

Text Engine

Projektplanering utifrån TDD:

Vi började med att diskutera om hur vår UML och Flowchart bör se ut. Vi började med att strukturera dem i Lucidchart. Därefter började vi skriva Unit-tester för att sedan skriva lagom mycket kod, och sedan upprepa processen tills metoden är klar. Därefter gör vi några enkla faktureringar och kod-optimering. Sedan la till kommentarer/Javadoc.

Bestäm en sortering algoritm:

Vi valde Quick-sort eftersom de verkar vara den mest effektiva sorterings-algoritm i de flesta fall.

Detta utifrån de vi lärde oss från lektion med Rickard där vi fick se jämförelse på de olika algoritmerna, och fick känslan av att Quick-sort rent allmänt är den mest effektiva algoritmen.

Vi valde även att ha insertion sorting eftersom enkelheten och att det ska vara effektiv med att sortera sökning resultat.

Andra Algoritmer:

Vi har även valt att ha två sökalgoritm, en av dem använder sig av den linjära algoritmen. Den är enkel metod som passar sig med osorterad data som kan inträffa i fall användaren valde att leta efter ett ord utan att sortera texten först.

På grund av mjukvarans natur bestämde vi oss att ha med en effektiv söknings algoritm som anpassar sig bättre med sorterad data struktur än den linjära. Den andra blev den så kallade binary algoritmen.

Testkrav:

- Alla uppgiftens krav måste funka.
- Ge tydlig och enkel upplevelse för användaren.
- Innan vi börjar sortera tar vi bort onödiga tecken för att få ett noggrannare resultat på sökningen.
- Ord ska gå att söka på ord utan att texten först är sorterad.
- Programmet ska inte krascha.

Vilka Tester - varför:

1. Unit:
 - Identifiera problem/buggar från början.
 - Vid ändringar i funktioner/algoritmer kan testerna identifiera ifall den nya logiken inte stämmer med tidigare uppsatta "regler".
 - Obligatoriskt.
2. Integration:
 - Hitta buggar i kombinationen av funktioner.
3. System:
 - Hitta buggar i helheten.
4. Beta:
 - Hitta möjliga buggar som program testerna eller vi inte hittat.
 - Få reflektion om mjukvaran av användaren.

Resultat av tester:

Resultat av Unit tester:

- Alla tester gick igenom

Resultat av integration tester:

- Alla tester gick igenom

Resultat av system tester:

- Alla tester gick igenom

Resultat av beta tester:

- Vi fick inte många att buggtesta.
- Hittade 'exception' vid sortering av text som innehåller ord av specialtecken. Löste problemet genom att fixa buggen i 'removeSigns'-metoden.
- Fick förslag att ha med ett meddelande efter sortering för att förbättra UX. Vilket vi la till.

Resultat av kvalitetssäkring:

Vi har följt TDD-planeringen. Vi skapade en UML och flowchart. Vi gjorde Unit Tester och byggde programmet utifrån dem. Därefter utförde vi integration och system-test. Vi identifierade inga större problem med dem. Vi utförde en beta test. Vi valde också att ta bort smoke-test från sprinten, eftersom vi ansåg att den kändes onödigt med det aktuella projektet.

På slutet av projektet fick vi en defekt med att exekvera testerna i IntelliJ på grund av junit paketet misstänkte vi. Problemet var att medan man hämtar koden från github så identifiera IntelliJ TextEngineBetaTest.jar som ett bibliotek och la den under dependencies fält (File > Project structure > Modules > Dependencies). Lösning var bara att radera det därifrån.