

Software Engineering og Testing

Introduksjon

Agenda

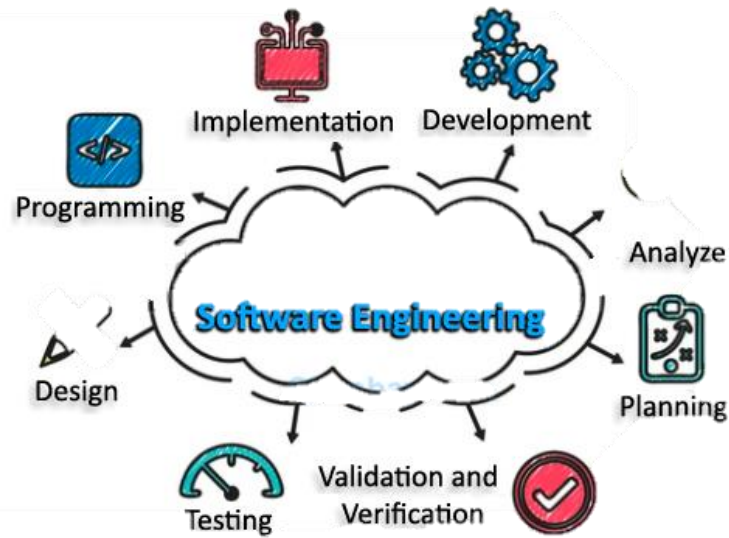
Kursinformasjon

Eksamen

Hva skal vi
lære?

Prosjektoppgaven

Kursinformasjon – hva skal du lære?



- ▶ Ulike faser og aktiviteter i software engineering
- ▶ Ulike metoder, modeller og teknikker for systemutvikling
- ▶ Ulike testeteknikker og verktøy
- ▶ Forskjellige typer dokumentasjon som benyttes i utviklingsprosessen

Kursinformasjon - “Forelesninger”



- ▶ Onsdager 10:15 - 12:00
- ▶ Fredager 10:15 - 12:00

- ▶ Øvingstimer
 - ▶ Obligatoriske innleveringer, snakk med foreleser
 - ▶ Veiledning med studass

Kursinformasjon - canvas

- ▶ Det meste av informasjon samles i Canvas
 - ▶ <https://hiof.instructure.com/courses/6299>
- ▶ Canvas - App
 - ▶ [Android](#)
 - ▶ [iOS](#)
- ▶ Presentasjoner





Kursinformasjon - Arbeidskrav

- ▶ Lagt opp til 4 innleveringer
- ▶ 3-4 ukers mellomrom
- ▶ Alle må godkjennes for å få gå opp til eksamen
- ▶ Ingen utsettelse (krever legeerklæring)
- ▶ Leveres i Canvas



Kursinformasjon - Arbeidskrav

- ▶ Ingen muligheter for å levere på nytt hvis den ikke blir godkjent
- ▶ Plagiering = utestengelse i minst 6 måneder.
- ▶ Gruppen tildeles av dere/forelesere, det gjøres ikke endringer på disse!
- ▶ Timelister og refleksjonsnotat - jobber du ikke får du ikke godkjent!
- ▶ Har du spørsmål? Spør forelesere!

Kursinformasjon - Spørsmål

Faglige spørsmål

- Forelesere
- Canvas-diskusjon

Paktiske spørsmål

- E-post
- Kontoret til forelesere
- Canvas-chat

Problemer som student eller i emnet

- E-post til forelesere
- Kontoret til forelesere
- Studentrådgivningen

Tekniske problemer

- itvakt@hiof.no
- IT-vakt

Kursinformasjon - Faglitteratur

- ▶ Engineering Software Products -
An Introduction to Modern Software
Engineering
- ▶ Ian Sommerville



EKSAMEN

4 obliger må være godkjente

Prosjektoppgave i gruppe, teller 45%

3 timer skriftlig eksamen, teller 55%

Hva skal dere lære?



Emnebeskrivelse



Referansegruppe

Prosjektoppgaven

- Hensikt

- ▶ Sørge for at dere får erfaring med å jobbe som et utviklingsteam
- ▶ Dere er selv ansvarlige for å begrense oppgaven og prototypen deres, og selv velge form på det dere leverer
- ▶ Lag skriftlig forslag til løsning og en kravspesifikasjon av det komplette systemet
- ▶ Dokumentasjonen beskriver et større system, mens prototypen viser frem en del av systemet
- ▶ Fritt valg av programmeringsspråk

Selve oppgaven!

I dagens samfunn ser vi en stadig utvikling innenfor IoT og smartHus. Det er stadig flere som benytter slike smarte løsninger for å spare penger eller gjøre hverdagen mer sømløs. Dessverre ser vi også at dette per nå er forbeholdt personer med høy teknisk kompetanse og spesielle interesser for feltet.



En oppstartsbedrift ønsker å endre dette markedet og gjøre det enklere for mannen i gata å dra nytte av denne teknologiske utvikling.



Oppstartsbedriften ønsker derfor å lansere en plattform som gjør det mulig for mennesker også uten særlig teknisk kompetanse å benytte slik teknologi i sine hjem.

Prosjektet

- ▶ I dette prosjektet skal dere lage en prosjektdokumentasjon og en prototype for et produkt for en oppstartsbedrift.
- ▶ Denne oppstartsbedriften (kunden deres) ønsker å få laget en løsning som gjør det mulig for hvermannsen (brukeren) å benytte seg av teknologiske smarte komponenter i sine hjem.
- ▶ Oppstartsbedriften har imidlertid ingen egen, intern IT-kompetanse, men ønsker et forslag for hvordan et slikt system kan være og hva det skal inneholde.

- ▶ Dere skal beskrive kjernesystemet og miljøet rundt, samt definere hvilke funksjoner som er påkrevd for at oppstartsbedriften (kunden) får det systemet de etterspør.
- ▶ I tillegg skal det utvikles en prototype (et MVP) som implementerer et utvalg av funksjonene i systemet. Dere velger selv hvilke hovedfunksjoner dere vil vise frem i prototypen, så lenge prototypen har et visst omfang og vil kunne ha verdi for kunden.
- ▶ Det skal også lages nødvendige, automatiserte tester som dokumenterer at prototypen tilfredsstiller de viktigste kravene som dere har avdekket og som blir beskrevet i dokumentasjonen dere skal produsere for systemet.
- ▶ Det er viktig at dere passer på å dokumentere eksterne avhengigheter i systemet, dvs. der systemet må prate med systemer fra andre tjenesteleverandører (f.eks. betalingsløsning).
- ▶ Dere skal IKKE integrere mot noen eksterne leverandører i prototypen, men bør skrive små «stubs» som gir dere muligheten til å teste funksjoner som avhenger av dem.

Fritt valg av språk og rammeverk

- ▶ Det kan være hensiktsmessig å velge noe mange på gruppa kan hjelpe til med.
- ▶ Husk også at studassene ikke nødvendigvis kan hjelpe med detaljer i alle språk og rammeverk.
- ▶ Spør studassene på forhånd om dere er usikre og tror dere kommer til å trenge mye teknisk hjelp underveis!

Forslag til verktøy

Versjonskontroll

- Github
- Bitbucket

Prosjektstyring

- Github
- Trello
- Notion
- Microsoft planner

Diagrammer

- Plant UML

Dokumentskriving

- Overleaf
- Microsoft word
- Google Docs

- ▶ Husk at dokumentasjonen skal være forståelig for personer uten inngående kunnskap om fagområdet eller programmering!
- ▶ Implementasjonsdetaljer har sjelden noen plass i prosjektdokumentasjonen!
- ▶ Bruk diagrammer og modeller der det er relevant for å gjøre ting mer forståelig!
- ▶ Informasjon om prototypen i prosjektdokumentasjonen bør sjelden være mer enn et par-tre sider, prototypen i seg selv bør være selvforklarende!



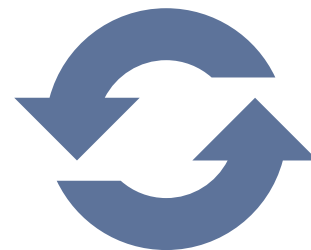
TIPS

Prosjektoppgaven - Innlevering



Prosjektdokumentasjon

Beskrivelsen av systemet, slik som forslag til arkitektur, krav, teknologier. Alt skal være dokumentert på et hensiktsmessig nivå med tanke på bruk



Prototype

Dere skal levere en fungerende prototype av det dere mener er en viktig del av systemet. Prototypen skal inneholde nødvendige tester for å vise at kravene i dokumentasjonen er oppfylt.

Innlevering forts.

- ▶ Kildekoden til prototypen skal leveres som et Git-repository som viser utviklingen av prototypen over prosjektperioden. Pass på at .git-mappen blir med i innleveringen deres, og at dere har committet alle endringene før dere leverer.
- ▶ Sørg for at det er beskrevet hvordan en person uten dyptgående IT-kunnskap kan bygge, kjøre og teste prototypen deres.
- ▶ Avhengigheter i applikasjonen deres skal kunne installeres automatisk ved hjelp av et pakkesystem for språket eller rammeverket dere benytter (f.eks. maven eller gradle om dere bruker Java, eller pip eller poetry for Python).

Evaluering



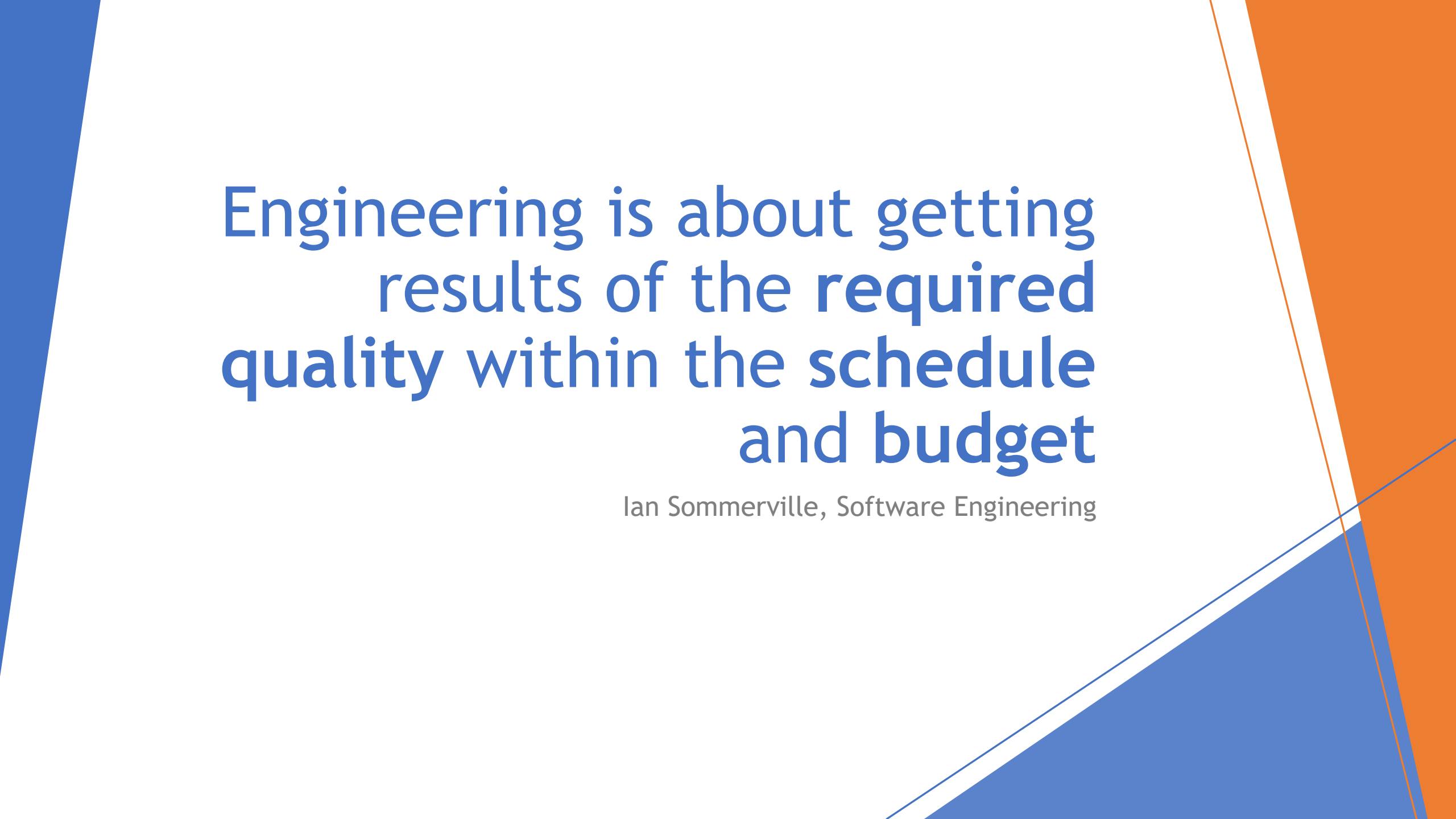
Innleveringen evalueres ut fra egnethet for produksjon, kvalitet på prototypen, testing av funksjoner i prototypen opp mot krav og prosjektdokumentasjonen.



Prosjektet teller 45%



Det gis individuell karakter A-F



Engineering is about getting
results of the **required**
quality within the **schedule**
and **budget**

Ian Sommerville, Software Engineering

I løpet av
semesteret
skal vi gå
hele veien!



Kravspesifikasjon

Hva skal vi lage?

- Krav
- Estimering
- Modellering



Applikasjonsutvikling

Hvordan lager vi det?

- Tester
- Løs kobling
- Refactoring



Vedlikehold, drift og produksjon

Hvordan difter vi
det?

- CI/CD
- Containerization
- Orchestration