Risikostyring og programvare utvikling i en smidig verden

Prosess for risikostyring og programvare utvikling ved stadige hurtigere endringer i teknologi og kortere tid til å gjennomføre prosjekter

ESRA årsmøte
Oslo, Torsdag1 Juni 2017: 13^{50}– 14^{30}

ESRA Norge
Norsk forening for risiko- og pålitelighetsanalyse

Thor Myklebust, SINTEF Digital
Tor Stålhane, IDI/NTNU
Geir K. Hanssen, SINTEF Digital
Ingar Kulbrandstad, Autronica Fire and Security

Tema

- Introduksjon
  - Terminologi
  - Smidig utvikling
  - SafeScrum
- Programvare
- Smidig risikoledelse
- Smidig og tradisjonell risikoledelse
- Sikkerhets-standarder og smidig utvikling
- Case
- Konklusjon

Trender

- Trend: Nye løsninger blir sikkerhets-systemer (IoT, trådløs)
- Trend: Information security = safety
- Trend: Innovasjon og utvikling går fra HW til SW
- Trend: Graden av endring øker
- Sum: SW- og prosjekt kompleksitet øker.

0 kodelinjer
(liten endring)

25 million kodelinjer
(medium endring)

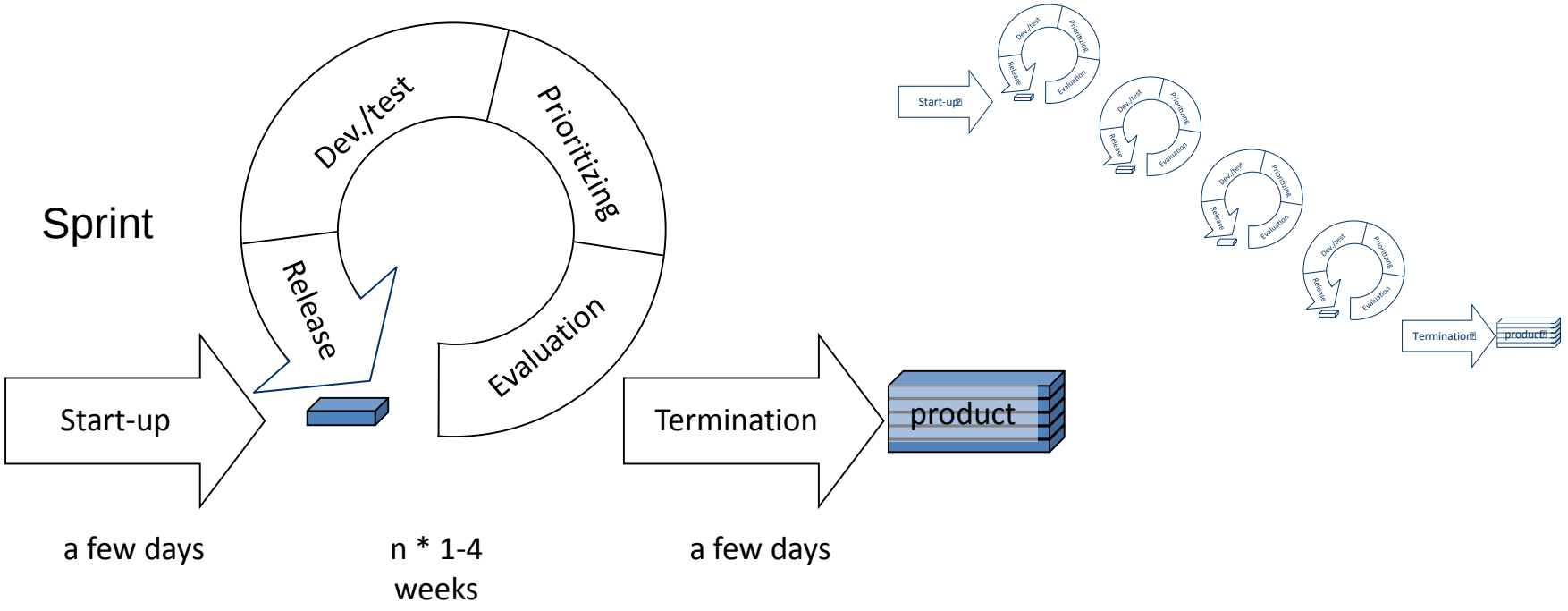
- 100 millioner kodelinjer
- (endres ofte)

Manifestet for smidig programvareutvikling

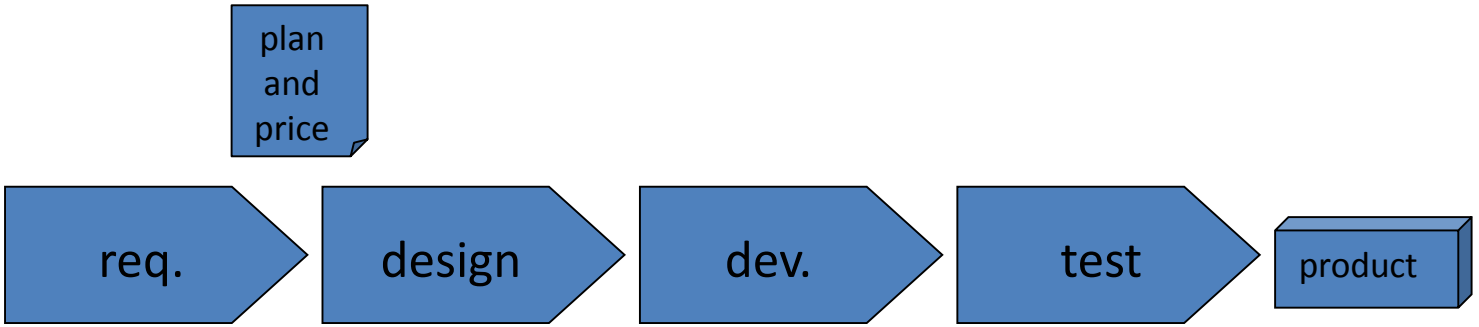
- **Personer og samspill** fremfor prosesser og verktøy
- **Programvare som virker** fremfor omfattende dokumentasjon
- **Samarbeid med kunden** fremfor kontraktsforhandlinger
- **Å reagere på endringer** fremfor å følge en plan

Kilde: http://agilemanifesto.org/iso/no/manifesto.html

Smidig livs-syklus. Inkrementell



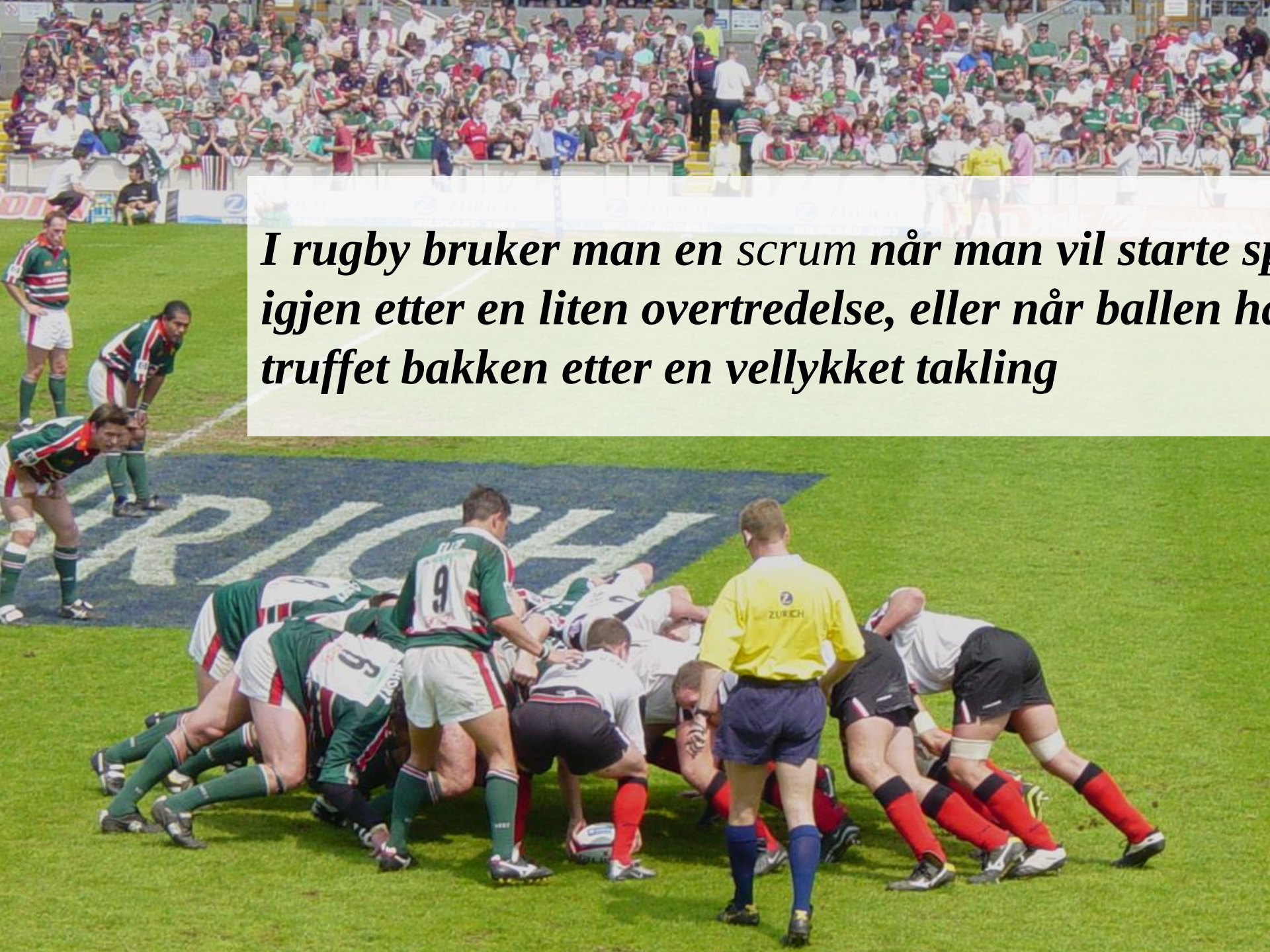
Plan-basert



Hvorfor smidig?

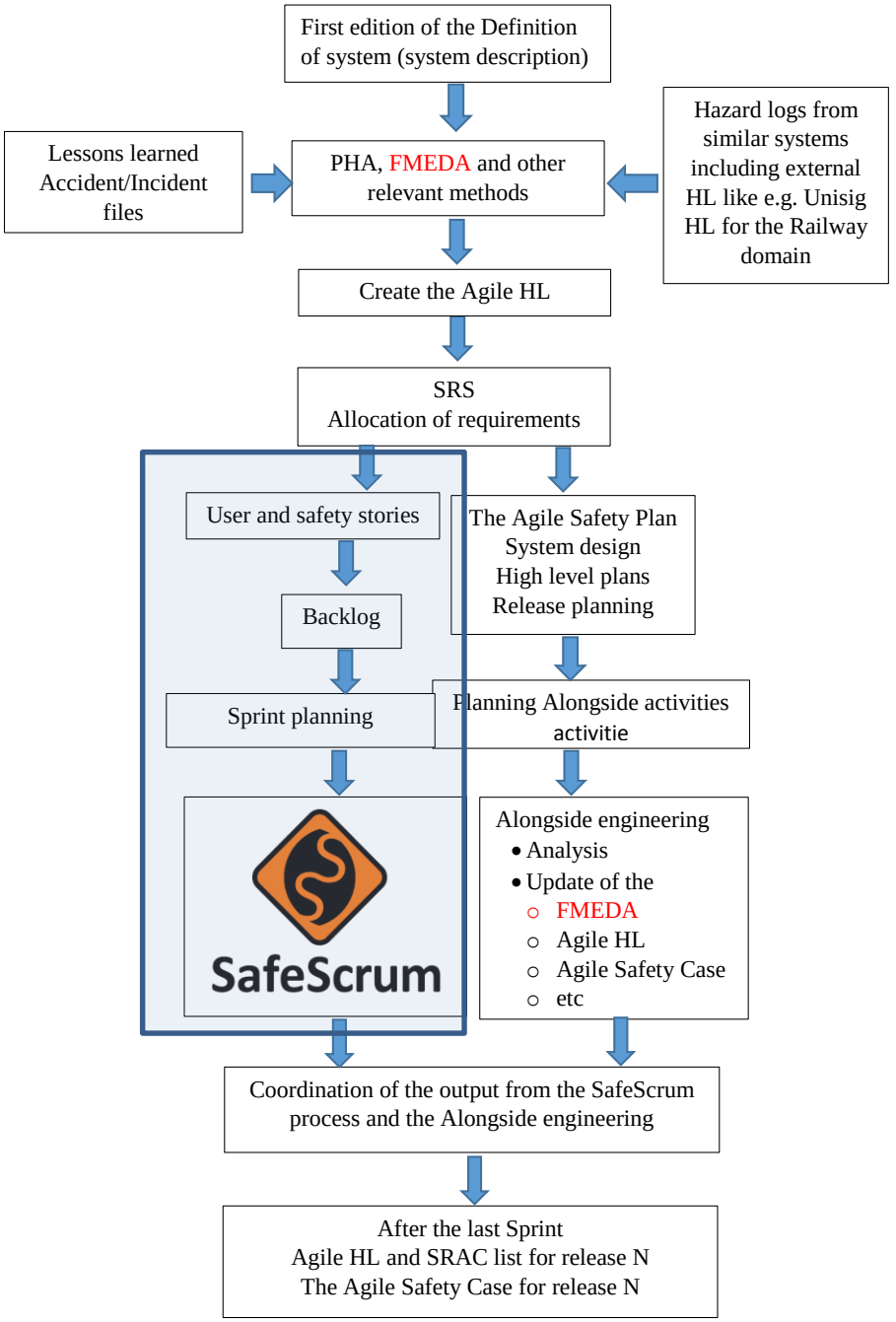
- **Raskere**
  - Økt etterspørsel etter raskere utvikling og leveranser
  - Tidlig utgivelse av programvare
- **Bedre**
  - Bedre forståelse av krav pga bedre mekanismer for feedback
  - Bedre design
  - Læring er integrert i prosessen
- **Billigere**
  - Lavere dokumentasjons-kostnader
  - Pragmatisk administrasjon
  - Mindre risiko for at det blir forsinkelser

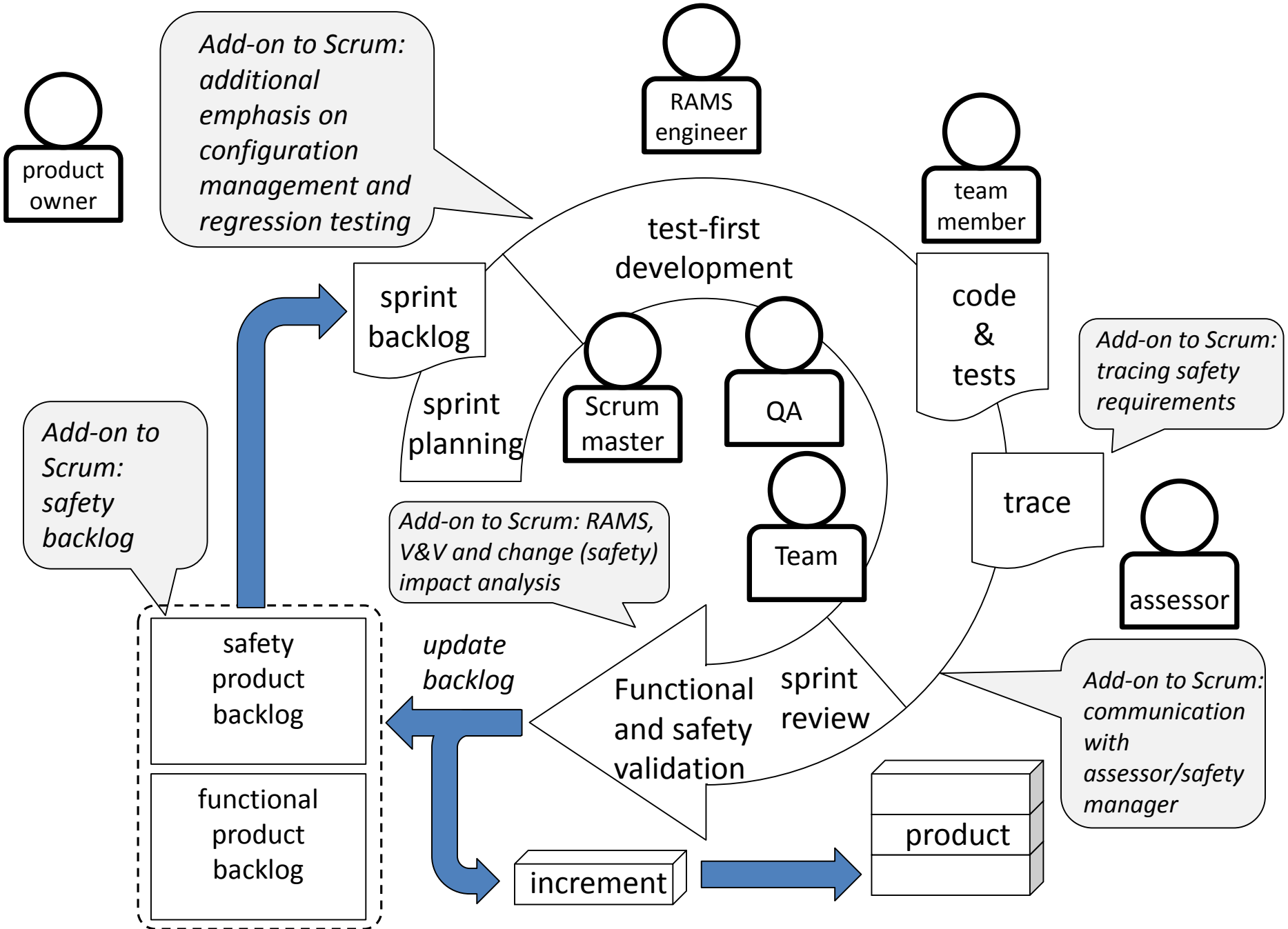
ISO 9001 assessor erfaring:
Flere firma er flinke til å lære av sine feil fordi de er så flinke til å gjenta dem!



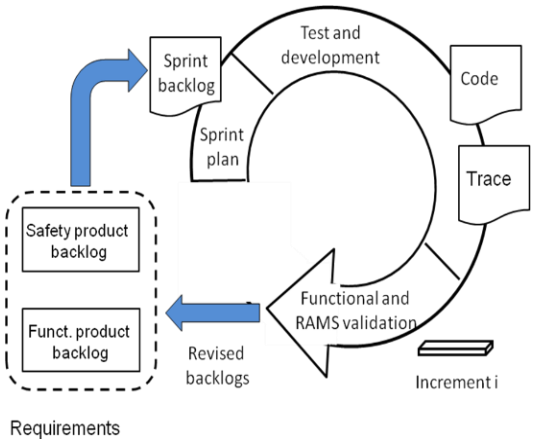
I rugby bruker man en** scrum **når man vil starte sp
igjen etter en liten overtredelse, eller når ballen h
truffet bakken etter en vellykket takling

Smidig livs-syklus




product owner
Add-on to Scrum: additional emphasis on configuration management and regression testing
RAMS engineer
team member
test-first development
code & tests
sprint backlog
sprint planning
Scrum master
QA
Team
Add-on to Scrum: tracing safety requirements
trace
assessor
Add-on to Scrum: safety backlog
Add-on to Scrum: RAMS, V&V and change (safety) impact analysis
update backlog
safety product backlog
functional product backlog
Functional and safety validation
sprint review
Add-on to Scrum: communication with assessor/safety manager
increment
product

Fra siste Sprint til aktuelt Tilsyn



Utviklet produkt

Uavhengige tester

SC forfatter

Analyser Inspeksjoner

TASC

SAR

Kommunikasjon mellom SC forfatter og assessor

Domene avhengig

SC: Safety Case
TASC: The Agile Safety Case
SAR: Safety assessment report

statens jernbanetilsyn

Tradisjonell og smidig risikostyring
Sikkerhets-systemer

Tradisjonell	Smidig
Tidlige risikoanalyser	Smidig er noe umoden på dette området. Vil utføre tidlige analyser med vekt på at det er lett å oppdatere dem
Utføres ofte uten programvare eksperter	Inkluder programvare eksperter
Omfattende risikovurdering før store innvesteringer	Starter prosjekter uten for omfattende risikovurderinger. Mange del-leveranser, slik at deler brukes tidlig. Deretter utvider man systemet

Sammen er vi gode

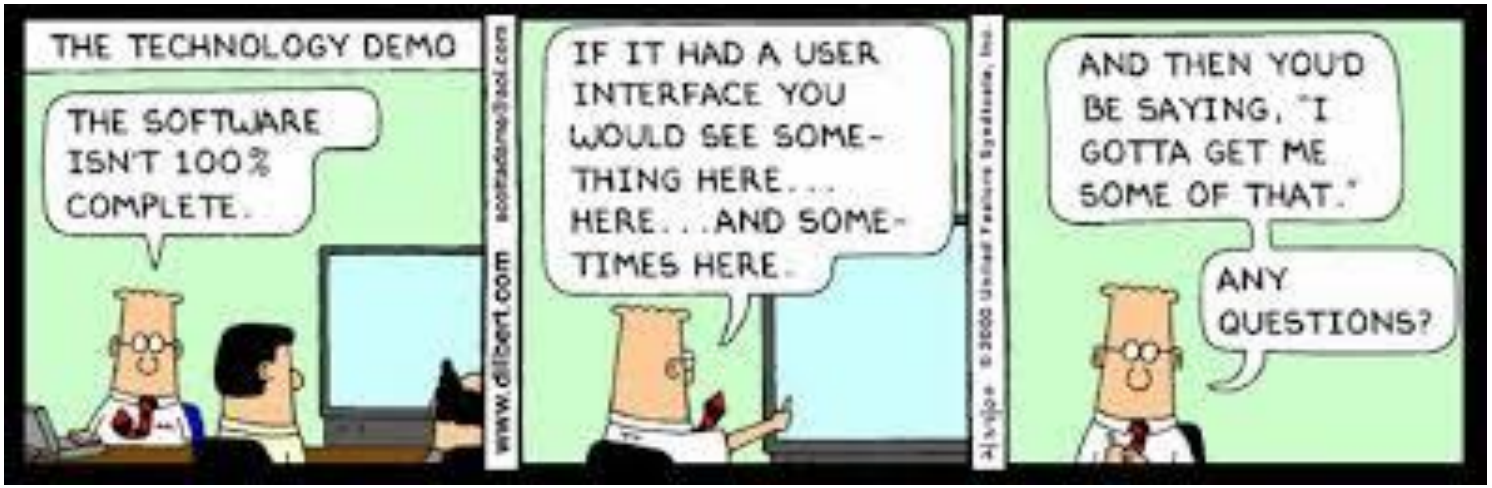
Tradisjonell	Smidig
Formelle krav	Smidig estimering
Detaljerte estimat av oppgaver	Burn down diagram
En leveranse	Inkrementell leveranse
Prosjekt plan	Kundeinvolvering
Formell risikoledelse	Retrospektiv

Ofte operer prosjekter midt i mellom tradisjonell og smidig

Smidige risiko-prinsipp

Transparent:

- Fokus på å vise frem resultater tidlig
- Har inkludert fremvisning av produkt i prosessen
- Egne praksiser for blant annet:
  - Sprint demo



Smidige risiko-prinsipp

Samarbeid rundt planlegging:

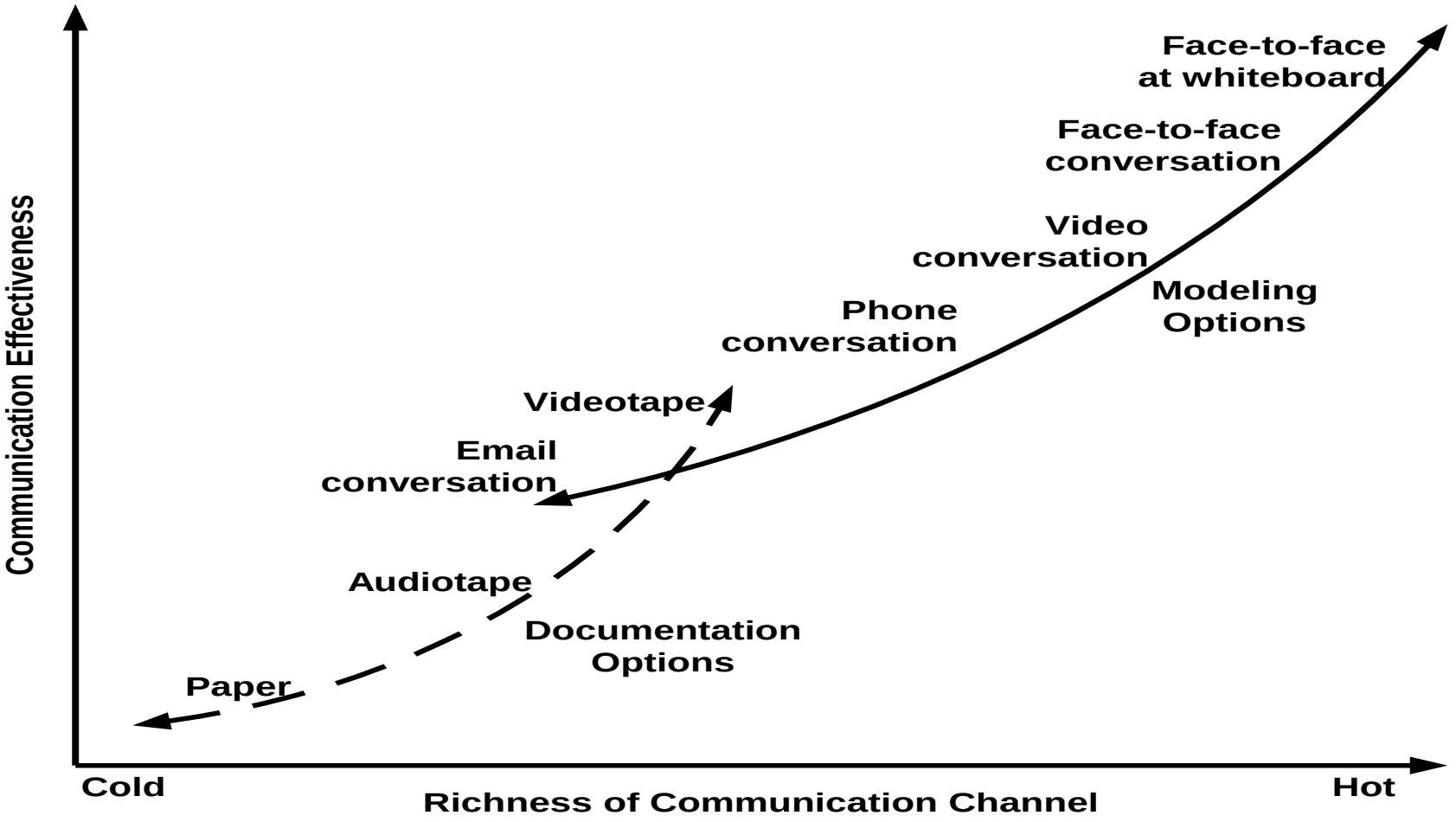
- Utnytte kompetansen og erfaringen til hele gruppen
- Viktig for å inkludere alle risiko-elementene
- Praksiser hvor dette kan inkluderes i tillegg til vanlige risikomøter
  - Sprint planlegging
  - Sprint review

Smidige risiko-prinsipp

Involvering av kundene:

- Redusere kunderisiko ved at de involveres i hele livssyklus
- Hvis kundene ikke kan eller bør delta så inkluder en intern senior som kan representere kunden
- Praksiser hvor dette kan inkluderes i tillegg til vanlige risikomøter
  - Sprint Demo

Communication – a critical component



Ref. Scott Ambler

Andre relevante praksiser

Daglig scrum

Inkluderer 3+1 spørsmål

1. Hva ble du ferdig med i går?
2. Hva har du planlagt å gjøre i dag?
3. Noen problemer eller utfordringer?

4. Risiko utfordringer?

Sikkerhets-standarder og smidig utvikling

Standarder hvor smidig kan brukes

Har sjekket disse ift SafeScrum

1. IEC 61508 generisk
2. IEC 61511 prosess industri
3. EN 50128 Jernbane
4. IEC 60880 Kjernekraft
5. ISO 26262 Bilindustrien
6. IEC 62304 Programvare til medisinsk utstyr
7. DO 178C Flyindustrien

IEC 61508-3 Programvare møter

Moderne SW utviklingsprosesser

- SafeScrum som eksempel
- Definisjoner
- Livssyklus
  - Inkrementell utvikling
- Konfigurasjonsstyring
- Endringsanalyse
- Dokumentasjon
- Regresjonstester

SafeScrum:
Sterk motstand på første møte!

Mindre motstand på det andre møtet

Tredje møte: Ingen motsa oss

Fjerde møte: Akseptert?

.

.

7de møte: Klargjorde inkrementell utvikling.i Del7: **Akseptert**!

Case: Autronica

Erfaring med Scrum siden 2011

- Noen personer har erfaring med Scrum fra andre bedrifter fra 2003 og 2007

Erfaring med SafeScrum siden 2014

Benyttet SafeScrum ifm 2 sertifiseringer

- SafeScrum er brukt i forbindelse med AutroSafe med SIL 2.
- SafeScurm er brukt i forbindelse med AutroMaster som har SOLAS (Safety of Life at Sea) sertifisering.

Begynt å bruke SafeScrum i forbindelse med et hardware prosjekt som skal lede til SIL 3 produkter.

Case: Autronica

Benytter Scrum ifm utvikling av

- SIL2 systemer og
- "ikke-SIL" system: Autromaster presentasjons-system
- i start gropa ved bruk av Scrum i hardware prosjekter

Case: Autronica

Vår bakgrunn og filosofi

- Utviklere er ikke skribenter
- Lag KUN dokumentasjon som er brukbar og nødvendig
- Man MÅ vedlikeholde dokumentasjonen
- Spør hvem du lager den for og hvorfor?

Realisering

- Høynivå arkitektur modell i UML (Unified Modeling Language)
- Lag design og kode dokumentasjon i Doxygen (Design og kode-dokumentasjon) samtidig som man skriver kode. Integrer UML tegninger og sekvens diagram i Doxygen
- Automatisering av tester (Unit/Integrasjon) skrevet før/samtidig som kode er viktig. Test dokumentasjon integrert med kode dokumentasjon.
- Kun noen få dokument skrives i Word og Excel. Eksempler er
  - FMEA (Failure Mode and Effects Analysis) og
  - hvordan vi etterfølger krav i tabellene i IEC 61508
  - Disse dokumentene oppdateres kun få ganger i løpet av samme prosjekt

Hva har vi lært?

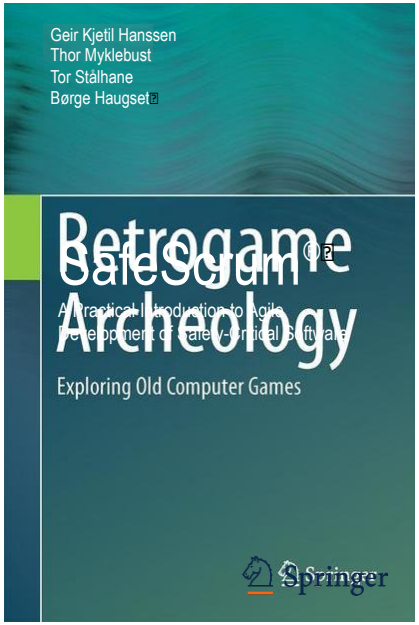
- Dette gikk bra!
  - Sprint Retrospective har fungert bra i forhold til å ta tak i utfordringer og holde høy kvalitet
  - Sprint review har fungert bra i forhold til programvare-teamet og intern produkteier
  - Demos er viktig for produkteier og ledelsen da de dermed lett ser den virkelige fremdriften
  - Å gjøre dette som et forskningsprosjekt har bidratt til gode diskusjoner og positiv holdning til endring
  - Verktøyene som brukes til utvikling må henge sammen, da mye informasjon ligger i linken mellom de forskjellige verktøyene. Tenk hel het i verktøy kjeden

Hva bør bedres?

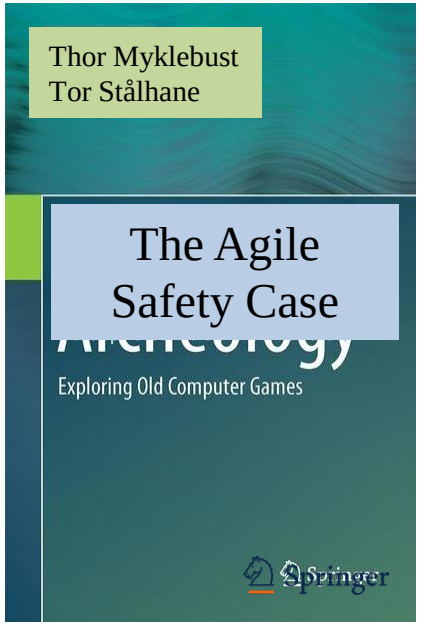
- Gjennomgang av design bedre for å forutse kommende design utfordringer tidligere
- Trenger gjennomgang av kildekode både i Bitbucket (versjonskontroll) og i tillegg foreta dybdegjennomganger (intensjon/design av koden)
- Kravstyring
- SafeScrum i forhold til Passport (gate system) må klargjøres og bedres
- Har etablert egen QA rolle i Sprint team da dette må bedres
- Estimering av tidsforbruk for utvikling av hele produktet
- Verktøykjeden må tilpasses aggregering av alle aktuelle dokumenter
- Organisasjonen må ha forståelse av hva Scrum er, da vi er avhengig av andre deler i organisasjonen. Kryss funksjonelle oppgaver er avhengig av personer utenfor selve teamet.
- Automatisering og regresjonstesting er svært viktig for å få til kontinuerlig forbedring av produkter.

Konklusjon

- ✓ Smidig har flere tilnærminger, f.eks. Sprint Demo som man med fordel kan benytte
- ✓ Smidig risikoledelse kan kombineres med tradisjonell risikoledelse
- ✓ Sikkerhets-standarder tillater bruk av smidige metoder, men tilpasning av smidige metoder er nødvendig



SafeScrum



Questions?

thor.myklebust@sintef.no
https://no.linkedin.com/in/thormyklebust
www.researchgate.net/profile/Thor_Myklebust

www.sintef.no/safetycase (Research, Certification and Consultancy)
www.sintef.no/IEC61508 (Research, Certification and Consultancy)
www.sintef.no/sjs (Railway)
www.safescrum.no (Software development)