboljšanju procesa. Zato je izjava v vprašalniku, ki se veže na inovativnost bolj odvisna od upravljanja s človeškimi viri kot pa samega procesa.

1. Capability Maturity Model. [↑](#footnote-ref-1)
2. Capability Maturity Model Integration. [↑](#footnote-ref-2)
3. Software Process Improvement and Capability Determination. [↑](#footnote-ref-3)
4. Digital Game Maturity Model. [↑](#footnote-ref-4)
5. Game Development Process Activities. [↑](#footnote-ref-5)
6. NA – number of applicable statements. [↑](#footnote-ref-6)
7. PT – Passing treshold. [↑](#footnote-ref-7)