UiO : Universitetet i Oslo

IN1030 - 20. april 2022

# Prosjektledelse, prosjektplanlegging, teamarbeid

Yngve Lindsjørn, ynglin@ifi.uio.no

# Temaer

- Risikohåndtering
- Ledelse av mennesker og motivasjon
- Teamarbeid
- Prosjektplanlegging
- Smidig planlegging og productbacklog

# Utvikling av programvare - Prosjektstyring

- Suksesskriterier
  - Levere programvaren til rett tid
  - Holde kostnadene innen budsjett
  - Levere programvare som møter kundens forventninger og krav
  - Opprettholde et velfungerende utviklingsteam

# Risikohåndtering

- En risiko er en sannsynlighet for at uønskede omstendigheter skjer
- Tre hovedtyper av risiko:
  - **Prosjekt-risikoer** vil ha effekt på tidsplanen og/eller ressurser
  - **Produkt-risikoer** vil ha effekt på kvaliteten eller av programvaren som utvikles
  - **Forretnings (Business)-risikoer** vil ha effekt på organisasjonen som utvikler eller eier programvaren

# Ulike typer av risiko – fra lærebok (kapittel 22.1.1)

| Risiko type | Mulige risikoer |
|---|---|
| Teknologi | Databasen som brukes i systemet klarer ikke å prosessere så mange transaksjoner per sekund som forventet |
| Mennesker | Umulig å rekruttere mennesker med den kompetanse som kreves<br>Nøkkel personell ikke tilgjengelig i kritiske faser |
| Organisasjon | Organisasjonen blir restrukturert slik at ulik ledelse er ansvarlig for prosjektet |
| Verktøy | Ulike programvareverktøy lar seg ikke integrere |
| Krav | Endringer av krav krever omfattende "redesign” |
| Estimering | Underestimering (i tid) av programvareutviklingen<br>Tiden det tar å rette feil er underestimert |

Hvilke hovedtyper (se forrige slide) av risiko inngår disse?

# Risikoanalyse

- Vurder sannsynlighet og mulig konsekvens for hver risiko
- Sannsynlighet kan være svært lav, lav, moderat, høy eller svært høy
- Konsekvensen kan være katastrofal, alvorlig, mindre alvorlig eller ubetydelig

# Risikotyper og eksempler fra lærebok (kapittel 22.1.2)

| Risiko | Sannsynlighet | Konsekvens |
|---|---|---|
| Det er umulig å rekruttere medarbeidere med kompetansen som er nødvendig | Høy | Katastrofal |
| Nøkkel personell er syke eller fraværende i kritiske faser av prosjektet | Moderat | Alvorlig |
| Det er foreslått endringer i kravspesifikasjonen som vil kreve store endringer i design av systemet | Moderat | Alvorlig |
| Organisasjonen restruktureres slik at ulik ledelse har ansvar for prosjektet | Høy | Alvorlig |
| Databasesystemet kan ikke prosessere antall transaksjoner per sekund som forventet | Moderat | Alvorlig |

# Ledelse av mennesker

- Mennesker er organisasjonens største ressurs
- Oppgavene til en leder er hovedsakelig orientert mot andre mennesker
- Mangelfull eller dårlig ledelse er ofte en av de viktigste årsakene til at et prosjekt feiler

# Motivasjon

- Er ikke folk motiverte, er de lite interesserte i hva de gjør
- Motivasjon er komplekst, men ulike typer motivasjon er basert på
  - Basis behov (mat, søvn, etc.)
  - Sosiale behov (å bli akseptert i en gruppe, etc.)
  - Personlige behov (respekt, selvtillit , etc.)

# Personlighetstyper

- Oppgaveorientert (Task-oriented)
  - Motivasjonen for å gjøre arbeidet er oppgavene i seg selv
- Selvorientert (Self-oriented)
  - Arbeidet er et middel for å oppnå individuelle mål – bli rik, få posisjon, reise, etc.
- Samspillorientert (Interaction-oriented)
  - Motivasjonen er først og fremst å samarbeide og ha det bra med medarbeiderne. Man går på jobb fordi man liker å gå på jobb

# Teamarbeid

- Et team som fungerer er samlet og har en god teamfølelse. Teamets mål viktigere enn egne mål
- Kommunikasjon er en nøkkelfaktor for å lykkes
- Fleksibilitet i teamsammensetning er ofte begrenset av hvem som er tilgjengelig

## Hva er spesielt med Software-team?

- Hyppige endringer gjør det vanskelig å planlegge
- Komplekse sosiale og tekniske system
- Få etablerte teorier om systemutvikling

# Hvem sa dette?

- Det viktigste i teorien er å gjøre andre bedre
- Du er ikke dyktig hvis du ikke kan bruke dyktigheten din til å gjøre andre bedre

# Fotballtreneren Nils Arne Eggen

- Han skrev også (i boka *godfoten*):
  - Det er laget og kollektivet som er viktig, ikke den enkelte spiller eller trener
  - Hele samhandlingsprosessen handler om å utvikle enkeltindividet, slik at vedkommende gjør kollektivet enda bedre
  - En fotballspiller er ikke et produkt av seg selv, men et produkt av laget

Dette er tanker om samhandling og kollektivet i fotballag, men kan også brukes om team innen systemutvikling.

# Effektivitet i team

- Teammedlemmer
  - Trenger en blanding av folk fordi systemutvikling involverer ulike aktiviteter som programmering, testing, programvarearkitektur og dokumentasjon
- Teamorganisering
  - Et team bør organiseres slik at alle teammedlemmene kan bidra best mulig og at oppgaver blir utført som forventet
- Kommunikasjon
  - God kommunikasjon mellom teammedlemmene, og mellom teamet og andre interessenter (stakeholders), er helt essensielt

# Fasilitering av team arbeidet er viktig

- Viktig at teamet diskuterer hvordan en oppgave skal løses sammen.
- Eksempel: analyse av et intervju, alle bør få anledning til å høre på intervjuet før analyse. Diskutere og bli enige om konklusjonen, selv om det kun er en som skriver selve analysen.
- Vanskelig å skrive oppgave sammen. Noen vil at det skal være bedre, noen er flinkere til å skrive enn andre.
  - Må gi slipp på det perfekte.
- Fordeling av oppgaver er en av de største utfordringene.
- Diskuter opplegget, hva er målet etc. før fordeling.

# Teammedlemmer som ikke bidrar

- Hva skal vi gjøre når X aldri stiller opp på møtene?
- X er sjelden med i undervisningen og det virker som det er helt andre ting enn prosjektarbeidet som blir prioritert.
  - Skal vi ha med navnet på prosjektrapporten?

DID NOT WORK WELL IN GROUPS
SIZEMORE

# Forskningsprosjekt innen teamarbeid

- Hva er effektivt teamarbeid i “smidige” prosjekter?
  - Måle effektiviteten innad i team og på tvers av team
  - Distribuerte team
- Noen dimensjoner ved teamarbeid som vi undersøker
  - Selvstyre i team
  - Kommunikasjon i team
  - Teamfølelse
  - Læring i team
  - Gjensidig support og oppgavefordeling
  - Koordinering

# Teameffektivitets-modell i studien (TWQ – Teamwork Quality)

**Kvaliteter i teamarbeid**
- Kommunikasjon
- Koordinering
- Bidragsbalanse
- Gjensidig support
- Innsats
- Teamfølelse

36 spørsmål

**Team Effektivitet**
- Produktkvalitet
- Tid og budsjett

15 spørsmål

**Personlig suksess**
- Jobbtilfredshet
- Læring

10 spørsmål

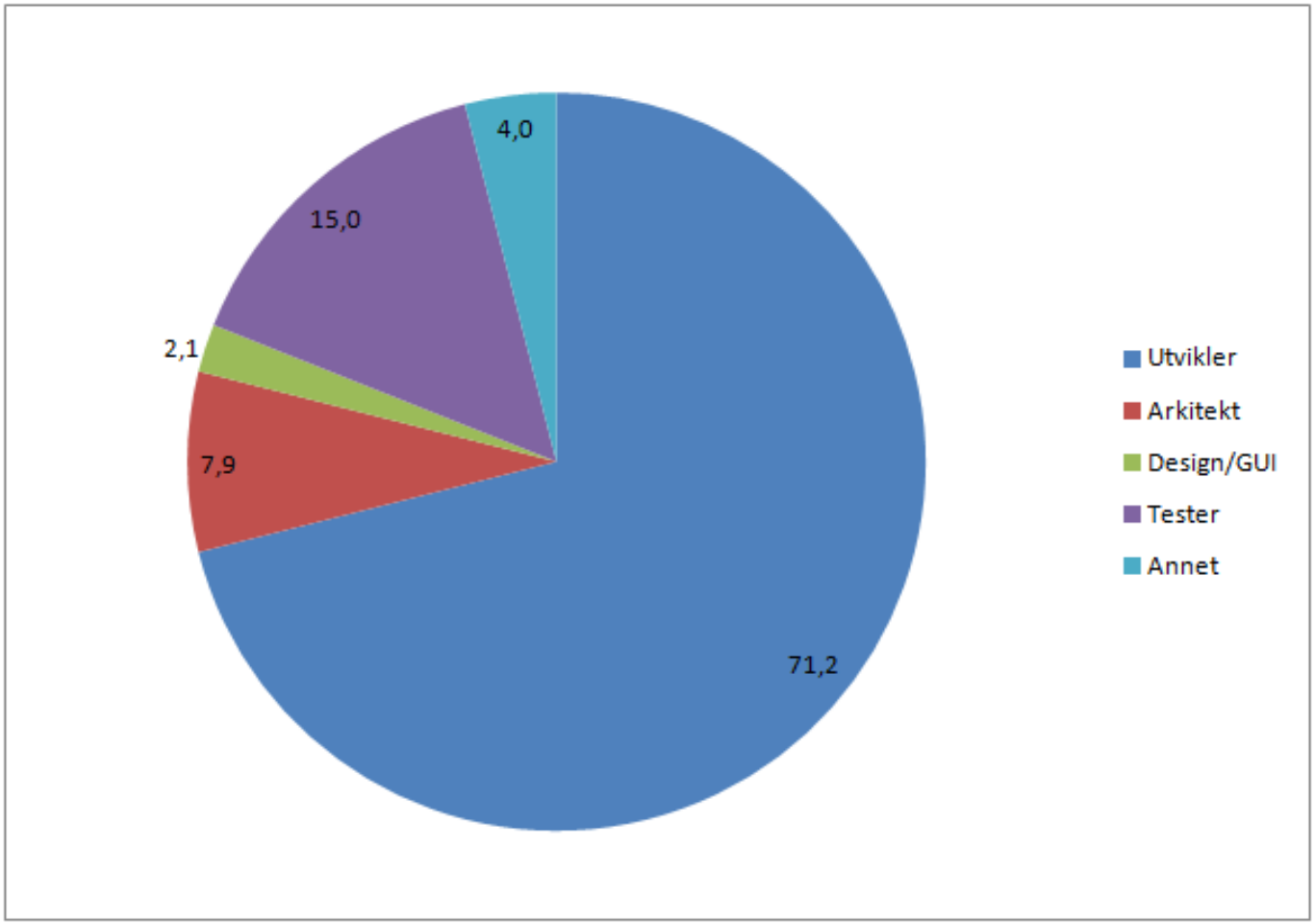
# Eksempler spørsmål

| Nr | Spørsmål Teamarbeid | Sterkt uenig | Uenig | Verken enig eller uenig | Enig | Sterkt enig |
|---|---|---|---|---|---|---|
| 1 | Det er hyppig kommunikasjon innad i teamet | | | | | |
| 2 | Teammedlemmene kommuniserer hyppig i spontane møter, telefon etc. | | | | | |
| 3 | Teammedlemmene kommuniserer for det meste direkte og personlig med hverandre | | | | | |
| 4 | Mye av kommunikasjonen i teamet går gjennom sentrale personer (teamleder eller visse teammedlemmer) | | | | | |
| 5 | Relevante ideer og informasjon om teamarbeidet deles åpent blant alle teammedlemmene | | | | | |
| 6 | I noen situasjoner blir viktig, relevant informasjon holdt tilbake fra andre teammedlemmer | | | | | |

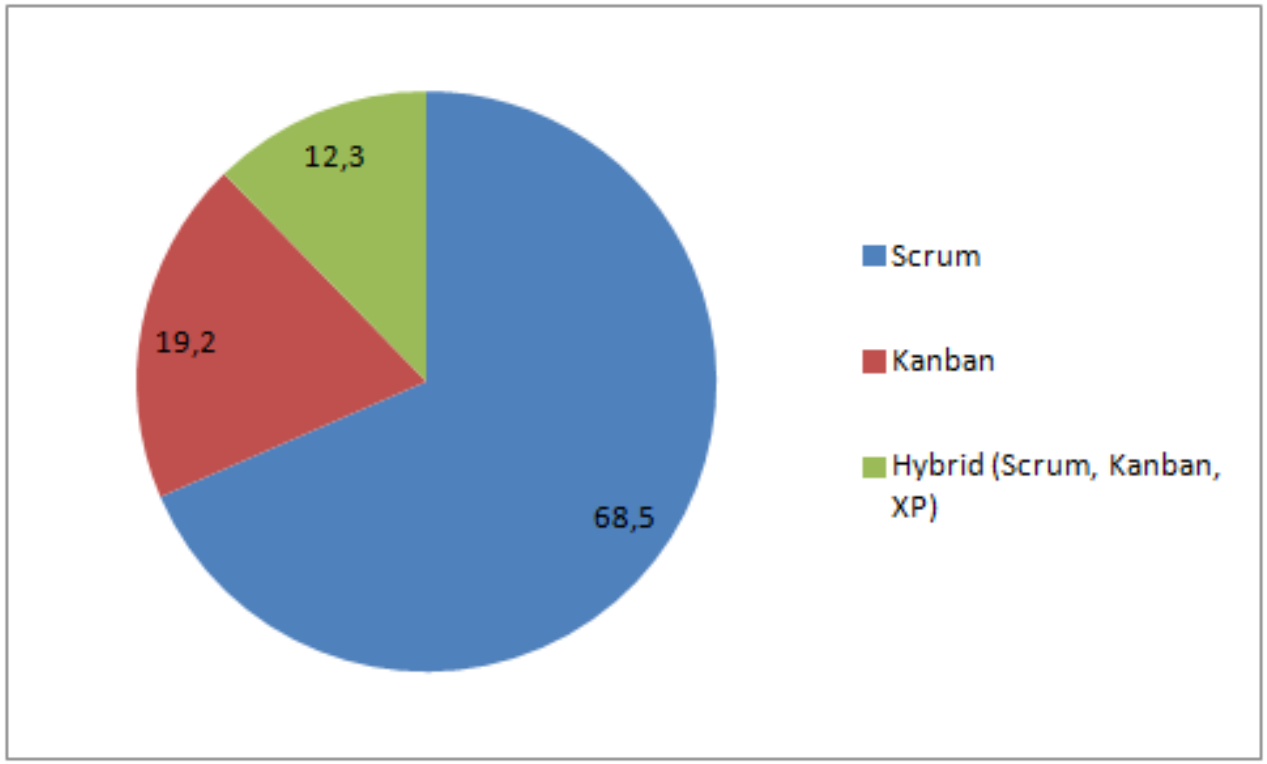
| | | | | | | |
|---|---|---|---|---|---|---|
| 39 | Så langt kan teamet være fornøyd med seg selv | | | | | |
| 40 | Teammedlemmene drar nytte av team-samarbeidet | | | | | |
| 41 | Teammedlemmene vil gjerne gjøre lignende samarbeid igjen | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 47 | Ut fra resultatene kan teamarbeidet betraktes som en suksess | | | | | |
| 48 | Alle krav fra kundene blir tilfredsstilt | | | | | |
| 49 | Sett fra bedriftens perspektiv oppnår teamet sine mål | | | | | |

# Teammedlemmer – primærrolle i team



# Type ”smidig” team



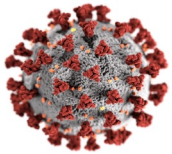
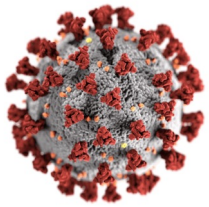
# Noen funn i undersøkelsen

- Fant en sammenheng mellom kvalitet på teamarbeid og kvalitet på prsosjektet (både produkt og prosjektkvalitet)
- Liten konsensus mellom de ulike raterne (teammedlem, teamleder og produkteier) når det gjelder team effektivitet
  - Ganske like resultater sammenlignet med studie av tradisjonelle (ikke smidige) team
- Kvaliteten på teamarbeidet (kommunikasjon, koordinering etc.) har  størst betydning på produktkvalitet, mindre på prosjektkvalitet (tid og kostnad)

# Teamarbeid i systemutvikling - studentteam

- IN2000: Stort prosjektkurs innen systemutvikling (20 stp.) ved IFI, 4 semester
- Svært relevant for arbeidslivet - smidig metodikk i praksis
- Teamarbeid sentralt
- Utvikle en app på Android plattform
  - Velge mellom ulike «vær case» med API (data) fra meteorologisk institutt
- Omfattende rapport
- Vurdering: 50 % gruppe (team) (applikasjon og rapport), 50 % individuelt (individuell eksamen). I tillegg tre obligatoriske oppgaver
- Utfordringer: teamarbeid, veiledning, teknologi og verktøy, undervisning og gruppeundervisning

# Teamarbeid over fem år – IN2000

| 2019 | 2020 | 2021 | 2022 | 2023 |
|---|---|---|---|---|
| 39 teams | 42 teams +  | 48 teams  | 39 teams | 40 teams |

# Sette sammen team

- Studentene melder inn i nettskjema
  - Mulighet til velge 1 eller 2 andre medstudenter, studieprogram og ambisjonsnivå og eventuelle kommentarer
- Kursledelsen setter sammen teamene (i utgangspunktet 6 studenter i hvert team) basert på input i nettskjema
  - Tar hensyn til kjønn og studieprogram etc. så godt det lar seg gjøre

# Samarbeidsverktøy studentene brukte i teamene

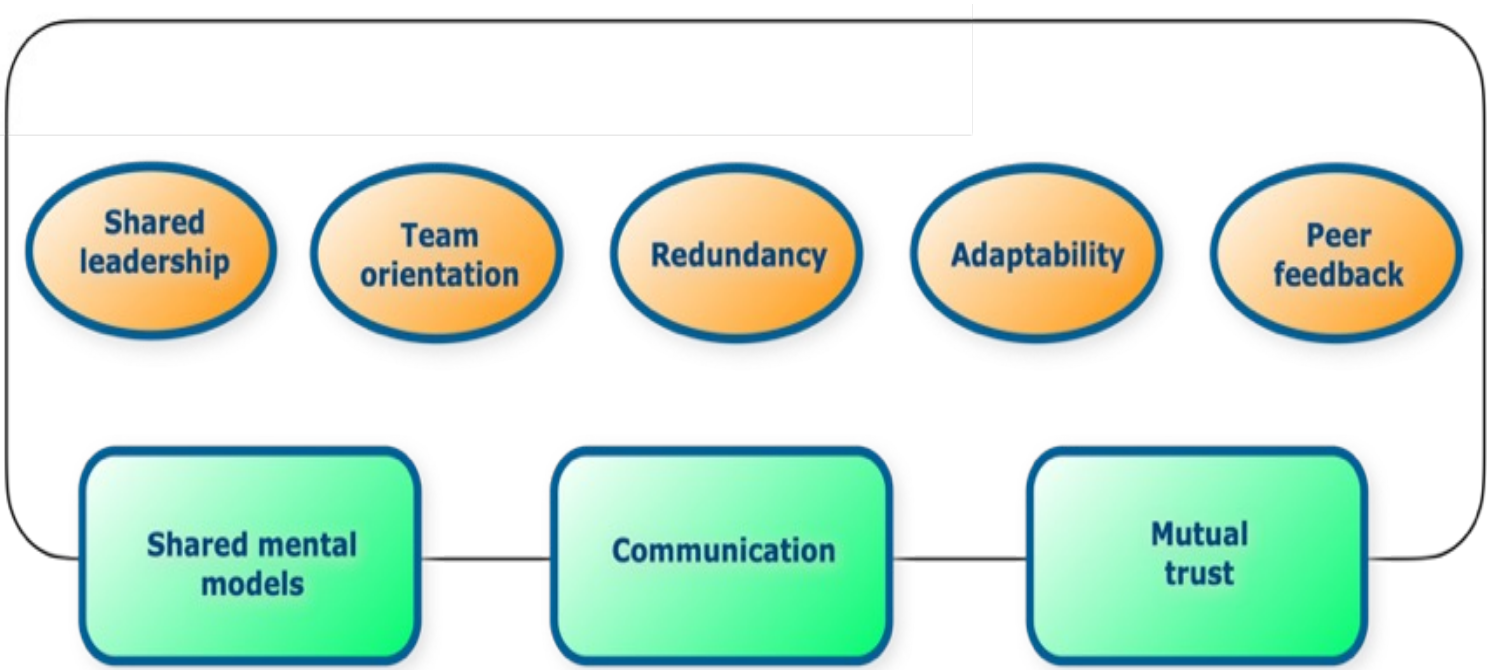
# Sammenligning 2019, 2020 og 2021

| | | | | 2019 | | 2020 | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|
| **Construct** | **Rater** | **Variable** | **No. items** | **Mean** | **SD** | **Mean** | **SD** | **Mean** | **SD** |
| Team Quality (TWQ) | Team member | Communication | 10 | 4,17 | 0,37 | 3,99 | 0,51 | 4,15 | 0,35 |
| | | Coordination | 4 | 4,05 | 0,40 | 3,98 | 0,45 | 4,14 | 0,33 |
| | | Mutual support | 7 | 4,42 | 0,37 | 4,33 | 0,47 | 4,50 | 0,33 |
| | | Cohesion | 10 | 4,26 | 0,45 | 4,20 | 0,45 | 4,37 | 0,37 |
| | | Effort | 4 | 3,86 | 0,65 | 3,73 | 0,62 | 3,97 | 0,53 |
| | | Balance of contribution | 3 | 4,25 | 0,41 | 4,24 | 0,50 | 4,32 | 0,46 |
| Team members' success | Team member | Work satisfaction | 4 | 4,28 | 0,44 | 4,37 | 0,40 | 4,41 | 0,43 |
| | | Learning | 4 | 4,41 | 0,47 | 4,42 | 0,45 | 4,55 | 0,31 |
| Team performance | Team member | Effectiveness_TM | 10 | 3,86 | 0,42 | 4,03 | 0,34 | 4,07 | 0,42 |
| | | Efficiency_TM | 5 | 3,81 | 0,60 | 3,98 | 0,55 | 4,07 | 0,51 |
| | | | | | | | 0,54 | 4,15 | 0,58 |
| | | | | | | | 0,75 | 4,12 | 0,72 |
| | | | | | | | 0,56 | 4,11 | 0,63 |
| | | | | | | | 0,73 | 4,12 | 0,72 |

28

På en skala fra 1 til 5 (svært uenig til svært enig) er gjennomsnittet relativt likt for 2019, 2020 og 2021 for alle faktorer.

# Revidert «Big five» model – Agile Teamwork Effective Model«ATEM»




https://www.cutter.com/article/right-thoughts-right-action-how-make-agile-teamwork-effective

# Coordination mechanisms with behavioural markers in ATEM model

| Teamwork coordination mechanism | Behavioural markers |
|---|---|
| **Shared mental models** | • Anticipating and predicting each other's needs.<br>• Common understanding of work process.<br>• Common understanding of tasks.<br>• Common understanding of individual skills. |
| **Communication** | • The team follows up on progress of tasks.<br>• Visualising project information.<br>• Facilitate informal communication. |
| **Mutual trust** | • Information sharing.<br>• Willingness to admit mistakes and accept feedback.<br>• Supportive environment. |

# Teamwork components with behavioural markers, ATEM model

| Teamwork component | Behavioural markers |
|---|---|
| **Shared leadership** | • The agile team facilitates team problem solving.<br>• The agile team synchronises and combines individual team member contributions using agile practices combined with automated tools.<br>• The agile team seek and evaluate information that affects team functioning.<br>• Agile values and methodologies determine team member roles.<br>• Agile values and methodologies determine the frequency and type of preparatory meetings and feedback sessions.<br>• A servant leader facilitates a boundary-spanning function. |
| **Team orientation** | • Taking into account alternative solutions provided by teammates and appraising that input to determine what is most correct.<br>• Increased task involvement, information sharing, strategizing, and participatory goal setting.<br>• The team "sticks together and remains united”. |
| **Redundancy** | • Recognition by potential backup providers that there is a workload distribution problem in their team.<br>• Shifting of work responsibilities to underutilized team members.<br>• Completion of the whole task or parts of tasks by other team members. |
| **Adaptability** | • Identify cues that a change has occurred, assign meaning to that change, and develop a new plan to deal with the changes.<br>• Identify opportunities for improvement and innovation for habitual or routine practices.<br>• Remain vigilant to changes in the internal and external environment of the team. |
| **Peer feedback** | • Identifying mistakes and lapses in other team members’ actions.<br>• Regular feedback regarding team member actions to facilitate self-correction. |

# Plandrevet utvikling

- Utviklingsprosessen planlegges i detalj
  - Basert på  tradisjonelle prosjektstyringsteknikker og har vært den mest vanlige måten å styre store programvareprosjekter på
- Argumenter for en plandrevet tilnærming er at tidlig planlegging sørger for at organisatoriske forhold, som tilgjengelige ressurser blir tatt hensyn til, og at potensielle problemer og avhengigheter blir oppdaget før prosjektet starter i stedet for midt i prosjektet
- Hovedargumentet mot plandrevet utvikling er at mange tidlige avgjørelser likevel må endres på grunn av endringer i omgivelsene der programvaren utvikles og skal brukes

# Prosjektplan

- I et plandrevet utviklingsprosjekt vil en prosjektplan blant annet si noe om tilgjengelige ressurser i prosjektet, arbeidsfordelingen blant ressursene og en tidsplan for når arbeidet skal gjøres.
- Planen inneholder gjerne følgende:
  - Innledning
  - Prosjektorganisering
  - Risikoanalyse
  - Maskinvare- og programvarekrav (standard programvare som skal brukes)
  - Arbeidsfordeling
  - Tidsplan for prosjektet
  - Rapporterings- og overvåkingsmekanismer

# Smidig planlegging

- Smidige metoder i programvareutvikling er en iterativ tilnærming der programvaren blir utviklet og levert til kundene som ”tillegg” (”increments”)
- Til forskjell for plandrevet tilnærming, er funksjonaliteten til tilleggene (increments) ikke planlagt på forhånd men avgjøres under utviklingen
  - Hva som tas med i en iterasjon avhenger av utvikling i prosjektet og kundens prioriteringer
- Kundens prioriteringer og krav endrer seg. Derfor kan det være fornuftig å ha en fleksibel plan som kan ta høyde for disse endringene

# Historiebasert (”Story-based”) planlegging

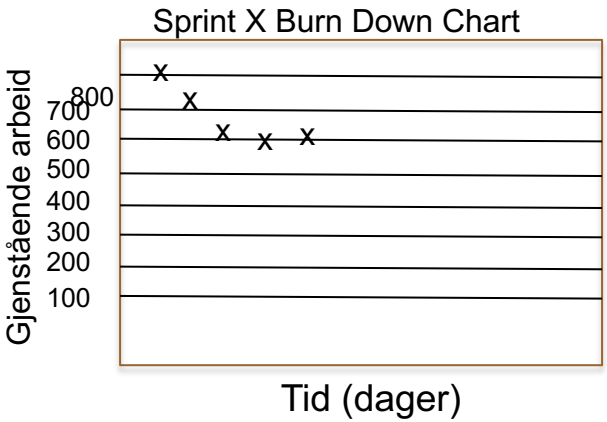
- System spesifikasjonen i XP (og SCRUM) er basert på brukerhistorier (user stories) som reflekterer egenskapene i systemet
  - ”Som student, ønsker jeg å melde meg opp i kurs”
- Teamet diskuterer historier og rangerer dem i forhold til tiden de tror det tar
- Historier som skal være med i en ”iterajon” velges, der antall historier reflekterer tiden det tar å levere en iterajon (typisk 2-4 uker)

Les mer om user stories og forskjellen til use cases (sentralt i UML) på http://en.wikipedia.org/wiki/User_story

# SCRUM - prosess og roller

## Daily Standup – 3 spørsmål

- Hva har du gjort siden i går?
- Hva planegger du å gjøre til i morgen?
- Hvilke eventuelle hindringer har du?




## Scrum Master

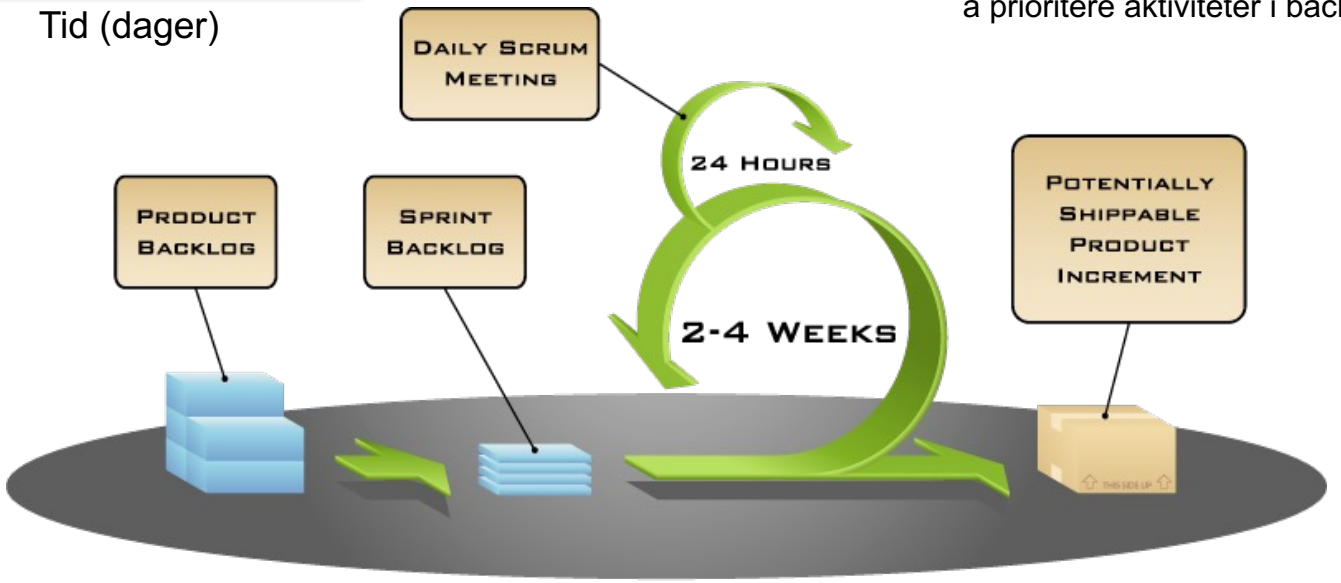
- Sørger for at Scrum prosessen blir fulgt
- Leder "daily stand up"
- Hindrer støy slik at utviklerne kan fokusere på oppgavene
- Teamets "coach og beste venn"
- Koordinerer SPRINT planlegging, og estimering av Sprint backlog

## Scrum Team

- Typisk 5-9 personer
- Utviklere, testere, arkitekt,…
  - Jobber fulltid
- Selvstyrt
- Utvikler systemet

## Product Owner

- Representerer kunden
- Setter opp mål for hver SPRINT
- Ansvar for product backlog, og for å prioritere aktiviteter i backlog




Copyright © 2005, Mountain Goat Software

# Product Backlog (produktkø)

| | | | Nye estimater i Sprint | | | | |
|---|---|---|---|---|---|---|---|
| **Prioritet** | **Brukerhistorie** | **Estimat (Initielt)** | 1 | 2 | 3 | 4 | |
| 1 | Som en bruker ønsker jeg å få en oversikt over luftkvalitet i Oslo en bestemt dag | 8 | | | | | |
| …. | …. | | | | | | |
| | Restrukturer koden for å ta i bruk MVC pattern | 13 | | | | | |

# Product Backlog (produktkø)

- Ulike “items”
  - Kunde spesifikke (Få en oversikt over luftkvaliteten i byer)
  - Forbedringsmål (Restrukturere kode)

- “Items” i produktkøen kan utrykkes som
  - User stories
  - Use Cases
  - Andre måter å spesifisere krav på som passer

# Sprint planlegging

En måte å lage sprint backlog

| | | | | Nye estimater etter dag | | | |
|---|---|---|---|---|---|---|---|
| **Brukerhistorie** | **Oppgave i Sprint** | **Estimat (Initielt)** | **Team medlem** | 1 | 2 | 3 | 4 |
| Som en bruker ønsker jeg å få en oversikt over luftkvalitet i Oslo en bestemt dag | Finne riktig API og tolke data | 2 | **Syver** | | | | |
| | Lage datastruktur - klasser | 5 | **Anne** | | | | |
| | Vise kart med fargekode | 8 | | | | | |
| | Gi tekstlig beskrivelse i kartet | 5 | | | | | |
| Restrukturer koden for å ta i bruk MVC pattern | | | | | | | |