

# Gruppetime 2

01.01.18



Bli med i Slack-kanalen vår hvis du enda ikke har gjort det!

[https://join.slack.com/t/in2001/shared\\_invite/enQtMzAyNTQ4NiI0NTAwLTUyMjBjZWZiNDM1YTVkMGZkMGMyYzJmYjk1M2NIZGQyNGNmOWM0Mzc1ODM4NTM5NzY](https://join.slack.com/t/in2001/shared_invite/enQtMzAyNTQ4NiI0NTAwLTUyMjBjZWZiNDM1YTVkMGZkMGMyYzJmYjk1M2NIZGQyNGNmOWM0Mzc1ODM4NTM5NzY)

# Gruppenindeling

## Gruppe 1

reino  
tobiano  
sodaniel  
simenjj  
tageg  
anastado  
zenonjm

## Gruppe 2

benjbr  
linneak  
tkgolber  
alexals  
steffa  
erlenste  
astridok

## Gruppe 3

elizabl  
rutadh  
melatft  
mikaekl  
brageja  
joneue

## Gruppe 4

eliassan  
simonpe  
pnraschh  
dilekd  
jonathdb  
bejnamsm  
nlhiorth

# Git og Github

## - Hva er git?

- Git is a type of **version control system** (VCS) that makes it easier to track changes to files. For example, when you edit a text file, git can help you determine exactly *what* changed, *who* changed it, and *why*. It's useful for coordinating work among multiple people on a project, and for tracking progress over time by saving "checkpoints". You could use it while writing a term paper, or to track changes to artwork and design files.
- Git isn't the only version control system out there, but it's by far the most popular.
- Mer info: <https://hackernoon.com/understanding-git-fcfd87c15a3>

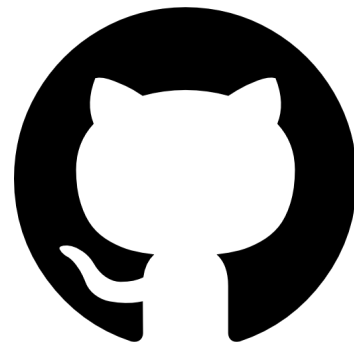
- Git kommer allerede med Mac OSX/Linux, men må installeres på Windows:  
<https://www.atlassian.com/git/tutorials/install-git#windows>

## - Hva er github?

- Github er *ikke* git. Github er en nettside som kan hoste prosjekter som bruker git.

# GitHub

1. Gå til <https://github.uio.no/>
2. Logg inn med UiO-brukeren din
3. Punkt 4-6 trenger bare én i hver gruppe å gjøre
4. Klikk på +-tegnet øverst i høyre hjørnet → new repository. Kryss av for at det er private.
5. Kall repositoryet *IN2001-GruppeX*
6. Gå inn på repositoryet → Settings → collaborators og legg til resten av gruppen og begge gruppelærerne (andrhst, elinhage) med UiO-brukernavnene
7. Sjekk at alle har fått tilgang til den
8. Lag et nytt, privat repository kun for deg som du kan bruke i forelesningen på tirsdag.



# Postman

1. Gå til <https://www.getpostman.com/>
2. Last ned korrekt versjon basert på ditt operativsystem (Windows, Mac, Linux)
3. Innstallér på vanlig måte
4. Lag bruker hvis ønskelig, ellers fortsett uten bruker



# Postman forts.

- Velg GET, og lim inn en nettadresse (f.eks. VG)
- Velg “Pretty” for javascript-koden, eller “Preview” for nettsiden

Dette kan også gjøres for API'er som inneholder data i databaser. Hvert kall til et API-endpoint skal returnere data (eller en http-feilkode)

- (I Postman) GET mot dette API'et for hunder → <https://dog.ceo/api/breeds/image/random>
- Liste av åpne API'er → <https://github.com/toddmotto/public-apis>

# Postman - Oppgaver

- API for TV-database → <https://www.tvmaze.com/api>

## Oppgaver:

- Oppg.1 Finn id'en til TV-serien Dexter
- Oppg.2 Finn navnet på skuespilleren som spiller Dexter Morgan
- Oppg.3 Finn navn og premieredato for den 4. episoden av Dexter sin 6. sesong



## Postman-oppgaver: Fasit

Oppg.1 Finne dexter: <http://api.tvmaze.com/search/shows?q=dexter>

Oppg.2 <http://api.tvmaze.com/shows/161/cast> 161 er dexter sin ID (funnet via oppgave 1), første person er Michael C. Hall

Oppg.3 <http://api.tvmaze.com/shows/161/seasons> for Sesong6-iden 724, deretter <http://api.tvmaze.com/seasons/724/episodes> for episode-listen til sesong 6. Scroll ned til ep.4, les navn og dato.

# Oppgave 1

a) Hva er en kravspesifikasjon?

b) Hvorfor er det nødvendig å lage en kravspesifikasjon?

# Løsningsforslag

1a) Hva er en kravspesifikasjon

Et dokument som spesifiserer hva som skal lages.

- Bruker- og systemkrav

- Ikke et designdokument

- Sier ingenting om hvordan oppgaven skal løses

Ofte en del av kontrakten

Ulike standarder brukes, feks IDEEE

# Løsningsforslag

## 1a) Hva er en kravspesifikasjon

### Brukerkrav

- Uttrykkes på naturlig språk / diagrammer

- Viser ønsket funksjonalitet og hvilke føringer som gjelder (kvalitetsegenskaper)

- Kunden skal kunne forstå disse kravene uten problemer

### Systemkrav

- Detaljert beskrivelse av systemfunksjoner og hvilke føringer som gjelder

- Definerer hva som skal implementeres

- Basis for kontrakt mellom kunde og leverandør

# Løsningsforslag

1b) Hvorfor er det nødvendig å lage en kravspesifikasjon?

Basis for anbud

- Kan tolkes på ulike måter

- Skape interesse og tiltrekke seg leverandører som kan møte behovet

Basis for kontrakt/design/implementasjon

- Leverandør og kunde blir enige om hva som skal lages

Spesifiserer ytelsen

- Skape felles forståelse av systemet/ enighet om hva som skal leveres

- Forhindrer eventuelle konflikter –Ulike forventninger

# Oppgave 2

- a) Hva er hendelsesdrevet modellering?
- b) Hva er et tilstandsdiagram?
- c) Hvorfor er det nyttig å lage tilstandsdiagrammer?

# Løsningsforslag oppg.2

a) Hva er hendelsesdrevet modellering?

# Løsningsforslag oppg.2

a) Hva er hendelsesdrevet modellering?

- Viser hvordan systemet responderer på eksterne / interne hendelser
- Viser hvordan tilstander er knyttet til hendelser i et system
- Spesielt nyttig for sanntidssystemer



# Løsningsforslag oppg. 2

b) Hva er et tilstandsdiagram?

# Løsningsforslag oppg. 2

b) Hva er et tilstandsdiagram?

- Representerer en tilstandsmaskin
- Grafisk representasjon av de ulike tilstandene et system kan være i
- Beskriver hvilke hendelser som får systemet til å endre tilstand
  - Altså: Hvordan går vi fra én tilstand og til en annen?

# Løsningsforslag oppg.2

c) Hvorfor er det nyttig å lage tilstandsdiagrammer?

# Løsningsforslag opppg.2

c) Hvorfor er det nyttig å lage tilstandsdiagrammer?

- Viser hvilke tilstander systemet kan være i
- Nyttig for å visualisere hvordan systemet kommer til å se ut
- Viser hvordan vi beveger oss mellom tilstander
  - Triggere
  - Forutsetninger
- Viser “deadlocks” i systemet
  - Tilstander vi ikke kan rømme fra

# Oppgave 3

Scrum har blitt en vanlig prosessmodell for utvikling av programvare.

- a) Gi en kort beskrivelse av rollen Scrum master og beskriv hvilke(n) smidige prinsipper denne rollen understøtter og på hvilken måte den gjør det.
  
- b) Gi en kort beskrivelse av teknikken «Retrospective» og beskriv hvilke(n) smidige prinsipper denne teknikken understøtter og på hvilken måte den gjør det.

# 12 smidige prinsipper

1. Vår høyeste prioritet er å tilfredsstille kunden gjennom tidlige og kontinuerlige leveranser av programvare som har verdi
2. Ønsk kravendringer velkommen, selv sent i utviklingen. Smidige prosesser bruker endringer til å skape konkurransefortrinn for kunden.
3. Levere fungerende programvare hyppig, med et par ukers-måneders mellomrom. Jo oftere, desto bedre.
4. Forretningssiden og utviklere må arbeide sammen daglig gjennom hele prosjektet
5. Bygg prosjektet rundt motiverte personer. Gi dem miljøet og støtten de trenger, og stol på at de får jobben gjort.
6. Den mest effektive måten å formidle informasjon inn til og innad i et utviklingsteam, er å snakke ansikt til ansikt.

# 12 smidige prinsipper

7. Fungerende programvare er det primære målet på fremdrift/progresjon.
8. Smidige metoder fremmer bærekraftig programvareutvikling. Sponsorere, utviklere, og brukere bør kunne opprettholde et jevnt tempo hele tiden.
9. Kontinuerlig fokus på fremragende teknisk kvalitet og solid design fremmer smidighet.
10. Enkelthet er essensielt.
11. De beste arkitekturer, krav og design vokser frem av selvstyrte team.
12. Med jevne mellomrom reflekterer teamet over hvordan de kan bli mer effektive, og adferden justeres deretter.

# Løsningsforslag 3a)

Gi en kort beskrivelse av rollen Scrum master og beskriv hvilke(n) smidige prinsipper denne rollen understøtte og på hvilken måte den gjør det.

Scrum master

Tilrettelegger for Scrum som utviklingsmetodikk

Arbeidsoppgaver

Gjennomføre daglige standups

Organisere øvrige møter

Planleggingsmøter / Estimeringsmøter / Demoer / Retrospektiv

Opprettholder kundekontakt

Beskytte utviklere fra eksterne forstyrrelser/påvirkning



# Løsningsforslag 3a)

Gi en kort beskrivelse av rollen Scrum master og beskriv hvilke(n) smidige prinsipper denne rollen understøtte og på hvilken måte den gjør det.

Smidige prinsipper som støttes av Scrum master

Individer og interaksjon, fremfor prosesser og verktøy → (5) (6)

Samarbeid med kunden, fremfor kontraktsforhandlinger → (1) (2)

Kvalitetssikring ved å gi gruppen stor frihet → (11)

Refleksjon av arbeidsprosess og progresjon → (12)

# Løsningsforslag 3b)

b) Gi en kort beskrivelse av teknikken Retrospective og redegjør for hvilke smidige prinsipper denne teknikken understøtter.

- Scrum-teamet samles for å diskutere siste sprint → (12)
- Formål: Reflektere over hvordan utviklingen går / identifisere forbedringsområder
  - Hva bør man slutte med? (Stop doing)
  - Hva bør man fortsette med? (Continue doing)
  - Hva bør man starte med? (Start doing)
- Fokus på individer og interaksjoner, fremfor prosesser og planlegging → (5)  
(6)

# Oppgave 4

I smidig utvikling benytter man gjerne brukerhistorier.

a) Hva er en brukerhistorie?

b) Nevn noen fordeler ved å bruke denne teknikken til å beskrive krav.

c) Drøft utfordringer ved å benytte brukerhistorier beskrevet på lapper på en tavle i store, smidige utviklingsprosjekter.

d) Skriv noen brukerhistorier for appen som dere skal utvikle.

# Løsningsforslag oppg.4a

a) Hva er en brukerhistorie?

- Brukerhistorier
  - Beskriver hva brukeren av et system ønsker å få ut av systemet
- Elementer
  - Brukerens rolle
  - Ønsket funksjon
  - Nytteverdi av funksjonen
- Format
  - Som [ROLLE] ønsker jeg [FUNKSJON] for å oppnå [NYTTEVERDI]

# Løsningsforslag oppg.4b

b) Nevn noen fordeler ved å bruke denne teknikken til å beskrive krav.

- Fordeler ved brukerhistorier
  - Krever ingen teknisk kompetanse for å forstå kravet
  - Skjuler kompleksitet
  - Kravene uttrykkes på en kort og konsis måte
- Bruker
  - Hver er kravet tiltenkt?
- Funksjon
  - Hva er den faktiske funksjonen som etterspørres
- Nytteverdi
  - Forstår nødvendigheten av ønsket funksjon

# Løsningsforslag oppg.4c

c) Drøft utfordringer ved å benytte brukerhistorier beskrevet på lapper på en tavle i store, smidige utviklingsprosjekter.

- **Utfordringer i praksis**
  - Kaotisk med mange lapper som hopper seg opp
  - Krever at man er tilstede for å kunne se hva som står på tavlen
- **Generelle utfordringer**
  - Kan tolkes på flere måter → Skjuler kompleksitet
  - Funksjonen kan være vanskeligere å implementere enn antatt → Skjuler underliggende krav
  - Hva er den faktiske betydningen av brukerhistorien?