Scaled Delivery

Samarbetsstrukturer för att självorganisera inom givna ramar

hakan.kleijn@Scania.com

DevLin

20 Oktober 2016

SCANIA

Scaled Delivery

Introduktion

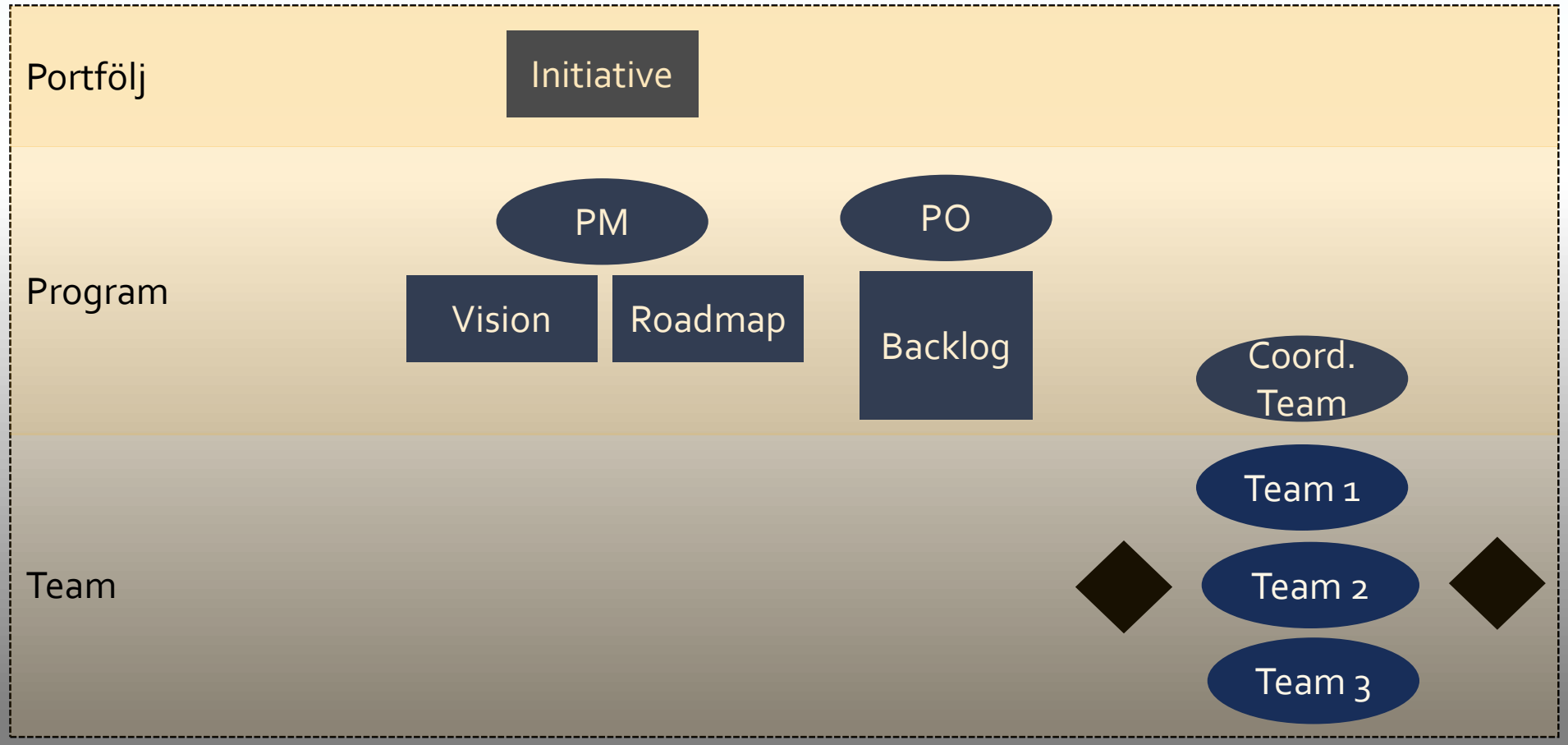
hakan.kleijn@Scania.com

DevLin

20 Oktober 2016

SCANIA

Scaled Delivery

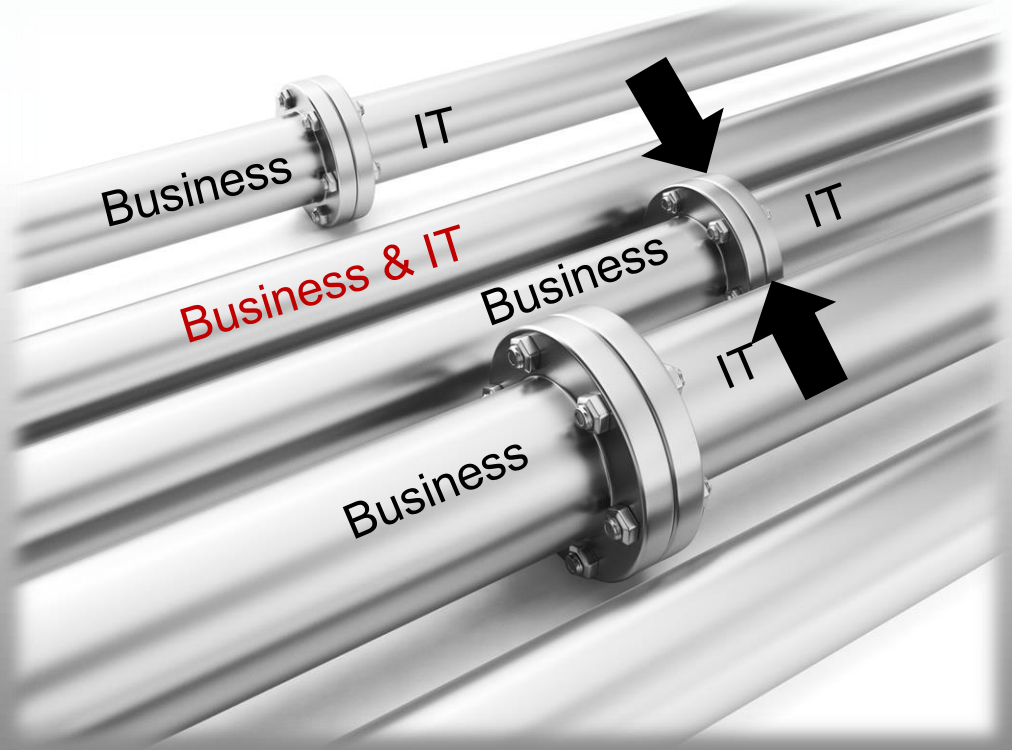


Varför ”scaled delivery”?

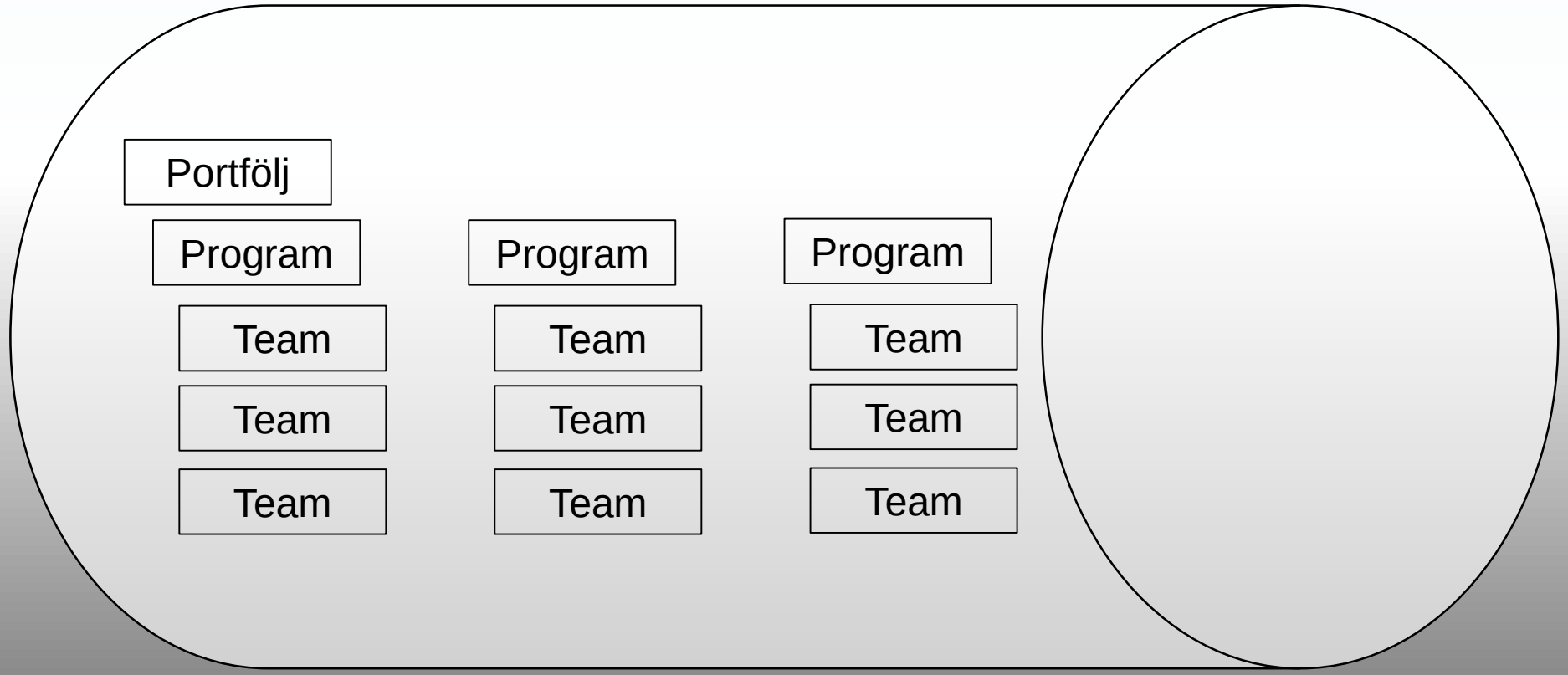
- Förbättra leveransförmågan
  - Hantera beroenden
    - mellan team
    - mellan system
    - mellan hårdvara / mjukvara
- Möjliggöra agil transformation

- Sträva mot samma vision / mål
- Leverera det viktigaste funktionerna
- Implementera enhetlig arkitektur
- Följa gemensamma utvecklingsprinciper / metoder
- Skapa enhetligt användargränssnitt

Organisation i pipelines



Varje pipeline indelad i 3 nivåer



Många beroenden mellan system

Serviceavtal

Exempel 1

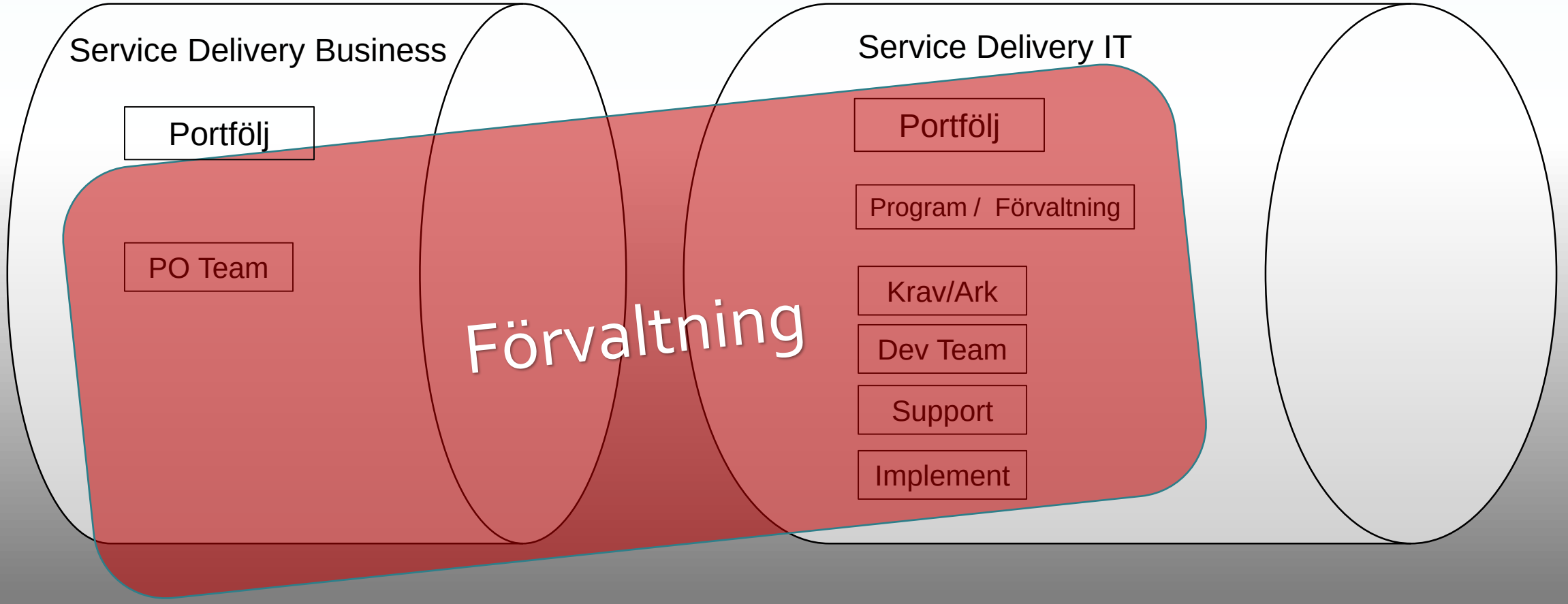
hakan.kleijn@Scania.com

SCANIA

Serviceavtal

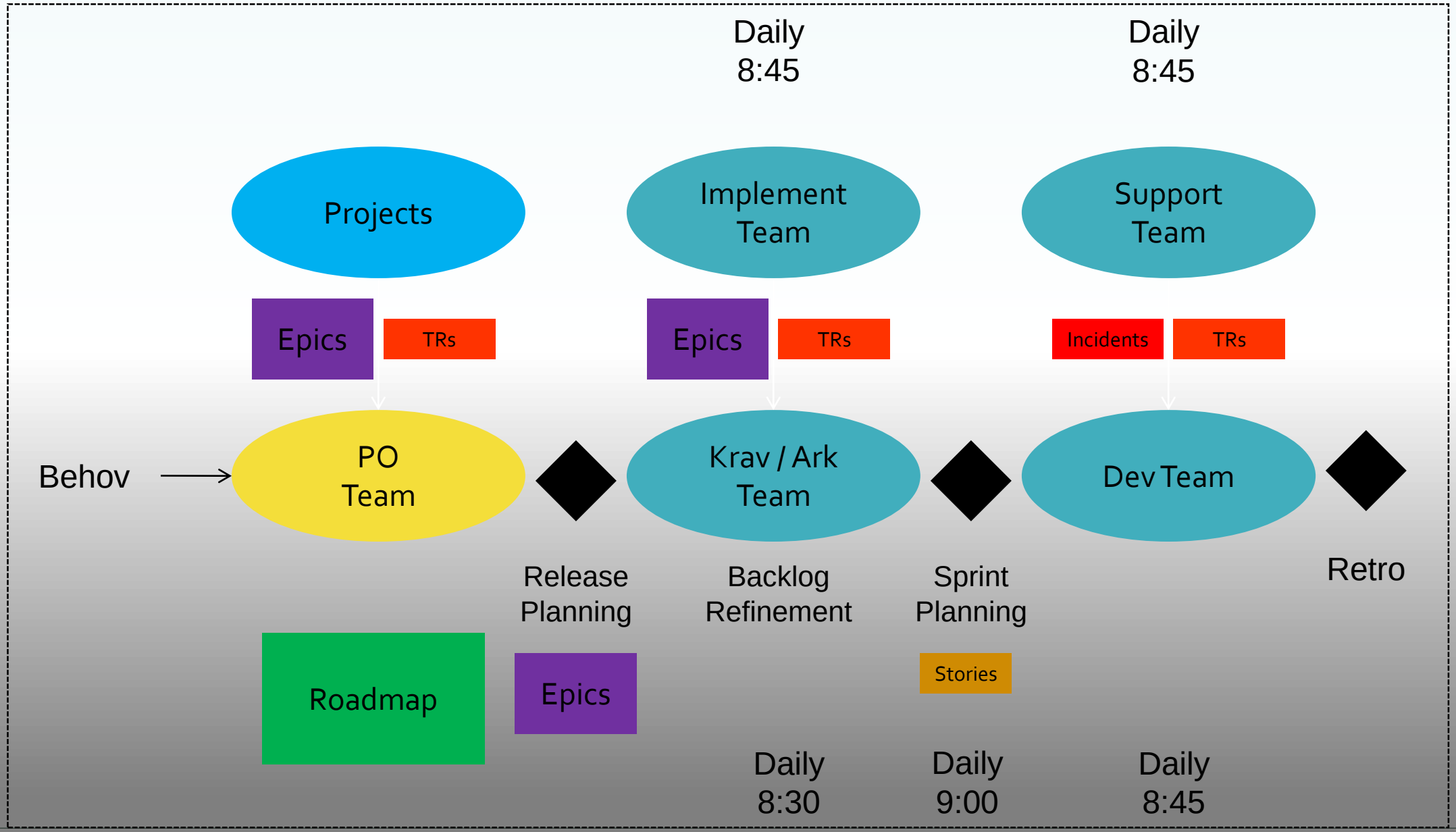
- Fakta
  - Global använding men 20 lokal installationer
  - Drygt 10 år gammal client-server arkitektur
  - Monolitisk kodbas
  - Automatiska byggen
  - Drygt 20 integrationer mot andra system både interna och externa
  - Manuell testning
  - 5 team, totalt 35 individer
  - Några individer jobbar ifrån extern lokal

Serviceavtal - Organisation



Serviceavtal - Utmaningar

- Produktägarskap
  - Många slutanvändare med olika behov
  - Närvaro hos teamet
  - Projekt som vill allokera teamet
- Prioritering
  - Affärsbehov, projekt v.s. arkitektur
- PO uttrycker behov som lösningar
  - Otydliga, ej testbara krav
- Lösningsförslag tas fram av krav/arkitekt och överlämnas till utvecklare
  - Utvecklare gör en annan lösning
- Testare vet inte vad eller hur de ska testa
  - Många patcher i samband med release


Daily
8:45
Daily
8:45
Projects
Implement
Team
Support
Team
Epics
TRs
Epics
TRs
Incidents
TRs
Behov
PO
Team
Krav / Ark
Team
Dev Team
Retro
Release
Planning
Backlog
Refinement
Sprint
Planning
Roadmap
Epics
Stories
Daily
8:30
Daily
9:00
Daily
8:45

Serviceavtal - Summering

Utmaningar

- Produktägarskap
  - Många slutanvändare med olika behov
  - Närvaro hos teamen
  - Projekt som vill allokera teamen
- Prioritering
  - Affärsbehov, projekt v.s. arkitektur
- PO uttrycker behov som lösningar
  - Otydliga, ej testbara krav
- Lösningsförslag tas fram av krav/arkitekt och överlämnas till utvecklare
  - Utvecklare gör en annan lösning
- Testare vet inte vad eller hur de ska testa
  - Många patcher i samband med release

Hantering

- Beskriva hela samarbetsstrukturen (med tavlor)
  - Definition of Done
- Release planning
  - 4 sprintars framförhållning
  - Prioritera uppdelning av monolit => testbar kod
- Epics / User stories
  - Acceptance test; given..when..then
- Sprint-indelning
  - 2 veckor leverans
  - 1 vecka med möten + lab/innovation
- Kvalitetsfokus
  - Plannera efter testkapacitet
    - => Fler testare
    - => Testautomatisering

Nästa steg - Serviceavtal

- Minska antalet lokala installationer
- Uppdatera arkitekturen - UI
- Dela upp monoliten i oberoende och testbar kod
- Testautomatisering

Projekt – Rebuild nytt system

Exempel 2

hakan.kleijn@Scania.com

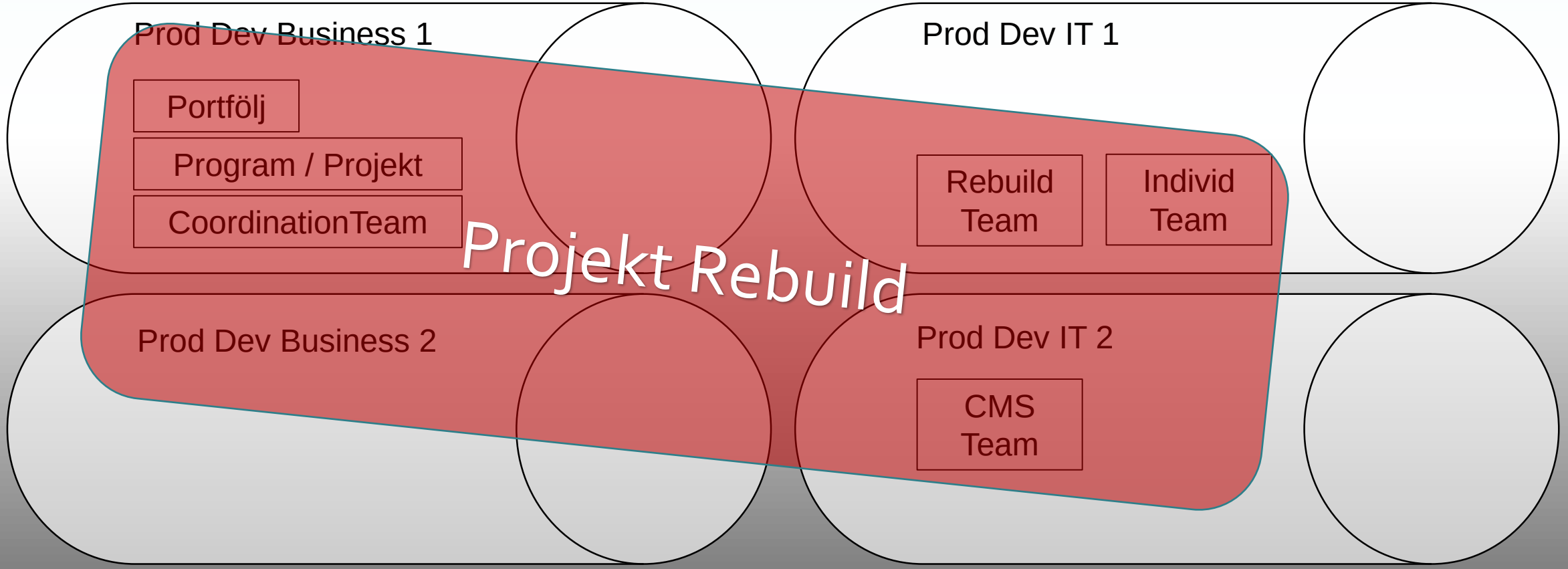
SCANIA

Project: Rebuild – nytt system

- Fakta
  - Global använding
  - Webbaserat användargränssnitt med integration mot två interna system
  - Automatiska byggen
  - Mest manuell testning
  - 4 team, totalt 25 pers
  - Sitter i olika lokaler

Organisationstillhörighet

Project: Rebuild – nytt system



Utmaningar

Project: Rebuild – nytt system

- 3 team i olika byggnader
- Explorativ utveckling
  - Sökalogritmer & API:er
  - Svårt att veta hur lång tid det tar att utveckla
- Personas, användarscenarion & wireframes
  - Detaljerade user stories med acceptanskriterier saknas
    - Testare vet inte vad eller hur de ska testa

Goals set on Sept 22nd

CMS

Create rest service API for advanced algorithm

Rebuild

Connect to new API
Create simple “POC” in the new GUI for testing the new service.
First version of SUS suggestion to discuss with SDP3

Individ

Map data for cable lists
Save data in CHIN for V-spec, UF’s , cable lists
Finalize how to handle missing data.
Identify if CHIN needs to work on changes
Decide how to send ”Certificate” information
Information regarding C-specification; who to talk & where to find it?

User stories

I&D for lock codes
Initial concept for RBS at the dealer
Use case for deviating time (Niklas will find examples)

End-user Demo

Fri

Stories

Stories

Project: Rebuild – nytt system

Utmaningar

- 3 team i olika byggnader
- Explorativ utveckling
  - sökalogritmer & api:er
  - Svårt att veta hur lång tid det tar att utveckla
- Personas, högnivå användarscenarion i kombination med wireframes
  - Detaljerade user stories saknas
  - Testare vet inte vad eller hur de ska testa

Succéer

- Att sitta ihop
- Release planning
  - Tydliga release mål / datum
- 1 veckas sprint
  - Tydliga sprintmål
- User stories
  - Acceptance test; given..when..then

Nästa steg – Rebuild nytt system

- Continuos Deployment
- Testautomatisering
  - Unit
  - Integration
  - Acceptanstest / validering

Scania Diagnose and Programmer (SDP)

Exempel 3

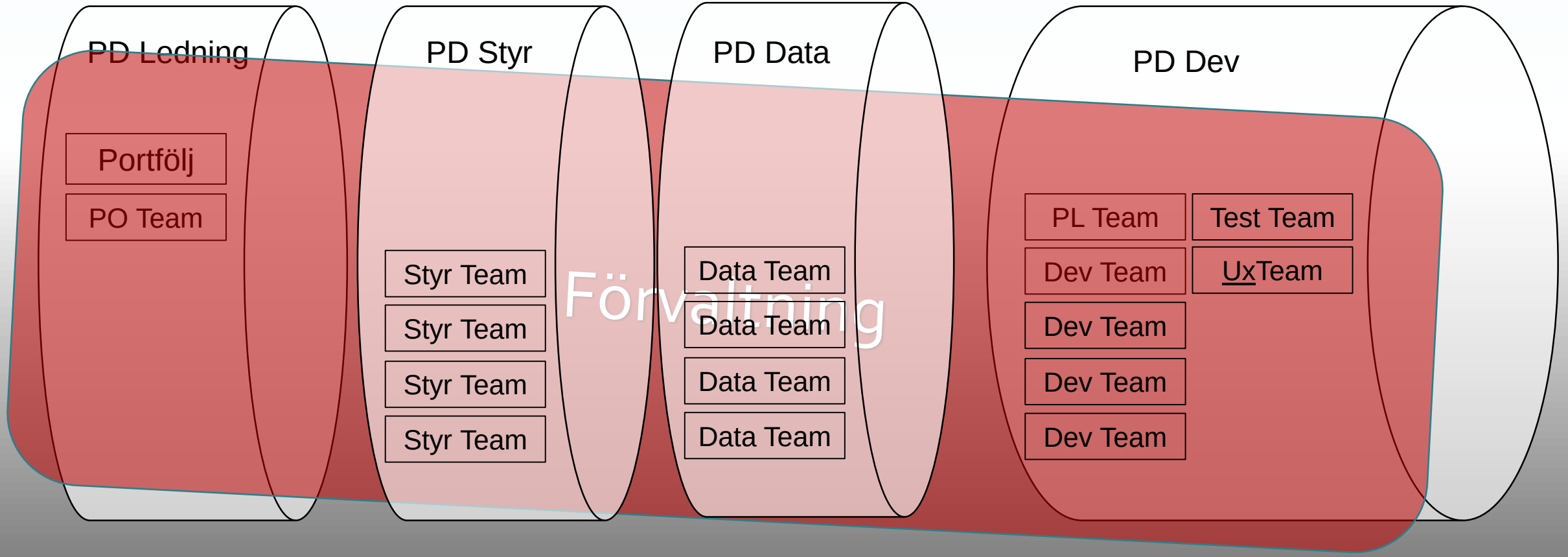
hakan.kleijn@Scania.com

SCANIA

SDP

- Fakta
  - Global användning
  - Drygt 10 år gammal client-server arkitektur
  - Monolitisk kodbas
  - Hårdvaruberoende = styrenheter
  - Automatiska byggen
  - Många integrationer mot andra system både interna och externa
  - Manuell testning
  - 7 team + två hela sektioner, totalt 150 individer
  - Sitter i två lokaler

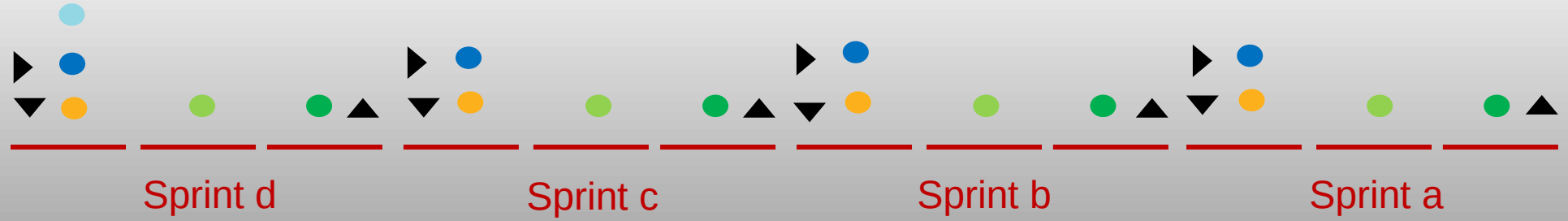
SDP - Organisation



SDP - Utmaningar

- Produktägarskap
  - För många ärenden i backlogg, + 500
  - Hinner inte gör rätt prioritering mellan affärsbehov
- Otydliga/obefintliga, ej testbara krav
  - Utvecklare vet inte vad det ska utveckla
  - Testare vet inte vad eller hur de ska testa
- Sekventiell leverans
  - Många överlämningar; Styr > Data > Dev > Test

- Releaseplanering - kommande och nästa - PO Team
- Roadmap – hur ser den grova planen ut? – PL Team + Data + Styr (virtuellt team)
- Featureplanering – vad ska brytas ned? – PL Team
- Featurenedbrytning – user stories – PL + Lead Dev + Ux + Behovställare (virtuellt team per feature)
- Story specification – acceptanskriterier – PL + Dev Team + Test Team



- Samsynk mellan teamen
- Sprintplanering i teamen
- Demo – teamen för varandra

SDP - Summering

Utmaningar

- Produktägarskap
  - För många ärenden i backlogg, + 500
  - Hinner inte gör rätt prioritering mellan affärsbehov
- Otydliga/obefintliga, ej testbara krav
  - Utvecklare vet inte vad det ska utveckla
  - Testare vet inte vad eller hur de ska testa
- Sekventiell leverans
  - Många överlämningar; Styr > Data > Dev > Test

Hantering

- Release planning
  - 4 sprintars framförhållning
  - Uppdelning av monolit => testbar kod
- Features / User stories
  - Acceptance test; given..when..then
- Beskriva hela samarbetsstrukturen
  - Bokade möten
  - Virtuella team med rätt kompetens

SDP - Nästa steg

- Trimma in samarbetsstrukturen
  - Virtuella team
- Mer testautomatisering
- Ersätta SDP med nytt system

Scaled Delivery

Summering

hakan.kleijn@Scania.com

DevLin

20 Oktober 2016

SCANIA

Scaled Delivery…

- …är ett medel
- …men bara om det är nödvändigt
- …behövas för att hantera stora ”initiativ” som berör många system
- …behövs för monolitiska system som kräver många team för att underhålla koden
- …kan minimeras vid oberoende, automatiskt testbar och deploybar kod som kan releasas on demand


SCANIA