

Software Engineering

Kristian Sandahl

Niklas Lanzén

Kort presentation



- Kristian Sandahl, 63 år
- Började Datateknik 1978
- Första arbetsgivaren efter examen var LiU
- Konsult 1986-1987
- Ericsson 1995-2001
 - olika roller: forskning, utbildning, mjukvaruutvecklare, projektledare,...
- Åter på LiU sedan 2001
 - Forskar inom industriell programvaruutveckling
 - Undervisar i projektkurser
 - Samverkanskoordinator
- Koordinerar profilen storskalig mjukvaruutveckling

Kort presentation

- Niklas Lanzén, 52 år
- Började Datateknik 1992
- Första arbetsgivaren efter examen var ett konsultbolag
 - Första uppdraget var på Ericsson
 - Många olika roller: mjukvaruutvecklare, testare, projektledare,...
- Anställd på Ericsson sedan 2008
 - Många olika roller: designer, projektledare, produktledare, chef, ...
- Anställd på Combitech sedan oktober 2016
 - Change manager på Ericsson
 - Projektledare – Digitaliseringsprojekt på LKAB
- Från sommaren 2022 - Utvecklingschef på Agricam
 - Agricam jobbar med digital djurhälsa för kor
- Mina erfarenhet bygger på 21 år på Ericsson
- Sitter med i den nämnd som bla ansvarar för U och D



Mål med föreläsningen

Motivera er att bli intresserade av alla perspektiv av ett programmeringsprojekt

Motivera er att bli stolta mjukvaruutvecklare



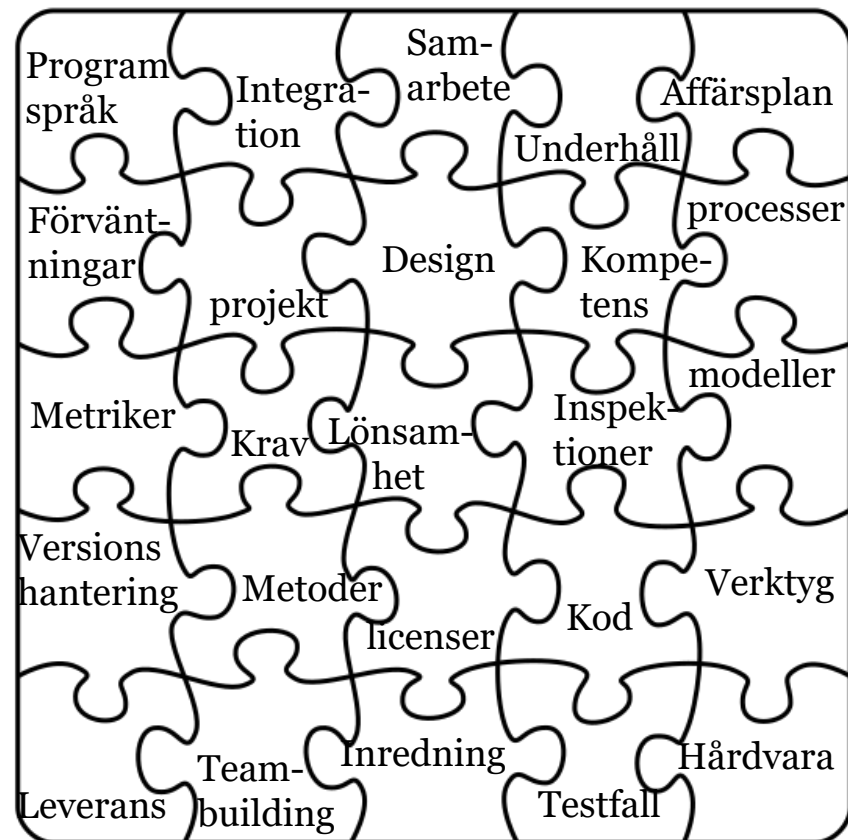
Definition av Software Engineering

- Software engineering is...
- ...”the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software”.

(ISO/IEC/IEEE 24765:2010 Systems and software engineering—Vocabulary)

Många bitar skall fogas till en helhet

- Blandningen av:
 - Programmering
 - Datalogi
 - Matematik
 - Människor
 - Ledarskap
- Är det sköna med Software Engineering



Programvara kan användas till mycket



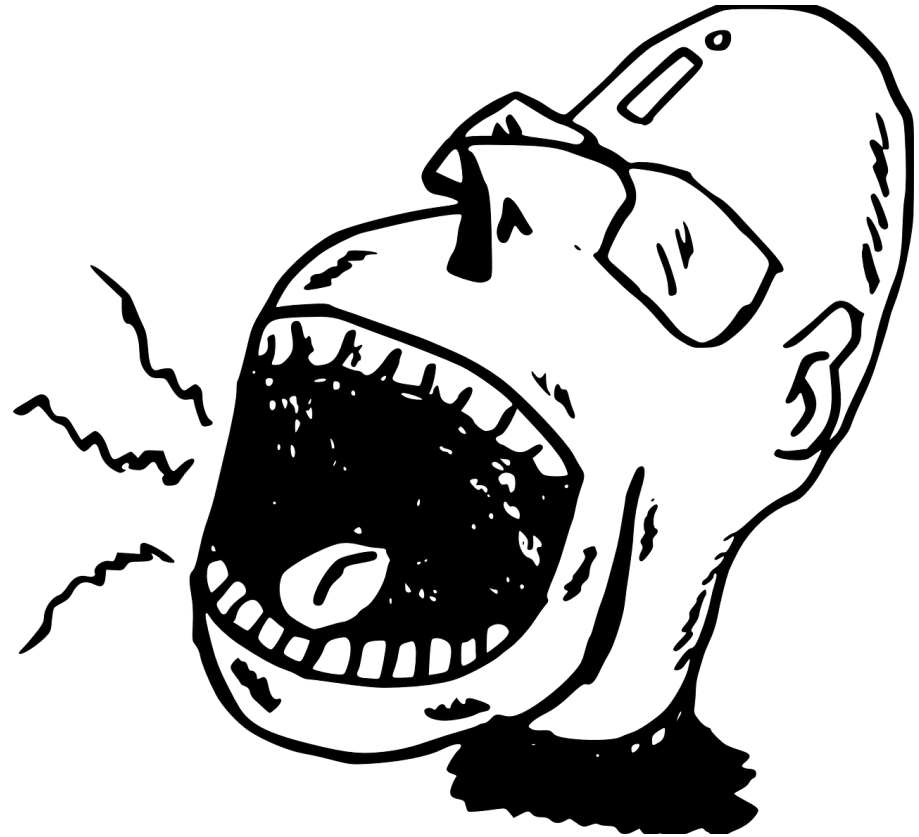
Pong

Therac-25



Kravhantering specificerar förväntningar och ramar

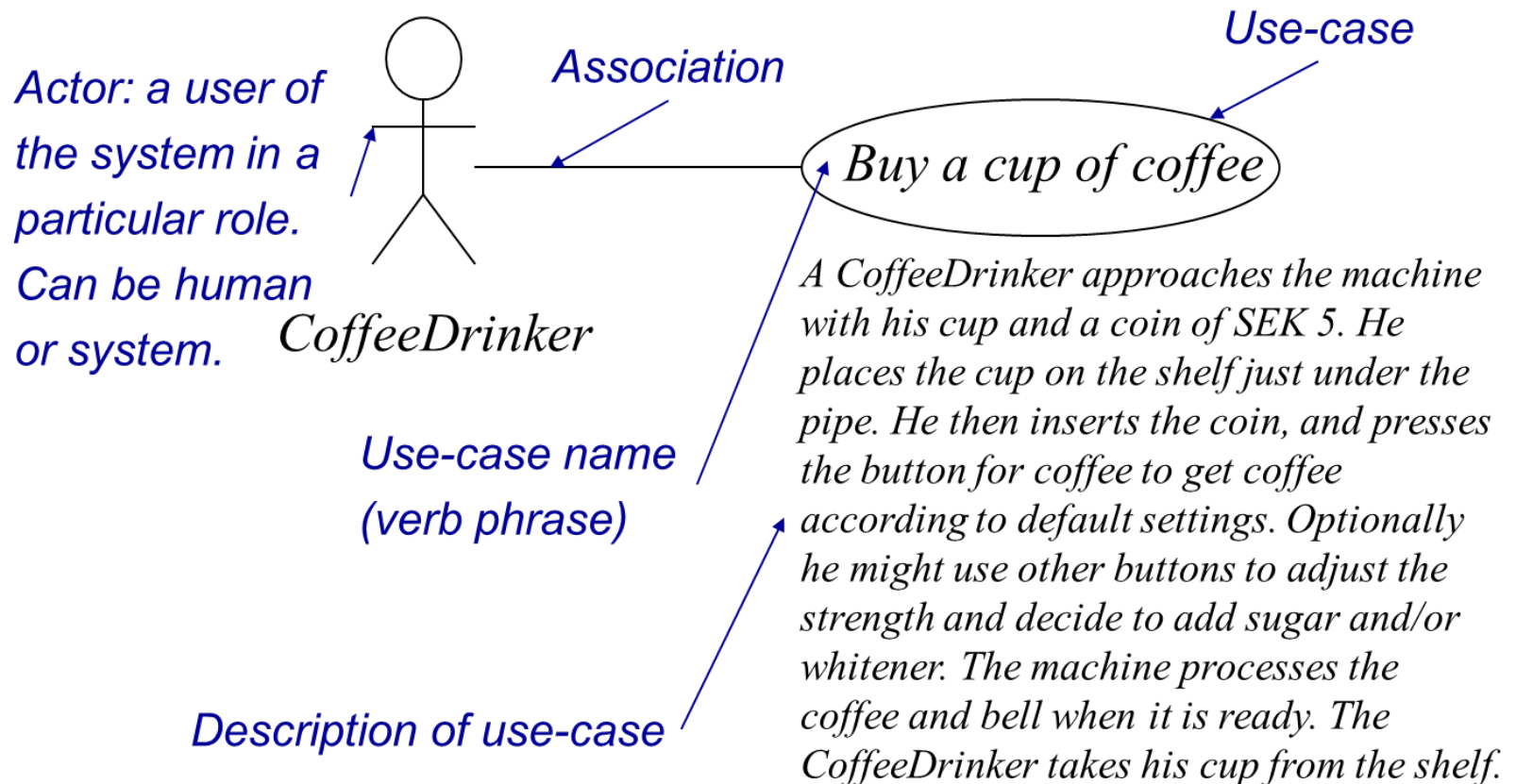
- Utmaningar:
- Finna **verkliga** behov
- Finna alla intressenter
- Välja och schemalägga
- Tolka och testa



Krav på väl organiserad textform

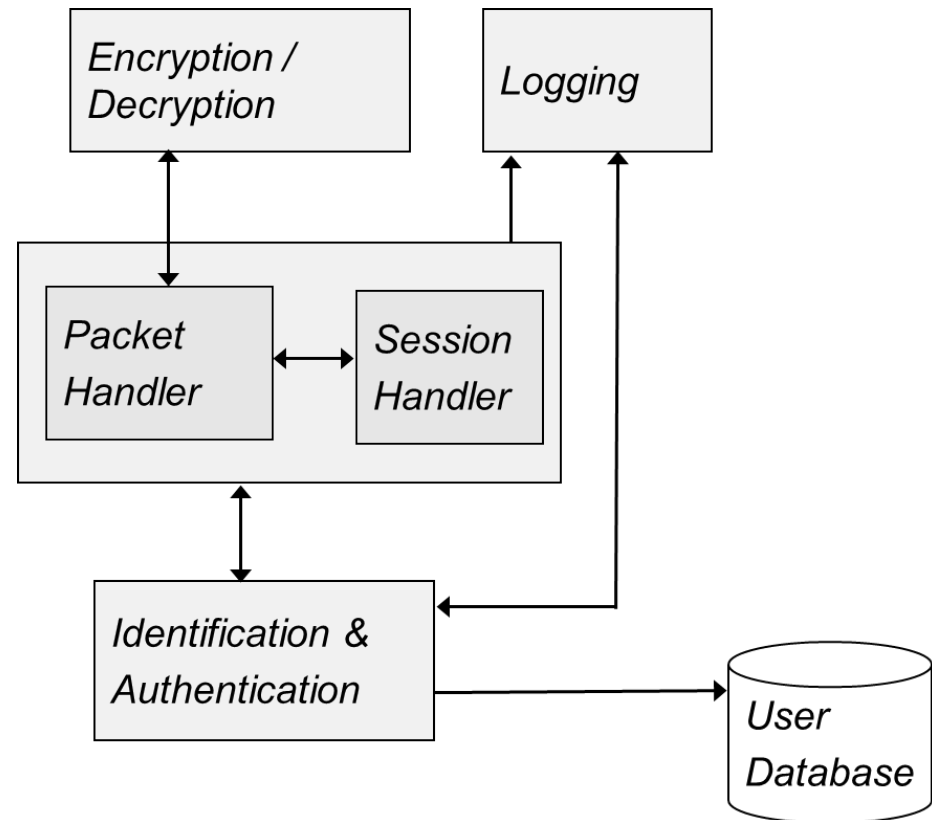
- 3.2 Functional requirements
 - 3.2.1 Show colors
 - When the user enters type of T-shirt, the system shall show a palette of available colours.
 - 3.2.2 Add to shopping basket
 - The user shall be able to add an item to the shopping basket.
- 3.3 Performance requirements
 - 3.3.1 Response time
 - The minimum response time is 2.0 seconds.
- 3.4 Design constraints
 - The system shall be implemented in php.

Krav som användningsfall

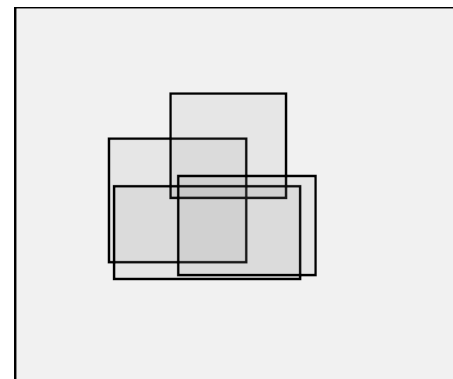
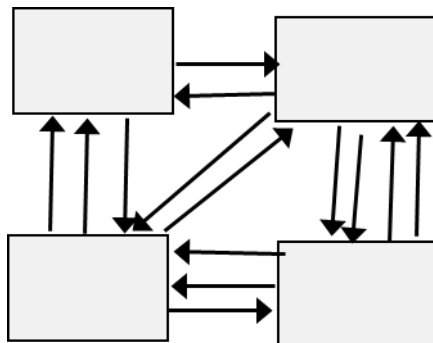
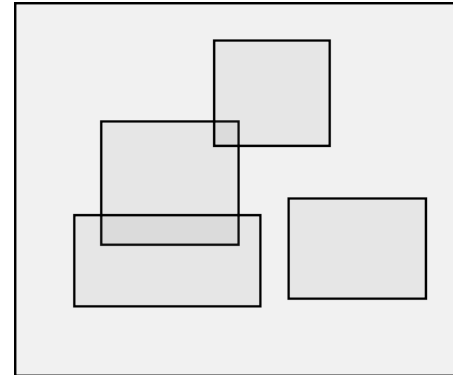
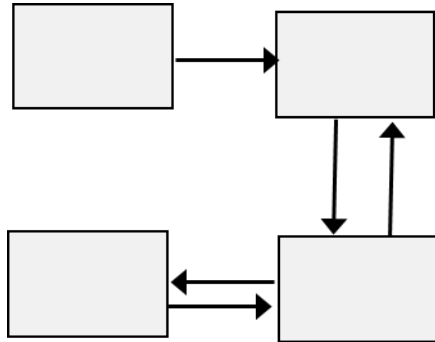


Arkitektur beskriver komponenter och gränssnitt

- Utmaningar:
- Känna till vad som finns
- Balansera kvalitetskrav
- Definiera stabila gränssnitt
- Förbereda för ändringar

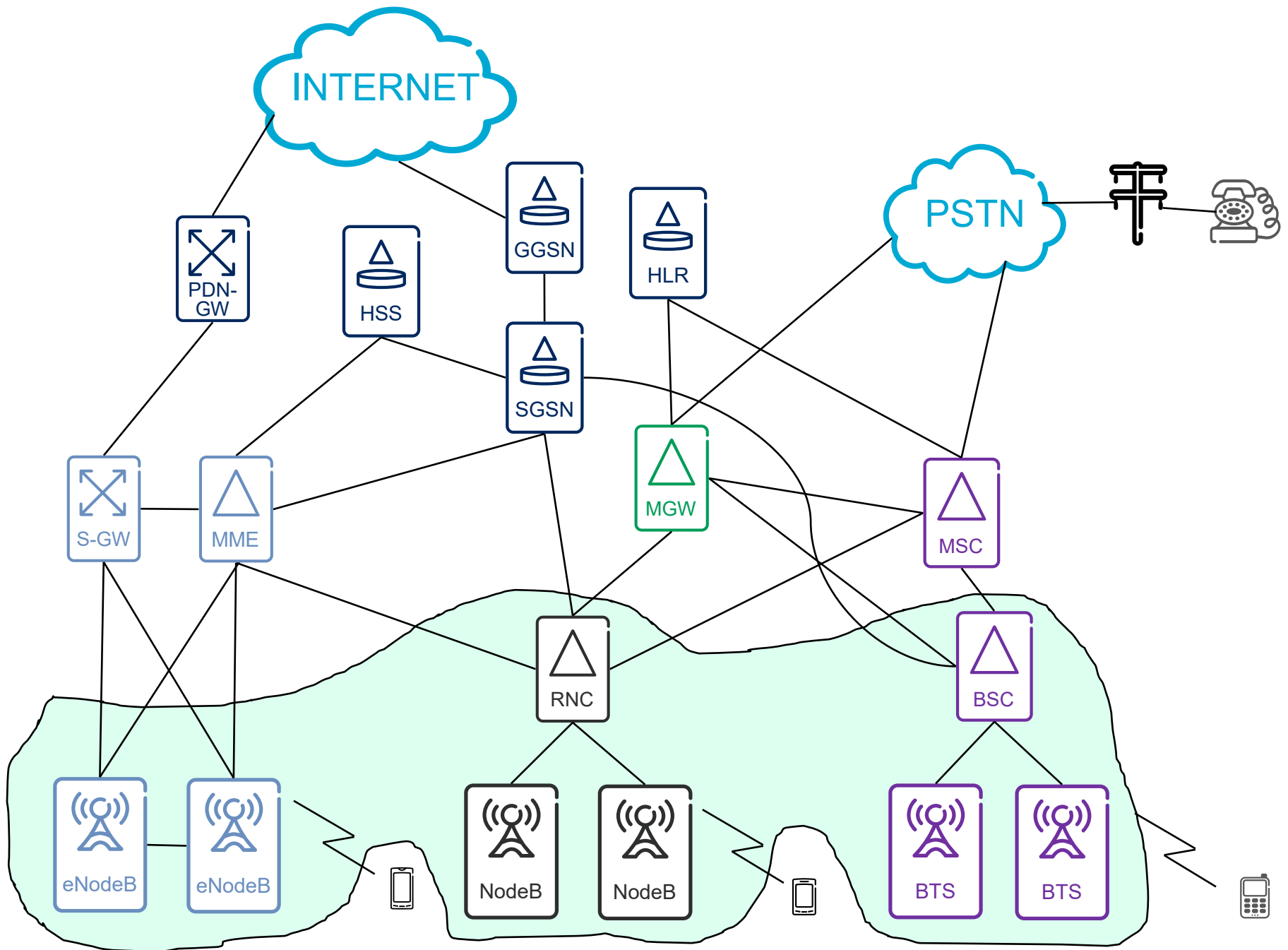


Koppling och kohesion



Vad gör Ericsson i Linköping?

- Research & Development inom (mjukvaruutveckling)
 - LTE
 - 5G
 - Plattform för basstation
 - Säkerhet
- Underhåller och bygger globalt labb



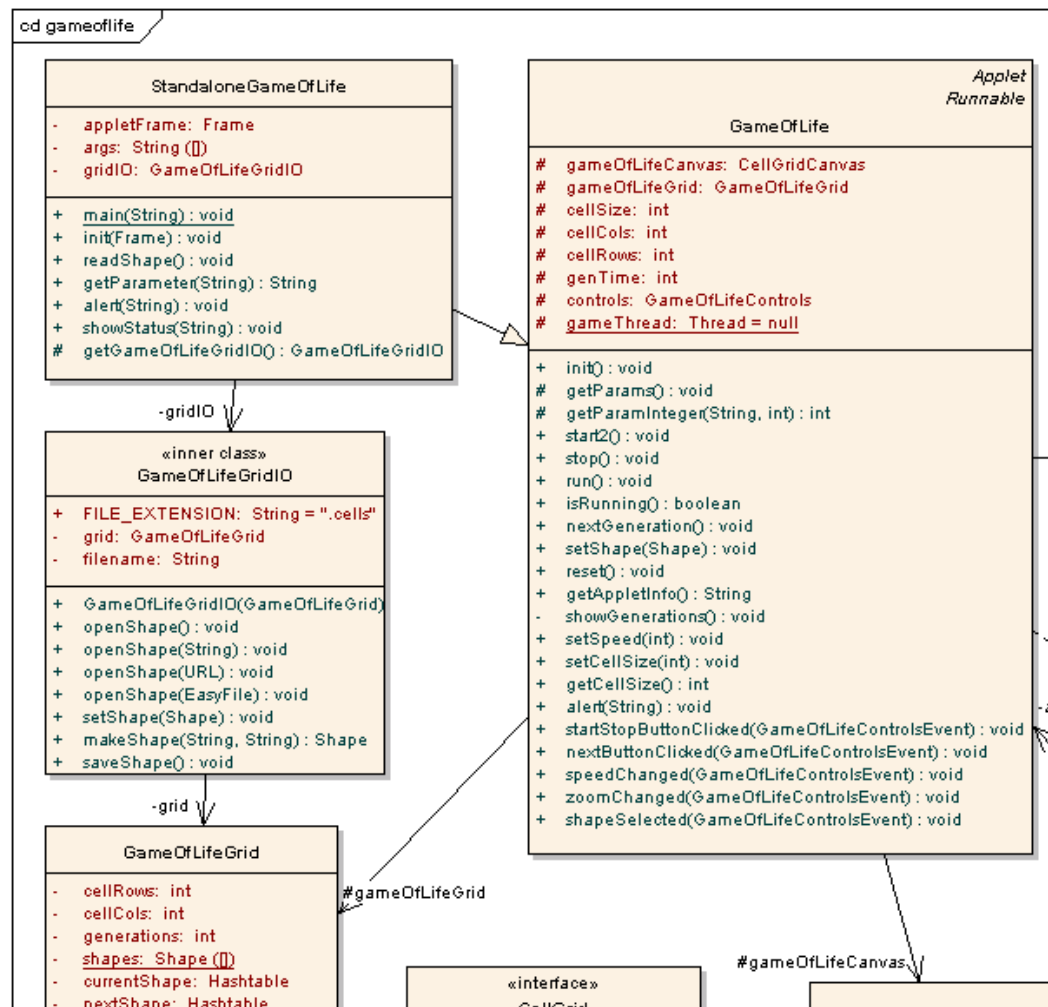
Mjukvaruutveckling på en basstation

- En beräkningstung del
 - Hårdvara: DSP-kluster med många kärnor
 - Språk: C
- En kontrolldel
 - Hårdvara: vanlig CPU med få kärnor
 - Språk: Modellering / C++
- En molndel
 - Hårdvara: Vanlig server-HW
 - Delar av kontrolldelen flyttar upp i molnet



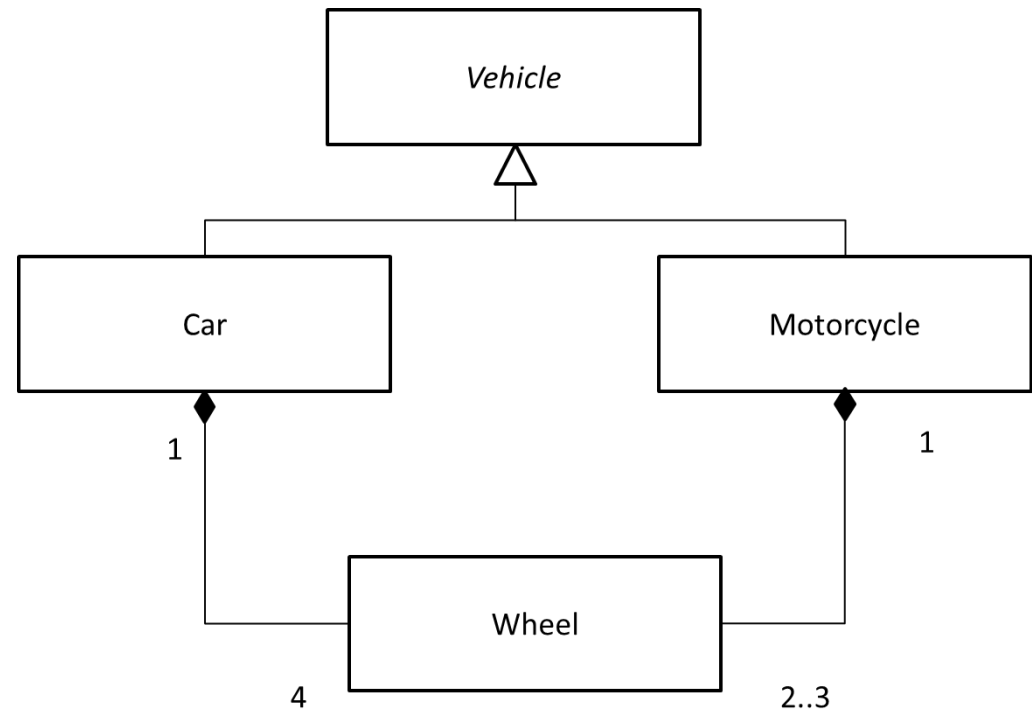
Detaljerad design

- Utmaningar:
- Realisera funktion
- Förbered för ändringar
- Begripligt system
- Bidrag till kvalitet
- Behöver ofta en förklarande text



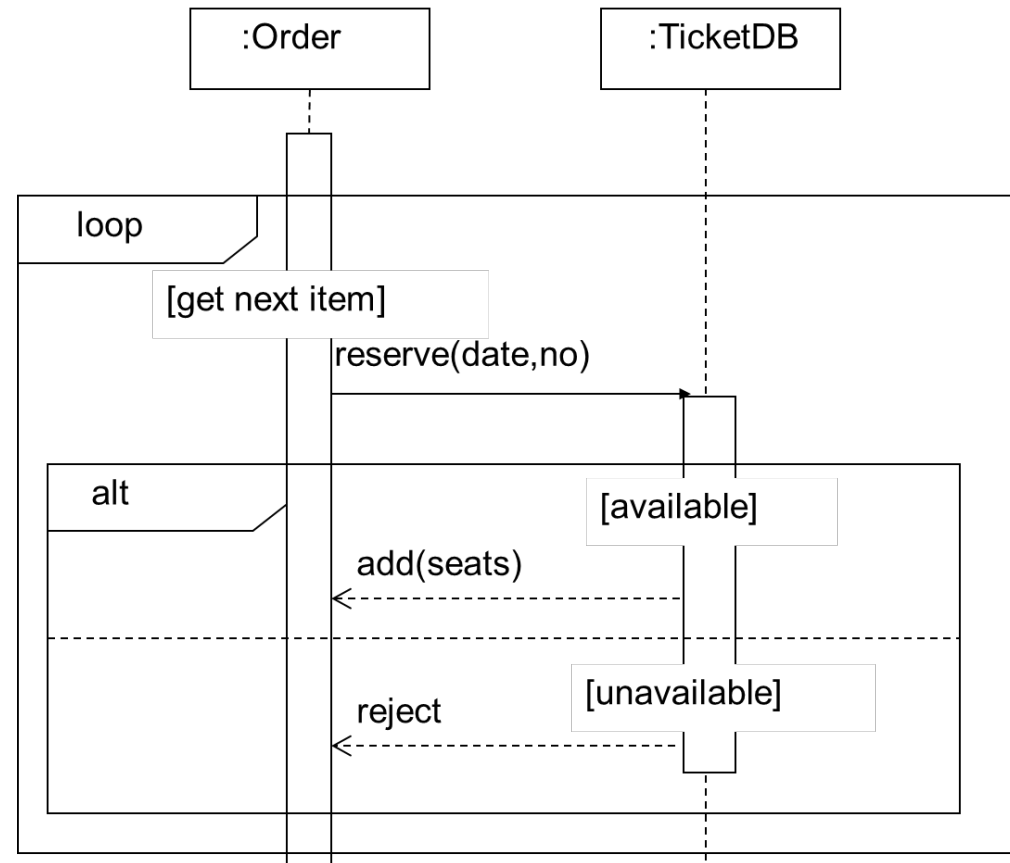
Statisk beskrivning med UML klassdiagram

- Mycket vanligt diagram
- Kan innehålla detaljer om exakta attribut och funktioner
- Kan generera kodskelett
- Kan genereras från kod

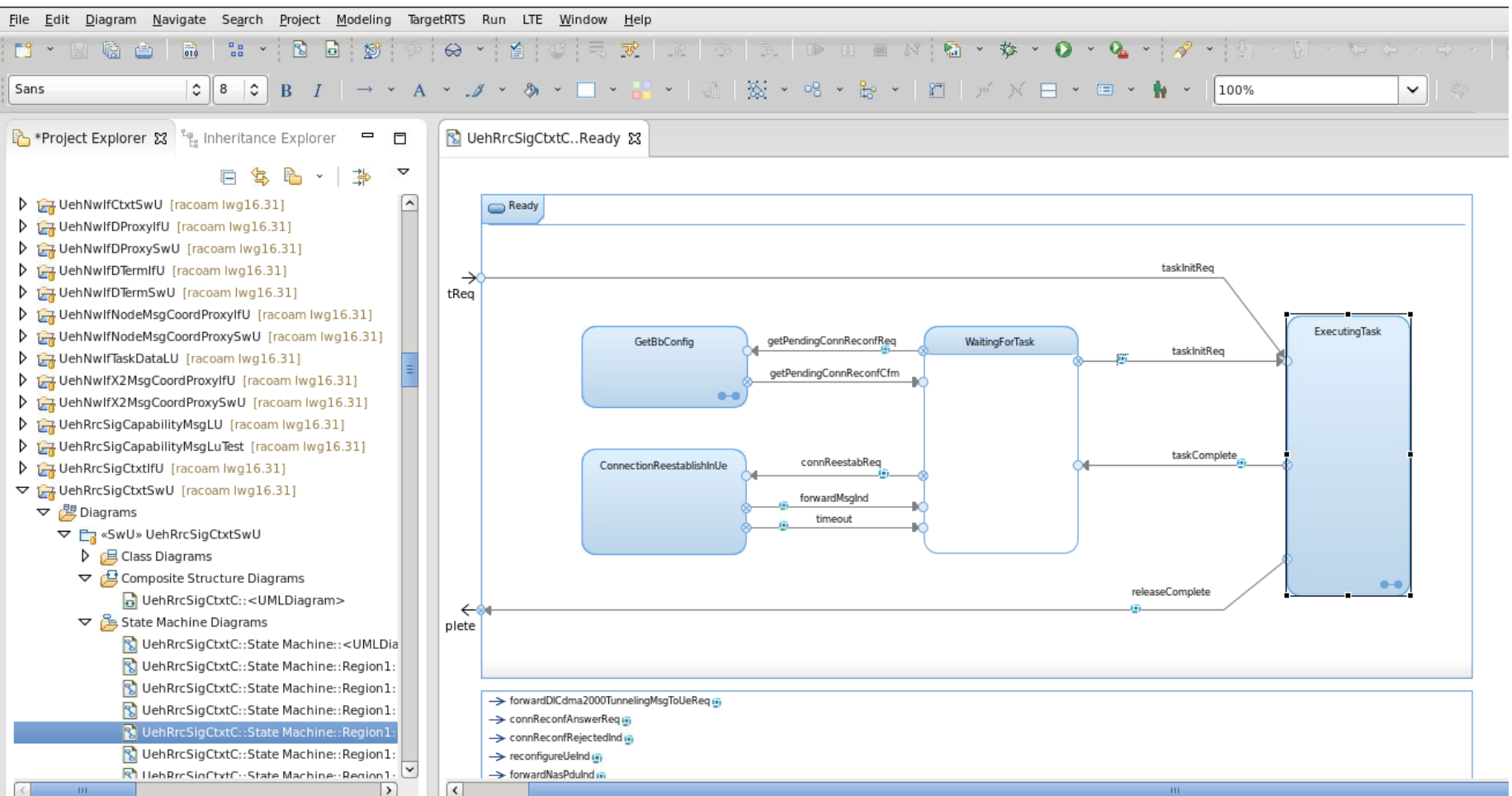


Dynamisk modell med UML sekvensdiagram

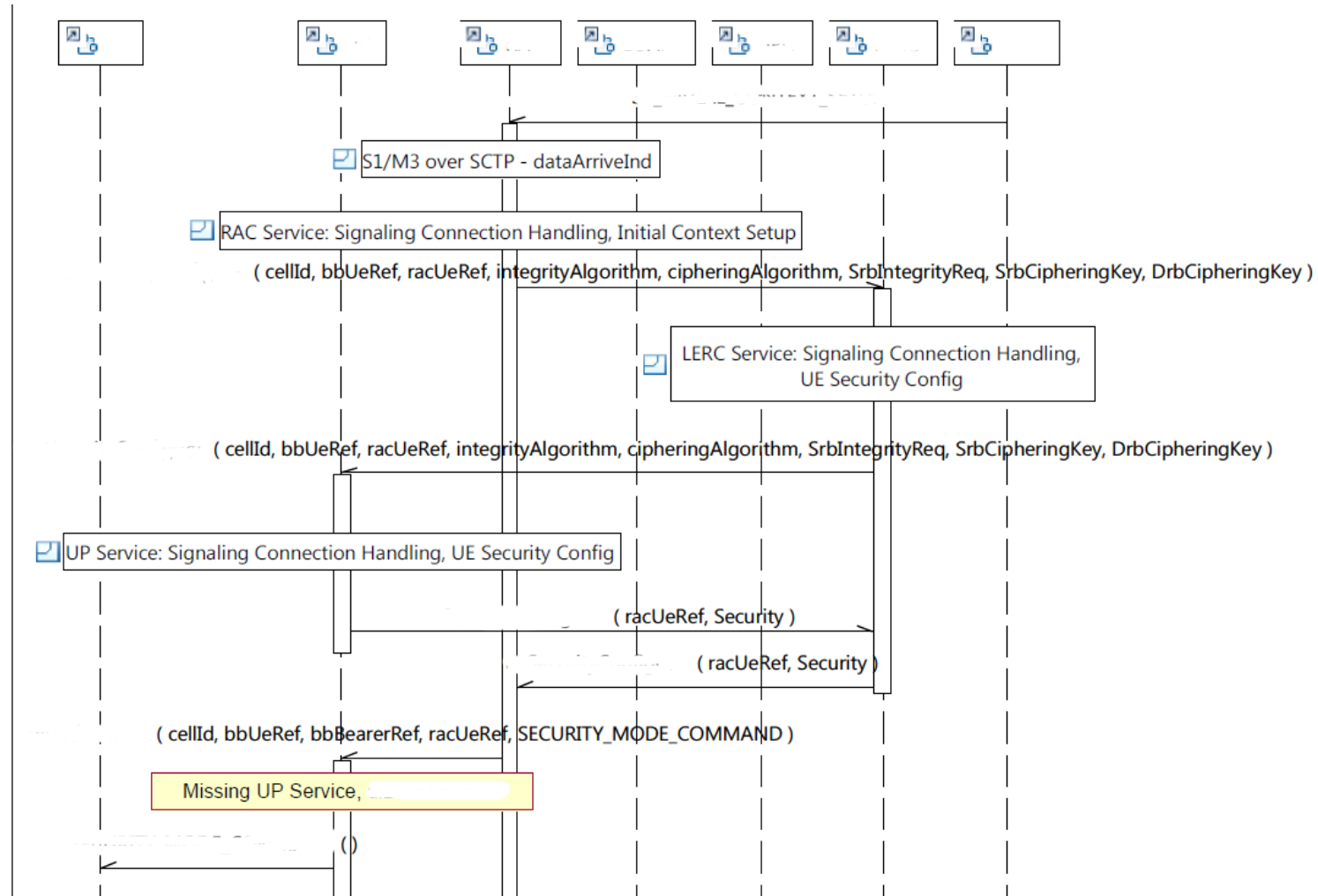
- Härstammar från telekommunikation
- Viktig för realtids-system
- Kan bli hur komplicerad som helst.



Exempel på modellering i en basstation 1 (2)



Exempel på modellering i en basstation 2 (2)



Implementation

- Programming
- Kodstandard
- Enhetstest
- Läsbarhet!

```
class Ball(pygame.sprite.Sprite):
```

```
# Constructor. Pass in the color of the block, and its x and y position
```

```
def __init__(self):
```

```
    # Call the parent class (Sprite) constructor
```

```
    super().__init__()
```

```
# Create the image of the ball
```

```
self.image = pygame.Surface([self.width, self.height])
```

```
# Color the ball
```

```
self.image.fill((white))
```

```
# Get a rectangle object that shows where our image is
```

```
self.rect = self.image.get_rect()
```

```
# Get attributes for the height/width of the screen
```

```
self.screenheight = pygame.display.get_surface().get_height()
```

```
self.screenwidth = pygame.display.get_surface().get_width()
```

```
# Speed in pixels per cycle
```

```
self.speed = 0
```

```
# Floating point representation of where the ball is
```

```
self.x = 0
```

```
self.y = 0
```

Programmeringstekniker VS Domänkunskap

Tekniker

- Språk – C/C++, Java, Python, ...
- Algoritmer/mönster
- Modellering

Domäner

- Ericsson – telekom
- Saab – flyg
- Apputveckling på Android

- Konsekvenserna skiljer sig om man gör fel – Hur?

En stolt utvecklare **försöker förstå** både **teknikerna** och **domänen** som används för att kunna utveckla bra kod!

Varför?

Omfattning?

Testning

En bra utvecklare kan testa sin kod

Testning – Varför?

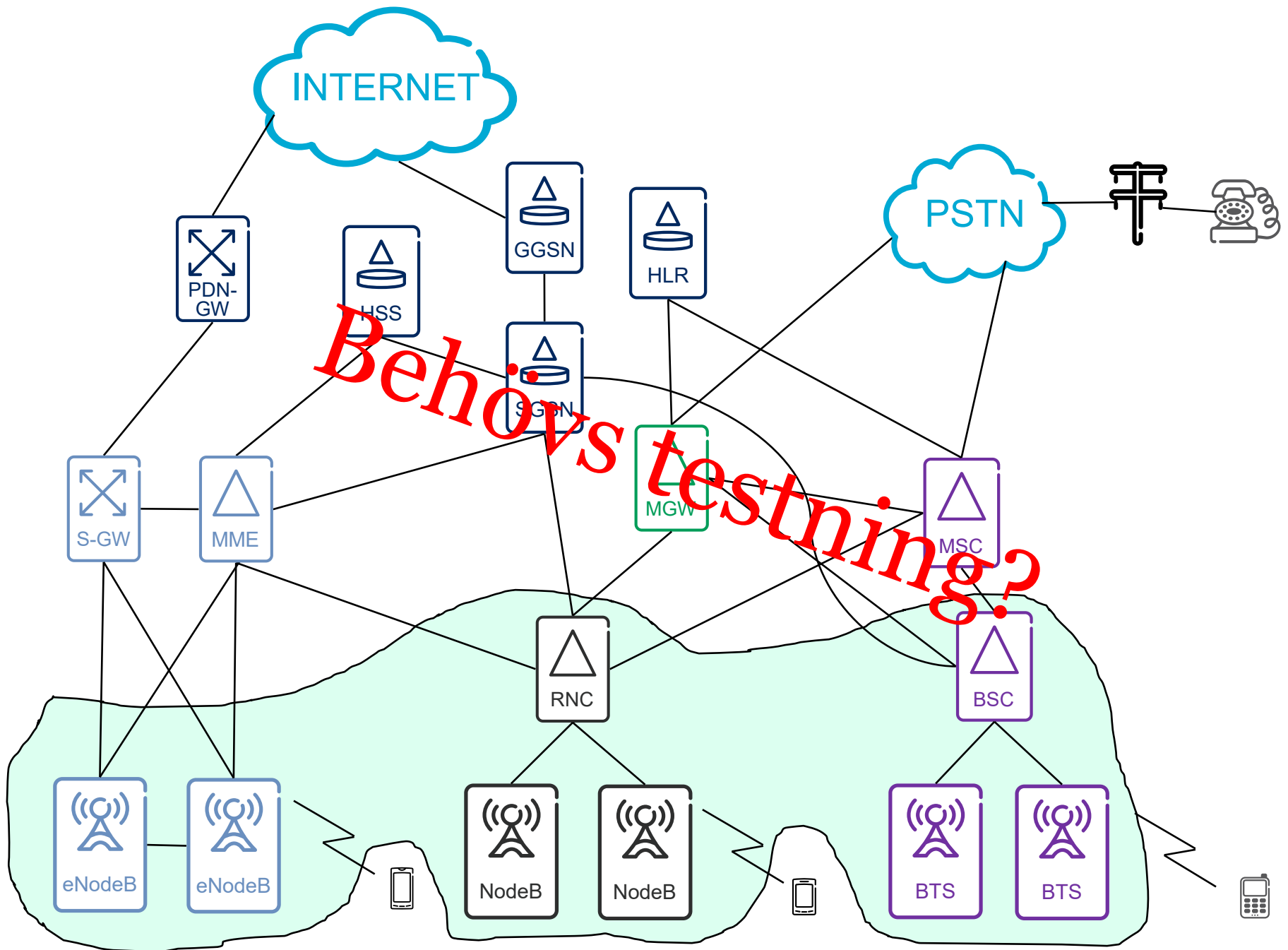
- Varje **användare** som råkar ut för en **bugg** blir **besviken**
 - Konsekvensen blir dock olika beroende på vilken mjukvara som drabbas – business drabbas på olika sätt
- **Svårt** att **prediktera** hur lång **tid** det tar att utveckla mjukvara **utan tester**
 - När är du klar om du inte kan visa att kraven är uppfyllda?
- **Svårt** att lägga till **funktionalitet** i mjukvara **utan tester**
 - Hur vet du att du inte gör sönder tidigare funktioner?

En stolt utvecklare lämnar inte ifrån sig otestad mjukvara!



Testning - omfattning

- Olika nivåer av tester
 - Basic-/modul-test – tex testning av funktioner
 - Funktionstest – tex testning av delsystem
 - Systemtest – tester på hela systemet, tex av prestanda, stabilitet
- Testkoden kan vara lika omfattande som själva produkten
- Viktigt att stor del av testningen är automatisk
 - Kostsamt att köra tester manuellt, man hinner inte
 - Med continuous integration kan varje litet funktionstillägg testat var för sig => enklare att hitta vad som introducerade ett ev fel



PAUS

Vattenfallsmodellerna

Krav

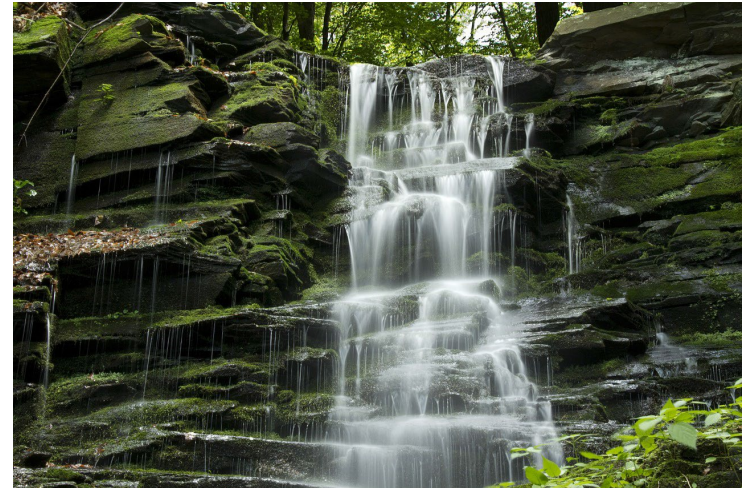
Arkitektur

Design

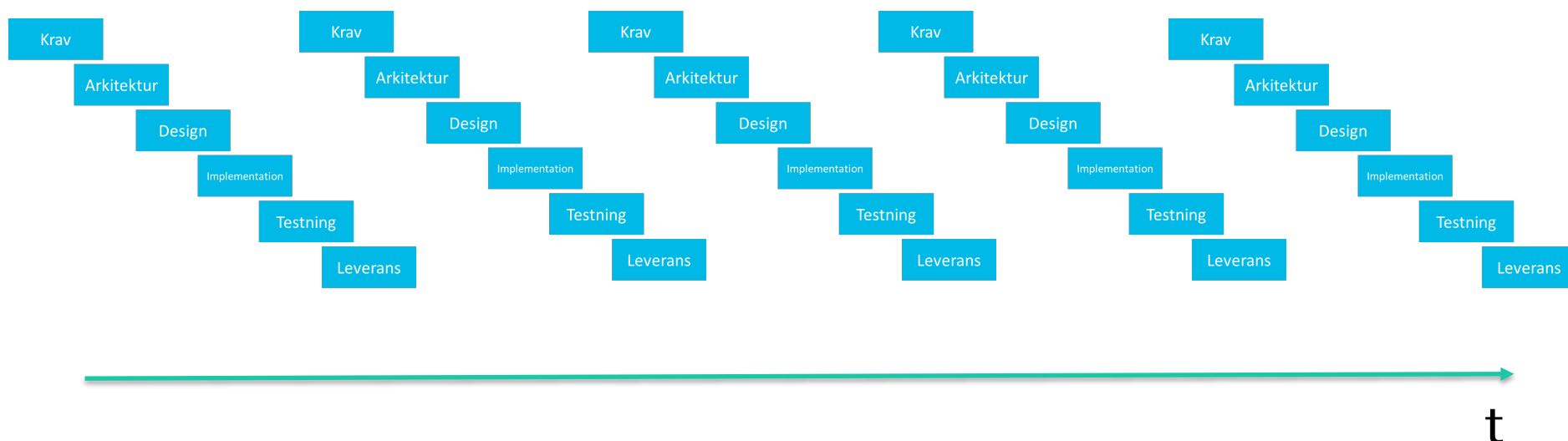
Implementation

Testning

Leverans



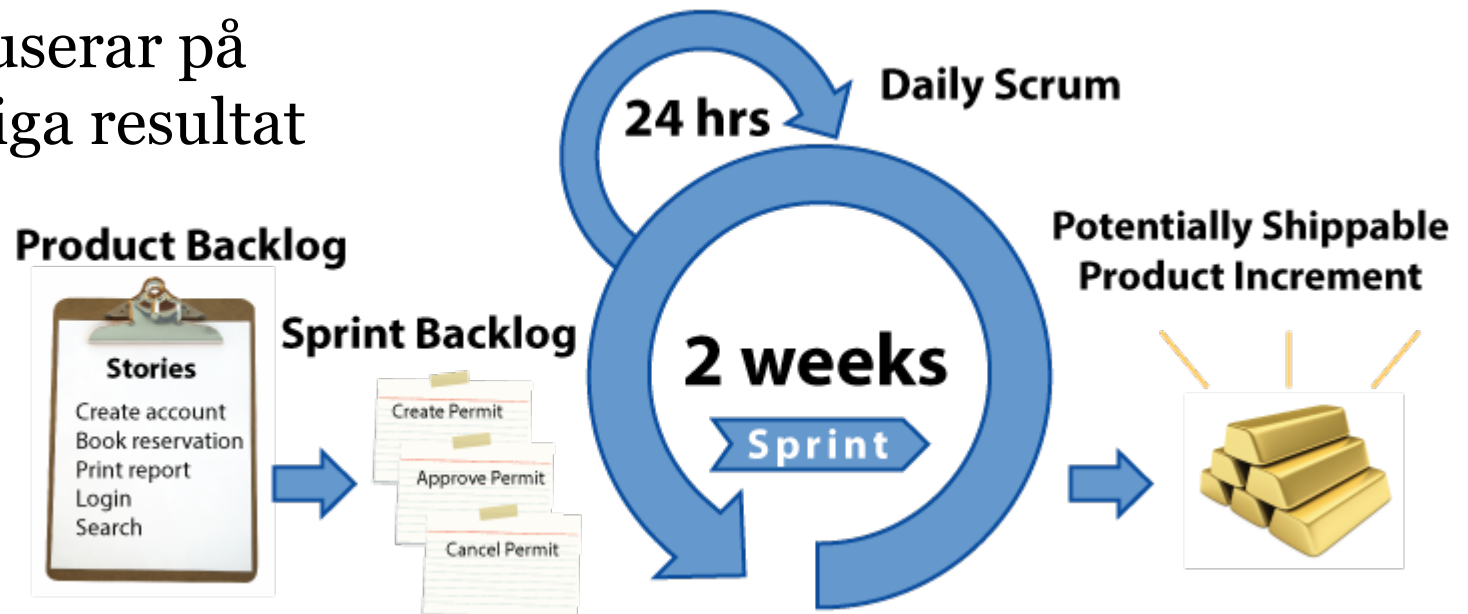
Iterativ modell



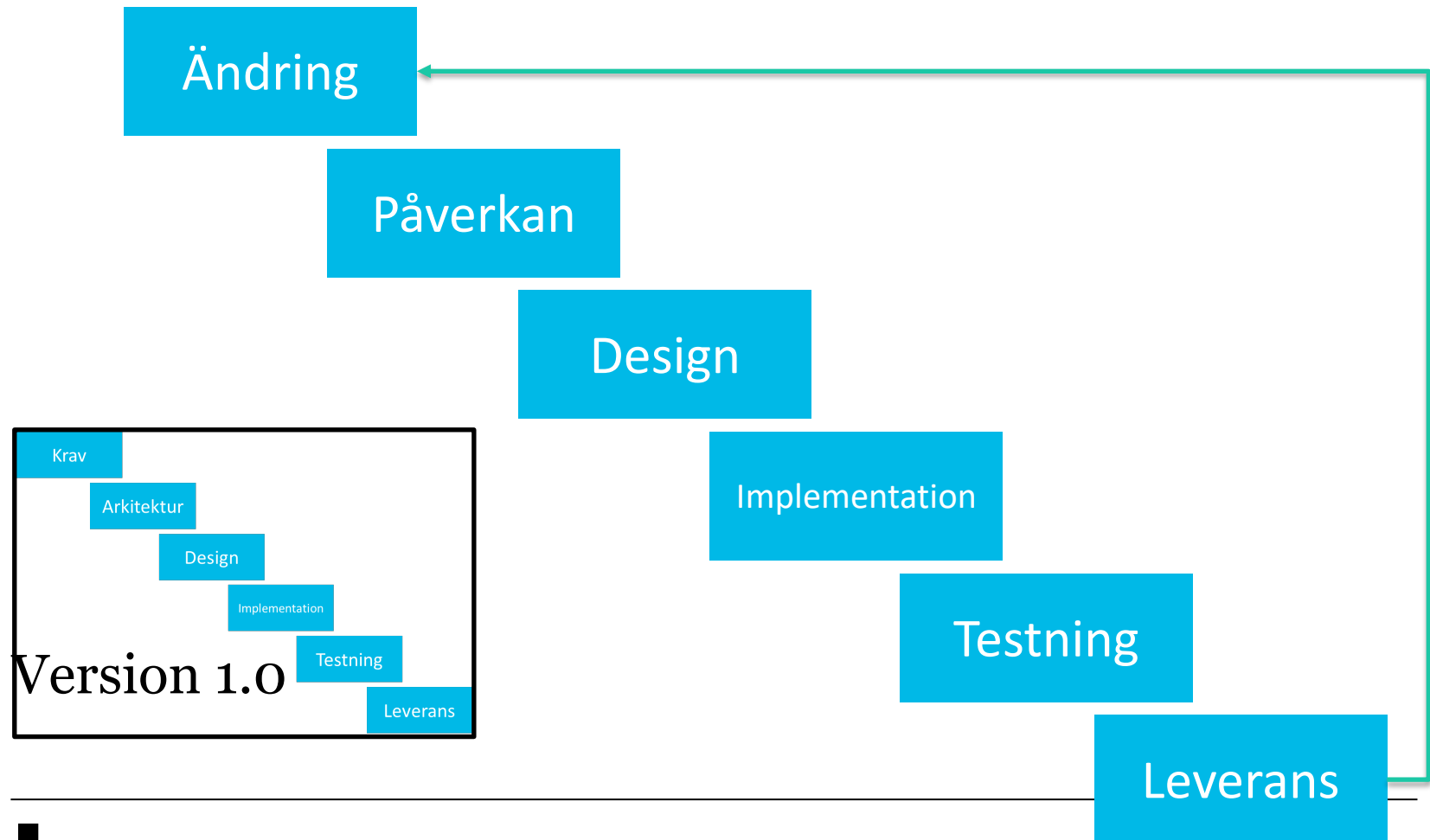
Återkoppling från kund oftare
Alla krav måste inte vara kända i detalj i förväg
Delleveranser möjliga
Ett måste för användbara system?

Följsamma (en. Agile) metoder

- Mycket korta iterationer
- Fokuserar på team
- Fokuserar på synliga resultat
- Undvik onödiga investeringar
- Hantera ändringar



Underhåll står för ca 70% av kostnaderna



Arbeta i team



Hur jobbar teamen på Ericsson?

- 7-9 personer
- **XFT** – Cross Functional team
 - Alla discipliner i samma team (**design, implementation, test**)
 - Utbildning för att klara alla discipliner sker ofta i projektet
 - Ett team kan ta ett antal krav och designa, implementera och testa dem så att de nya kraven kan levereras till kund
- **Bestående team** som lär sig jobba ihop
 - Grundas i respekt för varandra
- Nästan **alla team** jobbar **agilt**

Alla utvecklare jobbar i team!

Hur har pandemin påverkat teamarbete?

Hur fungerar det mellan team på Ericsson?

- Alla team levererar **kod** till **samma kodbibliotek**
 - Ställer stora krav på testning innan leverans
- Helst en funktion som implementeras av ett team
- **Implementera funktionen iterativt** så att det alltid går att leverera något som skapar värde
- En stor men viktigt ny funktion kan behöva delas mellan flera team
- **Många team** jobbar med **stora produkter**
 - Finns produkter med > 100 team utspridda över världen

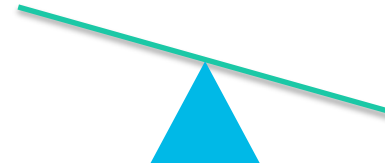
Alla behöver kunna jobba i team

Ensam kvinna i ett team

“Alla passar lika bra för linjen. Jag har aldrig tänkt att män passar bättre. “

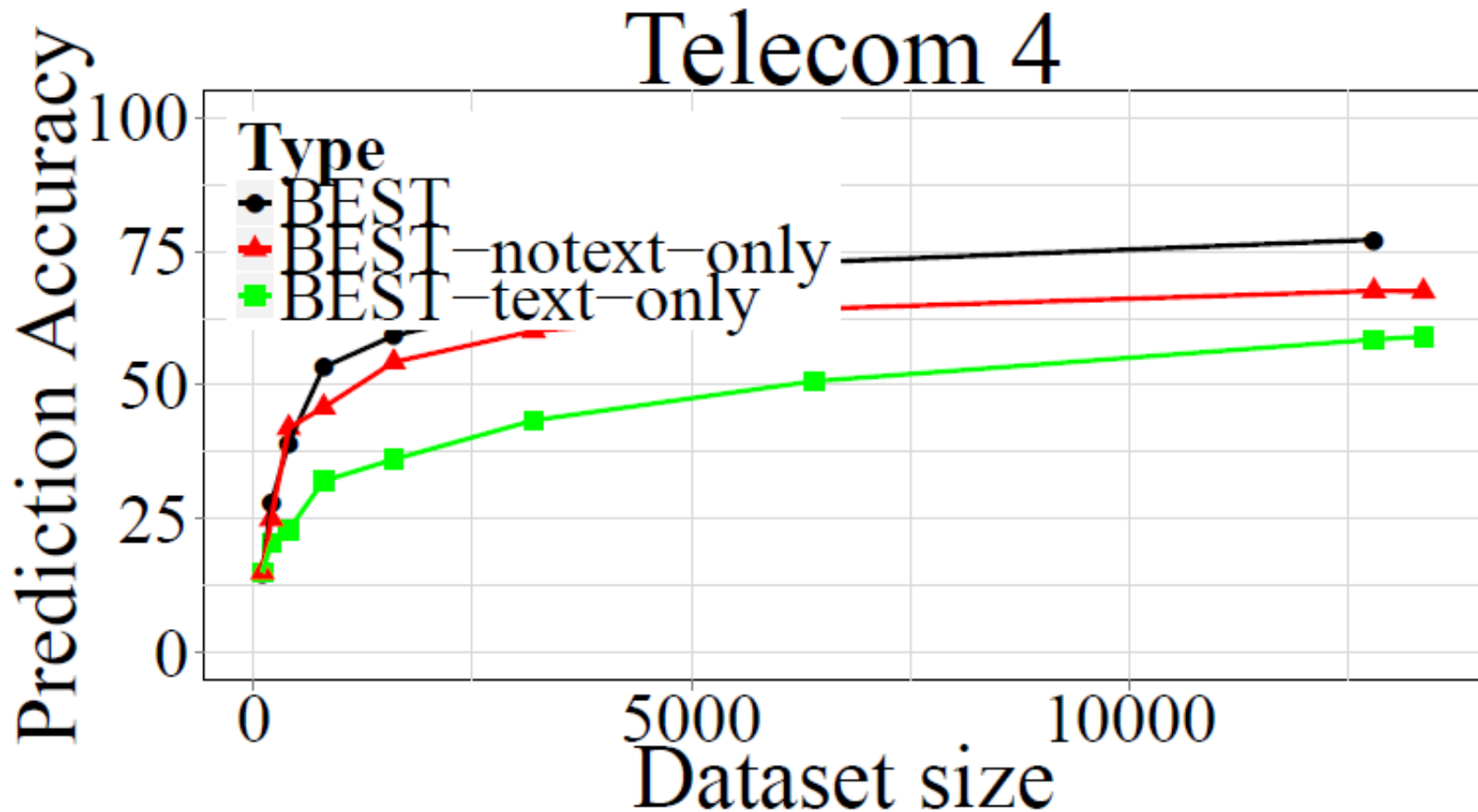
”Killar backar varandra väldigt bra och att tjejer på utbildningen inte gör det.”

“Får uppgifter som inte är lika krävande, jag som kvinna får inte möjligheten att utveckla kompetensen. Jag tror man har fördomar om vad kvinnor är kapabla av, att jobbet är för män och ger helst uppgiften till en man. “



”De som betar sig på ett sätt som är mindre bra mot kvinnor kan vara socialt otrygga.”

Forskning: Bug localization with ML



Kurser inom SE



En stolt utvecklare

- Lämnar inte ifrån sig otestad mjukvara!
- Försöker förstå både teknikerna och domänen som används för att kunna utveckla bra kod!
- Bidrar till teamet!



Questions?