PANNON EGYETEM

Release Recommendation Tool

Készítette

Suborits Péter

Műszaki Informatikai Kar

Pannon Egyetem

DÁTUM

1. Bevezető

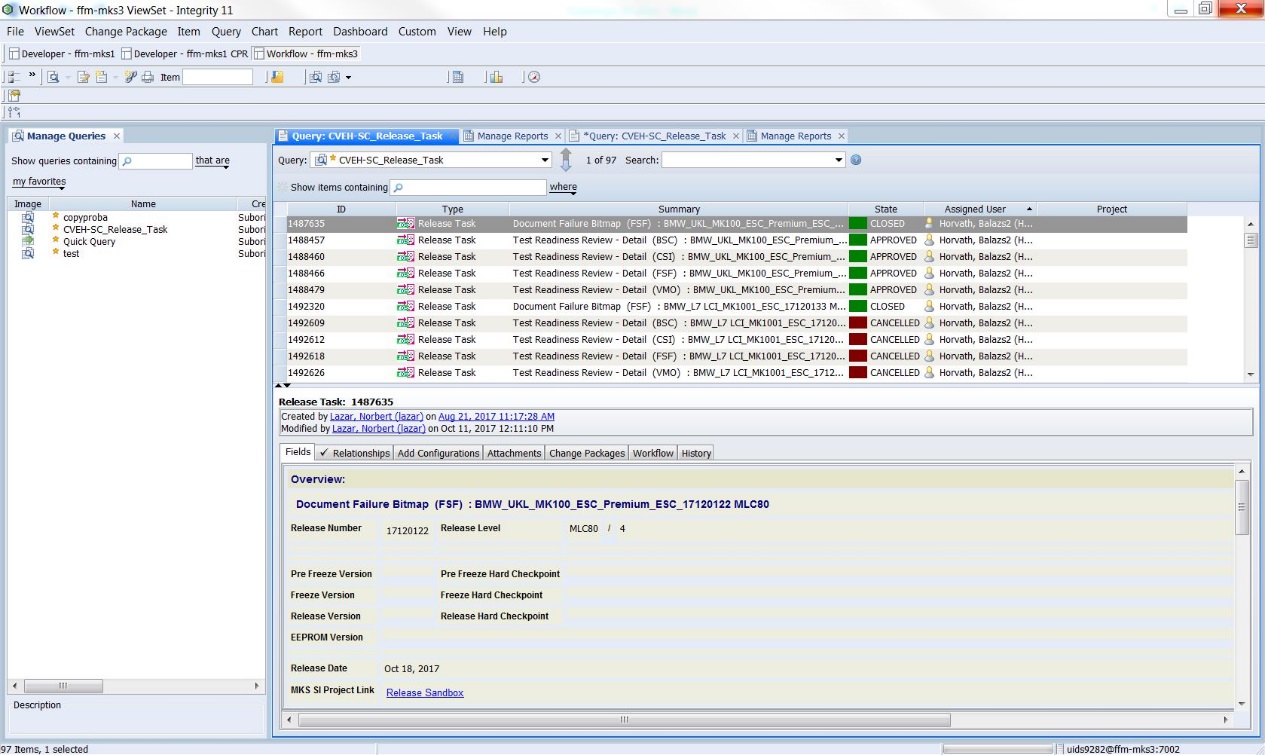
A Continental Automotive Hungary veszprémi telephelyén dolgozom diákmunkásként. A cég felépítése úgy néz ki, hogy vannak csoportok, ahol mérnökök dolgoznak, akiket egy-egy úgynevezett project manager felügyel, ellenőriz. A managereknek ez nagyon sok időt vesz el a munkaidejükből, amit dokumentációk átnézésével kell tölteniük. Ez napi szinten 20-30 perc, ami egy hónapban akár már 10 óra is lehet. Ezzel a szoftverrel, amit Zágon Ferenccel készítünk, próbáljuk napi szinten körülbelül 5 percre vagy még kevesebbre redukálni. Ezen dokumentumok ellenőrzésének nagy része automatizálható és mi ezt igyekszünk megvalósítani. Persze az idő csökkentésével semmit sem érünk el, ha megbízhatatlan lesz a program és rosszul értékeli ki az adatokat, ezért nagyon fontos szempont lesz az is, hogy a dokumentumokat, adatokat helyesen hasonlítsa, illetve értékelje ki az alkalmazás. Ez azért is nagyon fontos szempont, mivel ha rosszul értékel ki a program és esetleges hibák, ami akár szoftveres, akár dokumentációs hiba, csak a vásárlónál jönnek elő, az a vállalatnak hatalmas károkat tud okozni. Például ha egy autót rossz szoftverrel kezdenek el gyártani és ez csak forgalomba helyezésük után derül ki az összes gépjárművet vissza kell hívni és javítani a hibákat, ráadásul fennáll annak a veszélye is, hogy a hiba balesetet, súlyosabb esetben emberéletet is követelhet. Ez pedig a cég számára jelentős veszteséget okozna mind hírnévben, mind pedig pénzügyileg. Egy ilyen figyelmetlenség dolgozói oldalon pedig a dolgozó prémiumában fog meglátszani, tehát nekik is érdekük a felelősségen felül, hogy jól csinálják a feladatukat. Ezáltal nekünk is nagyon fontos hogy stabil, pontosan működő szoftvert készítsünk.

Továbbá a munkánk bonyolultságát fokozza, hogy a cégen belül nem egy szoftverstruktúrával dolgoznak, hanem kettővel, mivel volt váltás, de nem minden projekt állt át az új szoftverstruktúrára. A két szoftver a TB és a Vision Controls. Ezek nagyrészt eltérően állítják elő a kiértékelendő dokumentumokat, így valószínűleg minden dokumentumfajtára két algoritmust kell kidolgozni.

A probléma megoldására régebben már voltak próbálkozások, viszont egyik sem volt sikeres. Ez valószínűleg annak tudható be, hogy aki fejlesztette a szoftvert, nem volt a cég alkalmazásában sokáig és nem volt aki ezt a programot frissítette volna, illetve átláthatatlan volt és nagyon sok redundáns adatot tartalmazott, vagy ami kellett volna az hiányzott belőle. Mivel Ferenc már teljes munkaidőben dolgozik, és az Egyetem után én is szeretnék teljes munkaidőben a cégnél dolgozni, ezáltal mindig naprakészen tudnánk tartani a programot, illetve ha valami új dolgot kellene beépíteni, ezt el tudnánk végezni.

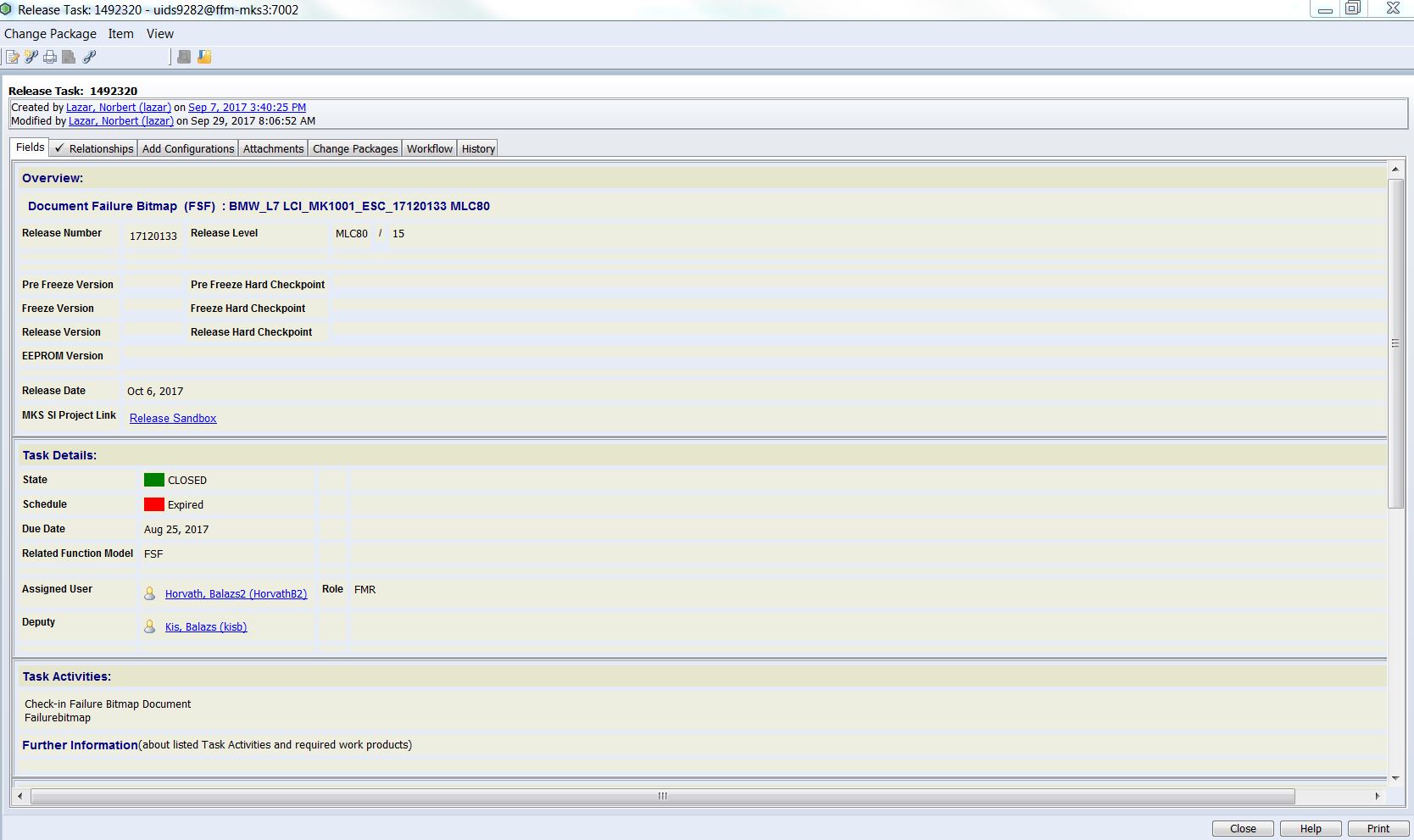
A szoftver megvalósítását Java programnyelven tervezzük, NetBeans fejlesztői környezetet használva. A dokumentációk kiértékeléséhez elképzelhető, hogy reguláris kifejezéseket is alkalmaznunk kell és ebben a nyelvben lehetőségünk van ezeket egyszerűen kezelni. Továbbá a verziókövető rendszer(MKS) is Java nyelven készült és tartalmaz egy manual-t amiben különböző parancsok vannak, hogy hogyan is lehet külső program(parancssor) segítségével elérni, dokumentumokat letölteni illetve lekérdezéseket írni és reportokat készíteni. Azért választottuk ezt a nyelvet, illetve fejlesztői környezetet, mivel felhasználóbarát, könnyen átlátható kezelőfelületet tartalmaz, valamint kényelmes használni. Mivel az Egyetemen órai keretek között már használtam a programot, ezért nem ismeretlen számomra.

A rendszer készítése során folyamatosan beszélünk a Continental-on belüli illetékesekkel, vezetőinkkel, hogy a projectünk állandóan nyomon követhető legyen, illetve az esetleges hibákat, félreértéseket elkerüljük. A megvalósítást próbáljuk ésszerűen felépíteni, pontról-pontra haladni, kezdve elsődlegesen az információszerzéssel, hogy a leendő felhasználóknak milyen igényei vannak a szoftverrel kapcsolatosan. Ezután az adminisztrációs munkához kellő dokumentumokról szerzünk információkat, hogy milyen szempontok alapján kell összehasonlítani őket, illetve mi az ami kell belőle és mi az ami nem, és mi elfogadható és mi nem az adott dokumentumoknál. Miután megkaptunk minden információt az elvárásokkal kapcsolatban, és tudjuk a pontos definíciókat, szabályokat egy-egy adott ellenőrzési folyamathoz, Ferenccel szétosztjuk egymás között ezeket a kiértékeléseket. Ezzel párhuzamosan Ferenc a felhasználói interfacet fogja megtervezni, jómagam pedig a Continental belső verziókövető rendszeréből való lekérdezéseket illetve dokumentum letöltését, valamint az MKS-sel(verziókövető rendszer) való kommunikációt fogom implementálni .

A szoftverfejlesztők egy verziókövető rendszert(MKS) használnak a dokumentációk és programkódok tárolására. Ebben az adatbázisban/rendszerben, több tíz-száz ezer fájl/adat/dokumentáció van, ebből kifolyólag ha sokan dolgoznak benne lassú is. 

*1.1 ábra*

Az 1.1-es ábrán látható az MKS kezelőfelülete. Itt látható a fejléc, ahol a program készítéséhez a query és report füleket biztosan fogjuk használni, csak parancssoros megoldással. Mivel említettem, hogy lassú a rendszer ezért elengedhetetlen, hogy jól optimalizált, és helyesen működő query-ket készítsünk, az nem elég, hogyha csak működik. A query-k készítése során több száz opció alapján van lehetőség szűrni, ezek közül fontos, hogy olyanokat válasszunk ki ami az adott részhez kell, és ne legyenek benne felesleges adatok, mivel ez is növeli a lefutási időt. A query készítés persze még csak arra elegendő, hogy az MKS-en belül kilistázza az adott feltételeknek megfelelő task-okat. Ha a query megfelelően működik utána az alapján kell futtatni a report-okat amik generálnak egy a beállításaiban beállított fájltípusnak megfelelő fájlt. Itt nagyon sok ellenőrzésre lesz szükség, hogy minden esetben megfelelően futottak-e le a reportok, mivel ha valahol hiba volt és nem sikerült a futtatása ez a kiértékelésben is gondokat okozhat, vagy hibás adatokat generálhat.



*1.2 ábra*

Az 1.2 ábrán látható egy task, ami számos fontos adatot tartalmaz a szoftverünk szempontjából. Egy ilyen taskban találhatóak a fájlok amiket össze kell hasonlítanunk.

??????

2017 nyarán a szakmai gyakorlatomat a Continental Automotive Hungary KFT-nél töltöttem, ami két hónap volt. Maga a cég és a munkakörnyezet nagyon barátságos, mindenki segítőkész. Ezért is döntöttem úgy, hogy iskola mellett, diákmunkában vállalok egy kis pluszt. Kaptam egy nagy feladatot, amit Szakdolgozat gyanánt elkészíthetek itt a Continental-nál. Egy kevés információt szeretnék megosztani a szakmai gyakorlatomon szerzett tapasztalataimról, illetve hogy hogyan is jött ez a lehetőség, hogy elkészíthessem ezt a programot és mi a célja, mire lesz jó ez az applikáció.

Szakmai gyakorlatom során az excellel kellett foglalkoznom azon belül is makró kódokat írni. A feladatokhoz általában egyszerűbb algoritmusokat, tömbkezelést, illetve fájlok közti ugrálást, létrehozást, feldolgozást kellett elkészítenem. A kész kód körülbelül 5000 sorosra sikeredett, amit 6 hét alatt készítettem és a mai napig fejlesztés alatt áll, mivel mindig kerülnek elő olyan dolgok amik jók lennének ha benne lenne és tudná kezelni a program. Nagyon sok funkciót lát el többek között képes egy verziókövető rendszerrel kommunikálni és adatokat letölteni róla majd feldolgozni, amit például majd a címben említett applikációhoz is használni kell. A gyakorlat során megismerkedtem ennek a verziókövető szoftvernek a felépítésével, hogyan lehet benne úgynevezett query-ket és report-okat létrehozni illetve parancssorral futtatni. A makrónak körülbelül 100 excel fájlt kell átbogarásznia és kinyernie belőle az adatokat, illetve nagyon sok excel-t előzetesen a verziókövető adatbázisából állít elő megfelelő query-k alapján. 6 féle algoritmust kell alkalmazni hogy az összes excel-t tudja kezelni és a megfelelő helyről tudja az adatokat kinyerni.