8. Dokumentációkkal kapcsolatos követelmények

A H-s sorszámmal ellátott dokumentációk az előre meghatározott időpontban. html formátumban fognak megjelenni.

Célja hogy projekt haladása rendszerezve, nyomon követhető legyen.

A weblapon elérhető a munkanapló is, ami szintén szabvány html formátumban lesz.

Ezen keresztül lehet végig követni, hogy a projekt készítésekor ki mit és hogyan csinált, meg és hogy erre mennyi időt fordított.

A játék saját weblapján Dev Diary-t készítünk, hogy a jövőbeli játékosok nyomon tudják követni a játék fejlesztését, és hogy ki tudják a véleményüket fejezni a lehetséges új tartalmakról és változtatásokról.

Ezen felül Patch Notes formában értesítenénk a játékosokat, ha valamilyen új változást kiviteleztünk a játékban.

A dokumentáció során oda kell figyelni, hogy szabványok által adott kritériumokat teljesítsük. Ez meghatározza a legfontosabb információk körét és formáját. Továbbá lehetővé teszi, hogy egyazon feladaton a fejlesztésen dolgozó emberek ugyan azt értsék, és így ne legyenek félreértésből adódó hibák.

A fejlesztés egész fázisát egy fejlesztői dokumentációba írjuk le, amely az alábbiakat tartalmazza: A feladat specifikáció, a programterv, a forrásprogram, a kész program és a teszt adatok listája. Ennek segítségével lehet csak a későbbiekben módosítani a meglévő szoftvert. Mivel így minden eleméről pontosan tudni fogjuk, hogy milyen célt szolgál és mikhez kapcsolódik.

Következő fontos dokumentum, amelyet a fejlesztés során el kell készíteni a szoftverkövetelmények dokumentuma. Ez tartalmazza a rendszerfejlesztőkkel szemben támasztott elvárások listáját. Ezt a dokumentumot használják az alábbiak: megrendelő, menedzserek, fejlesztők, rendszertervezők és a rendszert karbantartók. Ezen a megrendelők változtathatnak, mivel megnézik a követelményeket, és ha számukra nem megfelelő, akkor módosítják. Ezzel elősegítve, hogy a megrendelő a számára megfelelő terméket kapja.

Felhasználói dokumentáció: Ez tartalmazza azokat az információkat, amelyek segítenek a felhasználónak abban, hogy a szoftvert rendeltetésszerűen használják. Ez okból a felhasználóknak a szoftver használata előtt ezt a dokumentumot érdemes elolvasniuk.

A felhasználói dokumentáció több részből áll:

1. Funkcionális leírás: ennek elolvasása segít eldönteni a felhasználónak, hogy az adott szoftverre szüksége van-e. Mivel ez tartalmazza az a szoftver által biztosított szolgáltatásokat.
2. Telepítési dokumentum: Pontosan leírja a szoftver telepítésének menetét, és leírja a minimális gépigényt.
3. Bevezető kézikönyv: Tartalmazza a bevezetést. Majd leírja a szoftver szabályos használatát. Leírja, hogy miként kell elkezdeni a szoftver használatát, és hogy az egyes szolgáltatásokat hogyan kell használni.
4. Referencia kézikönyv: Tartalmazza a hibák kijavításának útmutatóját, a hiba üzenetek listáját és a hibákhoz vezető lehetséges utakat.
5. Adminisztrátori kézikönyv: Ez leírja azokat az üzeneteket, amelyek akkor keletkeznek, amikor a rendszer egy másik rendszerrel lép kapcsolatba, és hogy ezekre az üzenetekre miként kell reagálni. Abban az esetben, amikor a szoftvernek hardverekhez is köze van akkor arról a kézikönyv magyarázatott nyújt és hogy ebből a hardverből adódó problémákat miként lehet azonosítani és javítani.

A dokumentumnak meg kell felelnie pár szempontból ilyen például a karbantarthatósága ugyanis az idő múltával a szoftver fejlesztik tovább így a dokumentumban leírtak elévülnek ez miatt a dokumentumot is változtatni kell. Továbbá tartalmazni kell az elvárt és a tartalomjegyzékben szereplő információkat. Másrészt oda kell figyelni hogy ezeket az információkat helyesen tartalmazza. A felhasználók szemszögéből fontos szempont hogy a dokumentum érthetően legyen megfogalmazva. Hogy könnyen és gyorsan megértse a leírt információkat.

9.Kockázat lista

Első sorban a legnehezebb feladatunk a játék kiegyensúlyozása lesz. Mivel szeretnénk egy kihívást jelentő játékot csinálni úgy, hogy ne érződjenek lehetetlennek az egyes pályák végig játszása azoknak a játékosoknak, akik újak az ilyen játékhoz. Így a játék minden egyes elemét alaposan át kell gondolnunk. Mivel megpróbálunk különféle tapasztalattal rendelkező játékosoknak készíteni a játékot, így nagyon könnyen beleeshetünk mindkét extrém szituációba, vagyis vagy túl egyszerűre sikerülnek a pályák és nem jelentenek kihívást a veteránoknak, vagy túl nehézre sikerülnek, és ez által elijesztjük az újonnan érkezőket. Ennek a probléma megoldására két megoldással is meg tudnánk oldani:

Az első a nehezebb út, amely során a játék egy nehézségi szinttel rendelkezik. Így mint ahogy feljebb említettem, alaposan át kell gondolnunk mindent. Bár ez a megoldás megvalósítása több problémával jár, de az a nagy előnye, hogy így minden játékost egyenlően kezelünk.

A másik megoldás a különböző nehézségi szintek bevezetése lenne. Ezzel minden játékos a saját tapasztalatának megfelelően választhatna egy szintet és azon játszana. Ezt megvalósítani egyszerűbb lenne, mégis a csapatunk abban egyezett meg, hogy minden erőnkkel azon leszünk, hogy az első megoldást próbáljuk megvalósítani.

További kockázat lehet a biztonsági hiányosságok. Ugyan is szoftverfejlesztés esettén gyakori hiba hogy hiába megfelelően működik, miután piacra kerül, biztonsági hibákra derül fény. Ez az okból lehet, hogy a megrendelt követelményeket kielégíti, de a biztonsága nem megfelelő.

Továbbá a kockázatok 3 féle képen jelenhetnek meg:

1. Projektkockázat: ez befolyásolja a határidőket és az erőforrásokat
2. Termékkockázat az adott szoftver minőségét és a teljesítményét befolyásolja.
3. Üzleti kockázat: A fejlesztő céget befolyásolja.

|  |  |  |
| --- | --- | --- |
| Kockázat | Hatás | Leírás |
| Munkaerő áramlás | Projekt | A projekt befejezése előtt képzett munkaerő kilép |
| Menedzsment változás | Projekt | A cég más prioritásokkal rendelkező menedzsmentet kap |
| Hardver késés | Projekt | Nélkülözhetetlen hardver beszállítása késik. |
| Követelmények változása | Projekt és termék | A vártnál több változás a követelményekben. |
| Követelmények késése | Projekt és termék | Fontos interfészdefiníciók késnek. |
| Méret alulbecslése | Projekt és termék | A rendszer méretét alábecsülték. |
| CASE eszközök gyenge teljesítménye | Termék | A projektben használt CASE eszközük a vártnál kisebb teljesítményűek. |
| Technológia váltás | Cég | A rendszerépítéshez használt technológia elavult. |
| Konkurens termék | Cég | Konkurens terméket dobtak piacra a rendszer befejezése előtt. |

A kockázat típusai:

* Technológiai: Például: Egy szoftver nem olyan gyors, mint várták.
* Személyi: Például: Nem találnak a feladathoz eléggé képzet fejlesztőt.
* Szervezeti: Például: A projekt új menedzsmentet kap.
* Eszköz: Például: A CASE eszközt nem lehet integrálni.
* Követelmények: Például: A követelmény, amelyet a megrendelő adott túlságosan nagy, ami miatt túlzott átalakítások szükségesek.
* Becslés: Például: Alábecsülték a szoftver fejlesztésének idejét.