

INDLEDENDE PROGRAMMERING (V22)

Christian Gram Kalhauge

TILLYKKE!

PRAKTISKE TING

- Lokale (BYG 308 LOK 1, 8 og 9)
- Mig (BYG 321-032 / <mailto:chrg@dtu.dk>)
- Hjælpelære (Anton + Marcus)
 - 10:00 - 12:00 og 13:00 - 15:00
- Discord <https://discord.gg/GaP2qZfpuk>

OVERBLIK

- Jan 2. Gruppedannelse
- **Wednesday Madness**
 - Jan 4. M1: Kundemøde
 - Jan 11. M2: Minimal Viable Product
 - Jan 18. F1: Ship it
- Jan 20. Aflevere Rapport

GRUPPEDANNELSE

- Grupper 4-6 personer.
- kl 15:00, Gruppekontrakt.
- Nye grupper, hvis I har lyst.

GRUPPEKONTRAKT

- Maks 1 A4 side.
- Gruppenavn, navne og kontakt oplysninger.
 - Github, Studienr, Discord, og Tlf nr?
- Forventnings afstemning.
 - Indsats, Tid, Steder, og Resultater.
- Konflikter bliver løst ved hjælp af kontrakten.
 - Kode ejerskab, sygdom og udsmidning.
- Ikke en del af bedømmelsen.
- Eksempel: tinyurl.com/2apbdb3w

PROJEKTET

- Kunden har udbudt at få lavet et Matador spil.
- Den 4. Januar: M1: Kravsanalyse
- Den 11. Januar: M2: MVP klar.
- Den 18. Januar: F1: Projektet shipper.
- Den 20. Januar: F2: Rapport.

KRAVSANALYSE

- Krav præsenteres som et nummer, en “user story” US, og en eller flere “acceptance test”:
 - <Hvem> skal <Hvad> sådan at <Hvorfor>.
 - K1: En spiller skal modtage penge når de passere start, sådan at der bliver flere penge til rådighed i spillet over tid.
 - Givet <Forudsætning> så <Resultat>.
 - K1_AT: Givet at en spiller passere start OG ikke er på vej i fængsel så skal spiller have 4000 kr mere.
- Kravene præsenteres (10-15 min) og afleveres i papir form.
- Kunden kommer muligvis med flere krav.
- Prioriter med kunden hvad der skal laves til MVP.

MVP: MINIMAL VIABLE PRODUKT

- Ugen efter præsentere I den første version a produktet.
- Spillet skal som udgangspunkt være spilbart.
 - Men måske kedeligt?
- Kunden kommer muligvis med flere krav.

PROJEKTET SHIPPER

- Ugen efter igen, skal produktet shippes.
- I skal indikere hvilke user stories I har nået.
- Tag'er med v1.0 og gemmer hashet:
 - 1cc5...a7c9
- Kunden vil bede om at få demonstreret "Acceptance Test", for nogle af de user stories i har nået.

RAPPORTEN AFLEVERES

- Fredag den sidste dag, skal rapporten afleveres.

RAPPORTEN

- Indledning
- Kravsanalyse
- Planlægning
- Design
- Implementation
- Validering
- Diskussion

FORMALIA

- Navn og studienummer på alle gruppe medlemmer.
- Et link til jeres Github page, samt commit hash.
- En beskrivelse af fordelingen af arbejdet.
 - Enten laves en tabel med gruppemedlemmer som søjler og sektioner som rækker. Skriv point (tid) brugt i hver sektion og summer søjlerne.
 - Ellers, skrives bare at arbejdet er fordelt ligeligt.
- Max 36 normalsider (2400 anslag) ex. Appendix og Diagrammer

INDLEDNING

- En indledning er en kort beskrivelse af settingen, problemet og løsningen.
- Brug fremad rettede referencer til at underbygge jeres konklusioner og give resten af jeres rapport struktur.
- Efter at have læst indledningen skal det være klart hvilken karakter i burde få.

KRAVSANALYSE

Her undersøger I hvad kunden vil have.

- Et overblik over domænet.
 - Identificer navne ord og lav en domæne model.
- En eller flere tabeller med alle de aftalte til systemet.
 - En US er et nummer, en user story, eventuelt prioriteter.
- En eller flere tabeller med accepttests.

Beskriv hvorfor I har valgt de krav I har, hvorfor I føler at acceptance testene er tilstrækkelige og hvilke krav har den højeste prioritet.

PLANLÆGNING

Her planlægger I forløbet. I indeler tiden i de to uger og så estimere I hvor meget I kan nå.

- Beskriv hvilke user stories I har prioriteret til de forskellige deliverables.
- Estimer hvormed meget tid som hver US tager.

DESIGN

Her beskriver i designet af systemet. Fokuser på de overordnede træk.

- Giv en introduktion til strukturen af jeres kode.
- I kan også bryde jeres user stories ned i use cases hvis I føler det er illustrerende.
- Beskriv designet i tekst og illustrer evt. med diagrammer.

IMPLEMENTATION

Her skal I beskrive hvad I har lavet.

- Et klasse diagram.
- Screen shots af fede ting I har tilføjet.
- Hvis i har gjort brug af GRAPS eller andre patterns, så beskriv dem med diverse UML diagrammer.
- Udvalgte kode eksempler.

VALIDERING

Her skal I bevise at det I har lavet det godt nok.

- En liste over acceptance tests, der er blevet udført
 - Hvornår, af hvem, og lykkedes de?
- En beskrivelse af brug af unit-test
 - Code coverage når man kører unit-cases.

DISKUSSION

Her skal I fortælle hvordan det er gået.

- Beskriv hvad I har lært,
- hvad I ville gøre bedre,
- og hvad de næste trin er hvis I havde mere tid.

Q/A