

Academy Profession (AP) Degree in Computer Science

**TITLE: MMO og systemudvikling mht.
drift**

PROJECT PERIOD:

Grn14dat2a,
February d. 17, 2016 - Marts d. 4, 2014

PROJECT GROUP:

15

STUDENTS:

Thomas Harald Marinussen
Morten Henning Jensen
Nicolai Bülov Nielsen

SUPERVISOR:

Palle Rosendahl Ehmsen

REPORT PAGES: ??

APPENDIX PAGES: ??

TOTAL PAGES: ??

TOTAL CHARS:

ABSTRACT:

Vedligeholdelse af et MMO er et bredt emne, lige fra hvad der definerer et MMO til hvilke problemstillinger der kan være.

Der tages udgangspunkt i MMO-genren generelt, og der bliver lavet en definition på hvad et MMO er, for at indskærpe genren.

Der vil blive kigget på, hvad der er vigtigt at vedligeholde og de områder, som er vigtige at være inde over, når styring og driften af et større spil skal holdes ajour.

For at finde ud af hvordan problemstillinger kan løses bedst, vil forskellige udviklingsmetoder blive gennemgået og vurderet i forhold til vedligeholdelsen af et MMO.

Artiklen runder af med en konklusion, som baserer sig på de forskellige informationer og kilder, der er blevet samlet og analyseret, for at finde frem til den udviklingsmetode vi mener egner sig bedst til driften af et MMO.

Indhold

| | | |
|----------|---|----------|
| 1 | Hvad er et MMO? | 2 |
| 2 | Hvor Massive er Massive? | 3 |
| 3 | Udviklingsmetoder | 4 |
| 3.1 | Vandfaldsmodellen | 4 |
| 3.2 | Anden sektion | 4 |
| 3.2.1 | under sektion | 4 |
| 4 | Hvad kræver det for at holde et MMO kørende? | 5 |
| 5 | Hvad er vigtigt at vedligeholde i et MMO? | 6 |
| 6 | Konklusion | 7 |
| A | Tabel: 1 | |

Kapitel 1

Hvad er et MMO?

MMO er en selvstændig verden, forstået på den måde, at verdenen udvikler sig uden der behøver, at være indblanding fra spillere. Det er et online spil, som kan holde til at mange spillere, der kan spille på samme tid i den samme verden[12]. MMO spil kan blive fundet og spillet på de fleste enheder med adgang til internettet, og er derfor ikke begrænset til én platform, men kan findes til både PC, konsoller og endda telefoner. Hvis man spiller et MMO ved man ikke hvor de andre personer man spiller med er fra, da folk fra hele verdenen kan spille online spil. Typiske mål i et online spil er enten at arbejde sammen om at klare en opgave, eller kæmpe mod hinanden for at vinde som et hold, eller enkeltperson. Der er mange genrer inden for MMO spil, det mest succesfulde er et MMORPG (Massive Multiplayer Online Role Playing Game)[6] (Baseret på indtjening)[1]. Der findes flere genrer end MMORPG, som er succesfulde inden for MMO genren[5], som dog ikke vil blive uddybet yderligere.

Kapitel 2

Hvor Massive er Massive?

Et af kendetegnene ved et MMO er antallet af spillere, som kan spille et spil på samme tid, i det samme miljø, men hvor er grænsen for, hvornår et spil går fra at være et multiplayer spil, til at være et massive multiplayer spil? Hvis man ser på spil, som har rigtig mange spillere, som for eksempel League of Legends(LoL)[4] eller Counter Strike(CS)[3], så mener vi ikke, baseret på vores definition af hvad et MMO er, at disse to spilgenre eksempler (henholdsvis MOBA og FPS) kan gå inden for MMO-genren. Grunden hertil er, at selvom der er mulighed for, at kunne spille et spil med alle andre spillere, så skal man lave et separat miljø (verden), som kun er tilgængeligt for bestemte og/eller et mindre antal af spillere. Eftersom begge spil (LoL og CS) har et stort community af spillere og under definitionen af massive [11], er antallet af spillere og dermed godt kan gå ind under MMO, baseret på det grundlag, så vælger vi at frasortere disse og fokusere på MMO-spil som holder sig inden for vores definition.

Vores konklusion på emnet er, hvis man er mange som kan spille sammen, på samme server(verden/miljø), vil dette kunne kaldes for et MMO.

Kapitel 3

Udviklingsmetoder

For at få et bedre overblik over, hvilken metode der vil være bedst til at styre driften for et MMO, er det relevant at kigge på forskellige udviklingsmetoder. I følgende afsnit, vil der blive kigget nærmere på metoderne; Vandfaldsmodellen, SCRUM og Unified Process(UP). Hvert afsnit vil indeholde en kort beskrivelse, samt nogle af de fordele og ulemper der er ved at bruge metoden.

3.1 Vandfaldsmodellen

Det er vigtigt her, at bemærke i figur ?, at der først testes efter udviklingen i projektet. Dette har for eksempel den konsekvens, at projektet er færdiglavet, men ikke er blevet testet igennem, for eventuelle fejl og mangler, og denne udviklingsmetode gør ikke brug af at springe tilbage i faserne, når de først er blevet lavet. Det vil sige, at når konstruktionsfasen er færdig, kan man ikke gå tilbage og ændre i den, uanset hvad for eksempel resultatet af testfasen viser. Denne metode er oftest brugt ved kortere projekter, hvor kravene er meget specifikke og der er så få parametre udefra som muligt, der kan gå ind og ændre ved projektet.

3.2 Anden sektion

3.2.1 under sektion

Kapitel 4

Hvad kræver det for at holde et MMO kørende?

Når man ser på et MMO, så er der mange ting, at tage med i overvejelserne for udviklingen. Et MMO er ofte så stort et spil, at det ikke er muligt for en enkelt programmør at holde systemet kørende selv. Dette vil sige bare i team størrelse skal man nok op omkring 100 - 200 mennesker[2], så det er svært at holde overblik og kommunikation i gang. Udover dette er der også den økonomiske side, løn til alle de programmører, penge til servere, vedligeholdelse og til eventuel videre udvikling af spillet. Der har man investorer til at hjælpe en i gang, men de penge vil de helst have tilbage igen. Til at hjælpe spilfirmaet med det har man heldigvis spillere, men for at betale et lån på 1 million dollars tilbage kræver det omkring 10,000 subscribers som hver betaler 100 kr. om måneden over 2 år[13], og så skal man helst have flere subscriber's end det, for at kunne vedligeholde omkostningerne. Det er lidt en ond cirkel når det kommer til det punkt. Der er brug for subscribers for, at skaffe penge til server vedligeholdelse, men serverne skal holdes vedlige for at spillerne. Hvis det går godt for spillet når der kommer der et overskud ud af den anden ende, og cirklen er hermed brudt.

Kapitel 5

Hvad er vigtigt at vedligeholde i et MMO?

Ofte når man spiller et online spil kommer der patches eller hotfixes ind i spil, som skal løse nogle problemer der er med spillet, eller udvide det der er i spillet. Disse problemer der kan komme i driften af ens produkt kan ofte findes af testere, men nogle gange ender det med at fejlene kommer helt ud i en endelig udgave, hvor det bliver nødvendigt at hotfixe problemet. Derfor er det vigtigt at lytte til den feedback spillerne af ens produkt kommer med, så man altid kan tilfredsstille både kunder, og evt. nye kunder.

Når man vedligeholder et MMO er det også vigtigt at komme med nye ting spilleren kan lave, som nyt content, eller noget så simpelt som nye skins til ens avatar, så spilleren har noget at lave, og de bliver ved med at have interesse i ens produkt.[14]

Man skal dog ikke kun lytte til kunder, men man skal også holde dem informeret om planer for fremtiden, og hvad de kan forvente der bliver lavet til spillet. Dette gør spil som Shroud of the Avatar[8], hvor de kommer med månedlige opdateringer om hvad de har fået lavet, og hvad de kommer til at lave det næste stykke tid[7]. Dette medfølger også at man skal være aktive på sociale medier, og man skal svare på folks kritik, og lytte til hvad de siger om disse planer, og bare generelt hjælpe dem[9]. For hvis du har et godt community, så skal de nok bakke op om dit spil, selvom der bliver taget en upopulær beslutning.[10]

Kapitel 6

Konklusion

I forhold til de problemstillinger der er ved at holde driften i et MMO, hvor der hele tiden kan være behov for ændringer og vedligeholdelse af forskellige dele af spillet vil en udviklingsmetode, der er flexible, være bedre egnet end en udviklingsmetode, der har faste rammer.

Efter vores research omkring emnet, mener vi at Scrum er den bedste måde at vedligeholde et spil på. Hvis man kigger på Scrum er det hele nemlig sat op i sprints, som ville være fordelagtige i vedligeholdelsen, da man kan planlægge fremad, med hvad der skal laves på eventuelle expansions, eller hvad der skal fixes i det spil man allerede har. På denne måde kan man også informere communityet om det man laver, imens man laver det, og derved kan man altid få feedback på det man laver. Da scrum også er en agil metode kan der komme løbende ændringer og dette er optimalt for vedligeholdelsen af et produkt. Samtidig er det muligt at kunne reagere på hvad folk har af input og lave om på ting, som måske troede ville virke, eller selv synes var en god idé, men viste sig ikke at være så god når det kom til stykket.

Vi mener derimod også at vandfaldsmodellen ville være den mindst effektive til at vedligeholde et spil, da hvis man opdager fejl i det endelige produkt, ville man skulle gå helt tilbage til designfasen, og så tjekke op derfra om der er lavet fejl, og rette det trin for trin, i stedet for at kunne hoppe direkte ind i koden og rette den linje kode, der kunne give fejlen.

Med henblik på vedligeholdelse, er det situations bestemt hvilken metode der vil være bedst at bruge. For eksempel hvis man har en klasse, der kunne

være en spilbar Race eller et våben, som er meget bedre end de andre alternativer i spillet, kan det være mere relevant at bruge vandfaldsmodellen, da den går helt tilbage til analyse- og designfasen og tager hele problematikken op igen. Hvorimod en Scrum metode ligger an til meget mere test og måske først ville gå ind og lave små ændringer og teste det igen, til problemet er løst.

Alle udviklingsmetoderne, som er gennemgået i denne artikel har alle sine fordele og ulemper, som skal tages hensyn til, når man vælger hvilken man vil gå ud fra når man ser hvad problemstillingen er.

Bibliografi

- [1] Omer Altay. *The Top 10 Most Profitable MMOs And MMORPGs*. April, 2015. URL: <http://mmos.com/editorials/most-profitable-mmorpgs>.
- [2] ApochPiQ. *Why You Shouldn't Be Making An MMO*. June, 2011. URL: <http://www.gamedev.net/blog/355/entry-2250155-why-you-shouldnt-be-making-an-mmo/>.
- [3] Vavle Corporation. *Counter Strike: Global Offensive*. 2016. URL: <http://blog.counter-strike.net/>.
- [4] Riot Games. *League Of Legends*. 2016. URL: <http://eune.leagueoflegends.com/>.
- [5] Cyber Creations Inc. *MMORPG Gamelist - All MMO Games*. Feb, 2016. URL: <http://www.mmorpg.com/gamelist.cfm/show/all/sCol/genreUC/sOrder/asc>.
- [6] Justin Olivetti. *Perfect Ten: The Top 10 Healthiest Live MMOs*. June, 2015. URL: <http://massivelyop.com/2015/06/11/perfect-ten-the-top-10-healthiest-live-mmorpgs/>.
- [7] Portalarium. *Shroud Of The Avatar Announcements and Dev Posts*. 2015. URL: <https://www.shroudoftheavatar.com/forum/index.php?forums/shroud-of-the-avatar-announcements-and-dev-posts.15/>.
- [8] Portalarium. *Shroud Of The Avatar, Forsaken Virtues*. May, 2016. URL: <http://kotaku.com/why-world-of-warcraft-lost-so-many-subscribers-1702814469>.
- [9] Bree Royce. *The Daily Grind: What Makes A Really Great MMO Community Manager?* December, 2015. URL: <http://massivelyop.com/2015/12/01/the-daily-grind-what-makes-a-really-great-mmo-community-manager/>.

- [10] Bree Royce. *Thread: The 5 worst decisions blizzard has made to WoW*. January, 2014. URL: <http://www.mmo-champion.com/threads/1446404-The-5-worst-decisions-blizzard-has-made-to-Wow-Your-opinions>.
- [11] Statista. *Most Played PC Games On Gaming Platform Raptr In November 2015*. 2016. URL: <http://www.statista.com/statistics/251222/most-played-pc-games/>.
- [12] Techopedia. *Massively Multiplayer Online Game (MMOG)*. 2016. URL: <https://www.techopedia.com/definition/27054/massively-multiplayer-online-game-mmog>.
- [13] Allen Varney. *Boutique MMOGs*. December, 2006. URL: http://www.escapistmagazine.com/articles/view/video-games/issues/issue_75/431-Boutique-MMOGs.
- [14] Gergo Vas. *Why World Of Warcraft Lost So Many Subscribers*. May, 2016. URL: <http://kotaku.com/why-world-of-warcraft-lost-so-many-subscribers-1702814469>.

Bilag A

Tabel: 1

| År | Budget | Resultat | Måneder at betale tilbage | År at betale tilbage |
|-------|---------|----------|---------------------------------|----------------------------|
| 1000 | 5000000 | 100000 | 50 | 4.2 |
| 3000 | 5000000 | 300000 | 16.7 | 1.4 |
| 5000 | 5000000 | 500000 | 10 | 0.8 |
| 8000 | 5000000 | 800000 | 6.3 | 0.5 |
| 10000 | 5000000 | 1000000 | 5 | 0.4 |