

D2: Delaflevering 2

Den anden delaflevering består i: **1)** jeres arbejde med første sæt af opgaver som I planlagde i første delaflevering, **2)** unittest og accepttest, **3)** agilt design af programmet, **4)** planlægning af næste iteration i jeres projekt.

Rapporten skal bestå af følgende dele som i alt må fylde maksimum 8 sider (inkl. figurer):

1. Implementering af opgaver

I skal implementere de opgaver som I har planlagt for denne periode. I kan oprette opgaver som *issues* i GitHub og bruge dem til at koordinere arbejdet hen imod næste delaflevering; se f.eks. *milestone*- og *assignee*-attributterne. Det er op til jer i gruppen hvordan I vil gennemføre opgaverne.

Redegør kort for hvordan I løbende har planlagt og koordineret arbejdet i denne iteration (eller iterationer, hvis I har planlagt sådan) i gruppen. Giv en kort sammenfatning af hvilke *use cases* I har færdigimplementeret i denne delaflevering.

2. Afprøvning

I skal implementere unittests der dækker de metoder I har implementeret op til denne delaflevering. I skal i denne iteration også have brugt en test-først tilgang til implementeringen af mindst én af jeres *use cases*, dvs. skrevet unittest før I har skrevet programkode der implementerer funktionaliteten. Redegør for i hvilket omfang I har brugt en sådan test-først tilgang og for jeres oplevelser (har I erfaret problemer, fundet det nyttigt, etc.).

I skal også inkludere accepttest for de use cases som I har implementeret (eller er begyndt at implementere) op til denne delaflevering.

I skal inkludere en testrapport i bilag som viser resultatet af både unittest og accepttest.

3. Design

Redegør for jeres design i denne delaflevering. Beskriv først de objekter, og deres indbyrdes ansvar og samarbejde, som er centrale i implementeringen af de vigtigste *use cases*. Brug UML-diagrammer hvis det er nødvendigt.

Redegør dernæst for de væsentligste abstraktioner og designmønstre i jeres design. I skal forklare hvordan de abstraktioner og designmønstre har gjort programmet lettere at forstå (hvad har I gjort for at begrænse unødigt kompleksitet), hvor

fleksibelt jeres program er for ændringer og videreudvikling (dvs. hvor ændringer kan begrænses til få steder), mv. Forklar jeres valg med henvisning til SOLID-principperne fra [Agile, kapitel 7-12] hvor det er relevant.

4. Planlægning af næste iteration

Beslut hvilke *use cases* I vil implementere i næste iteration (ved herunder at dele dem op i opgaver som kan estimeres). Redegør for hvordan I har planlagt, estimeret og evt. beregnet jeres hastighed (jf. *velocity* i [Agile]). Vedlæg en liste over opgaver som I planlægger at implementere i næste iteration.

Referencer

- [Agile] Robert C. Martin and Micah Martin. *Agile Principles, Patterns, and Practices in C#*, Pearson Education, Inc., 2007.