



# Arkitektur-orienterad teststrategi

SAST-Väst 2008-05-15

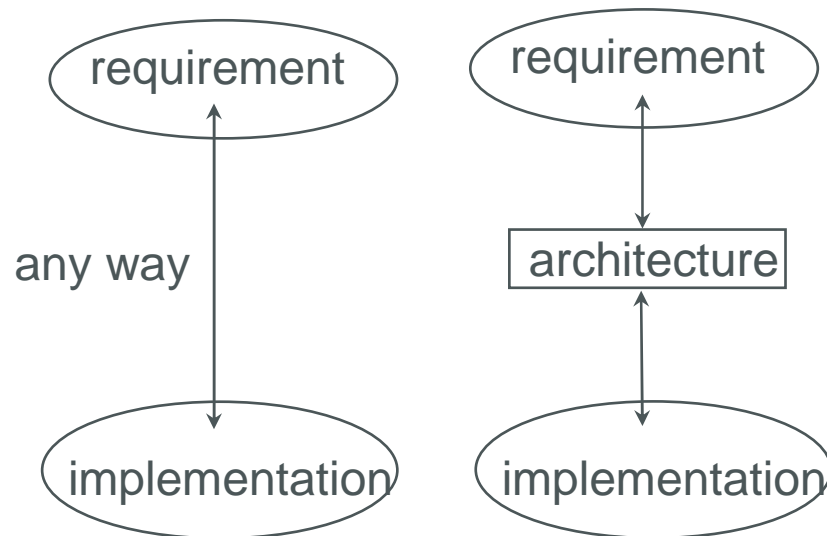
Stig Ursing

[stig.ursing@eis.semcon.com](mailto:stig.ursing@eis.semcon.com)

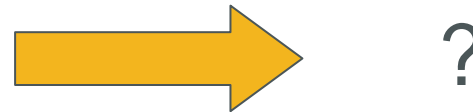


# Varför?

- Vi arbetar med stor ansträngning med att införa arkitekturarbete och arkitektur inom systemutveckling
- Då ställer man sig gärna frågan  
”Vad innebär detta för verifiering o validering?”



Kravbaserad test  
(Blackbox metoder)



Implementationsbaserad test  
(Whitebox metoder)

# Arkitektur-orienterad test

1. Test utgående från arkitekturbeskrivningar av testobjektet
2. Modellering av test med användning av diagram
3. Egenskapen Testbarhet

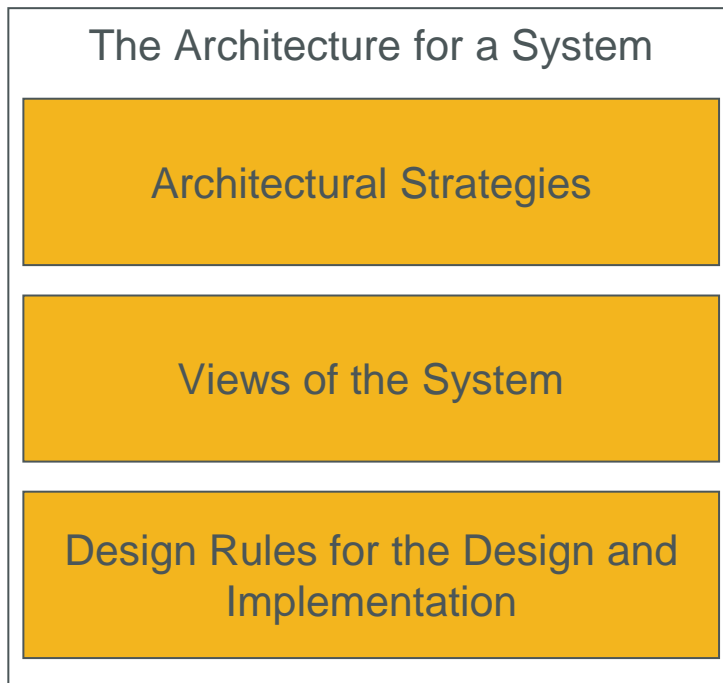
To describe a complex system in an understandable way requires that you can separate different aspects of the system and not try to describe all problems at one time.

# Arkitektur

- Ett viktigt syfte med ett arkitekturarbete är att hantera komplexiteten i ett system
- Vad är arkitektur?
  - De egenskaper vi vill ge en design

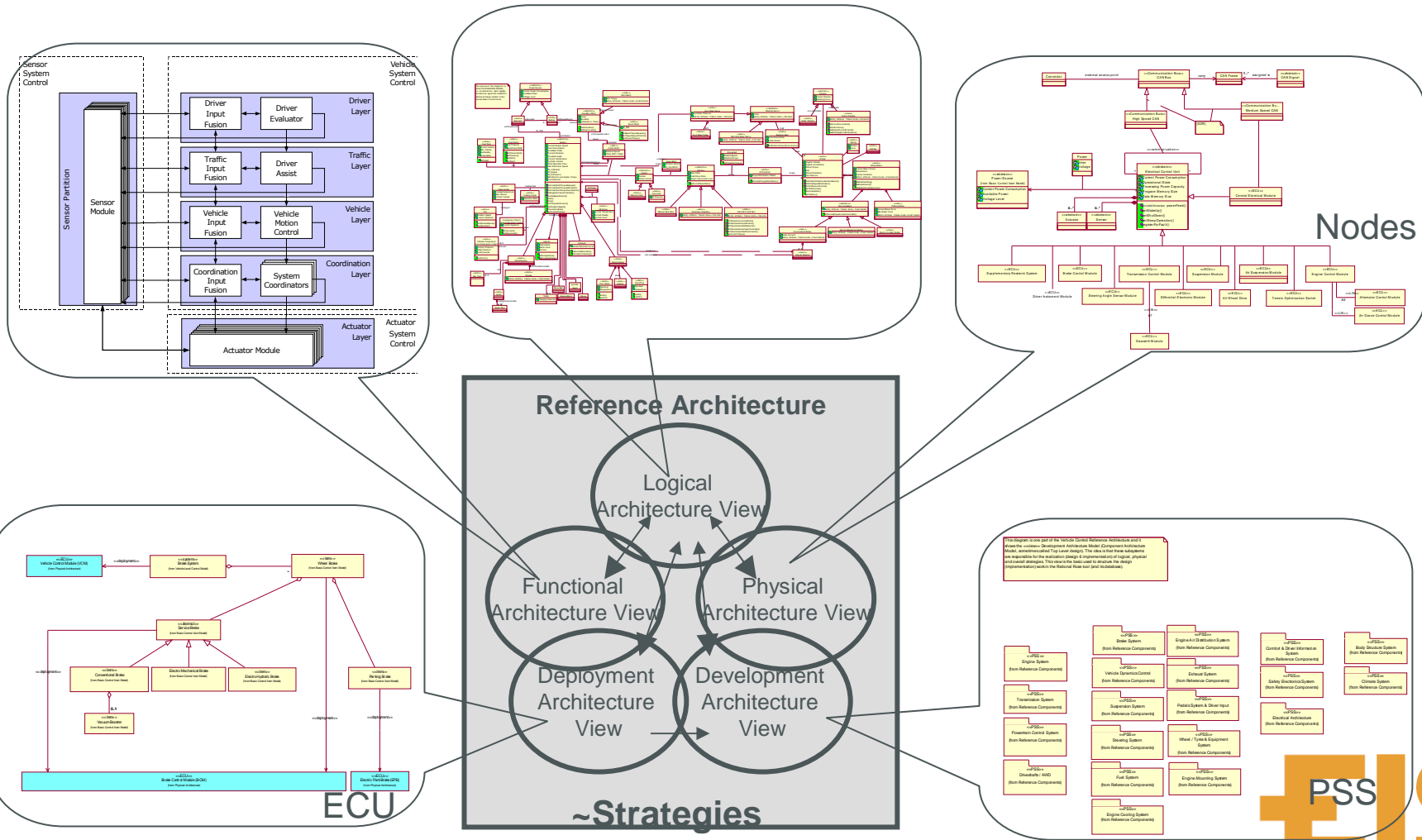
A system architecture is a description of the *structures* of the *components* of a system, their *interrelationships*, and *principles and guidelines* governing *their design and evolution over time*.

# Arkitekturens element

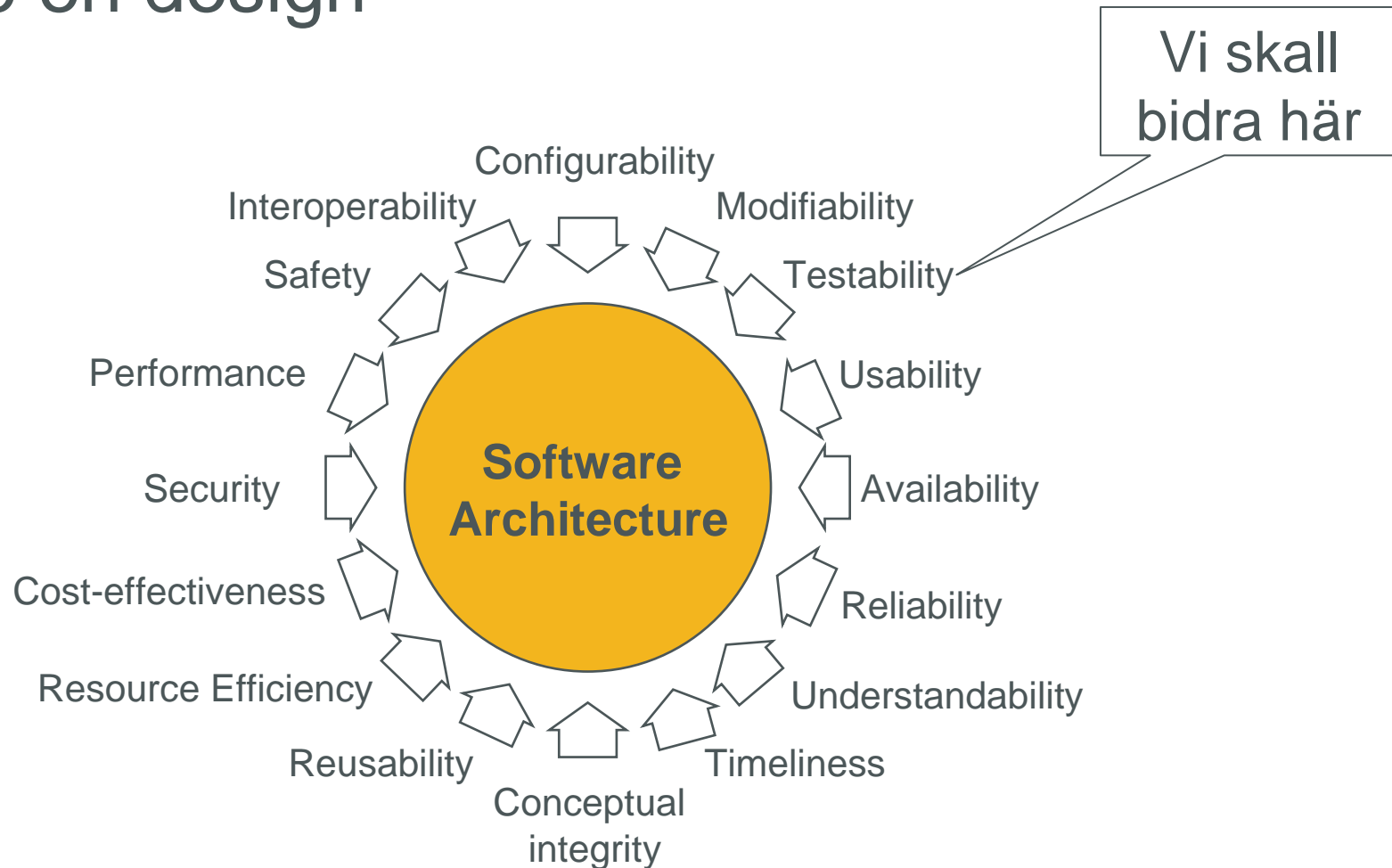


- De Arkitekturella strategierna är kärnan i arkitekturen.
- Systemvyerna beskriver de strukturella- och beteendenaspekterna.
- Designregler tillförsäkrar en sammanhängande och överensstämmande implementation.

# Arkitekturens beskrivning -Vyer



# Typiska egenskaper (attribut) vi vill ge en design





# Påverkan på test

- Exempel på stora värden är
- Gemensamma begrepp
  - När vi bygger upp förståelsen för det system som skall utvecklas och testas är det avgörande att hela projektet använder samma begrepp under hela arbetet
- Minskad komplexitet
  - När vi integrerar delar av systemet är det viktigt att ha få och tydliga beroenden och kopplingar
- Felsökning
  - Förståelsen av ansvar och orsak/verkan när man följer ledtrådar
- Testobjekt och Testmiljö
  - Stöd för identifiering av gränssnitt

# Hur utvärderar vi en arkitektur?

- Vi tittar på den lösning arkitekturen ger

- Discernable under operation =  
Egenskaper som kan utvärderas under exekvering
  - Ex: Performance, Safety, Usability
- Non- discernable under operation =  
Egenskaper som kan utvärderas genom analys och resonemang
  - Ex: Modifiability, Cost-effectiveness
- Exekverbar arkitektur
- Ett avgörande steg i systemutveckling är då vi har exekverbara modeller av systemet
  - Tidig validering av behov
  - Tidig verifiering av logik/beteende
  - Tidig bedömning av aspekter på realiseringen

# Systemmodeller

- Kraven är en modell av systemet
- Arkitekturen innehåller flera modeller
- Designen är en modell
- Ibland har vi andra separat simuleringsmodeller av delar av systemet
- Testspecifikationerna är en modell med input och förväntat resultat

## Konsistens

mellan dessa modeller är avgörande för tydlighet och effektivitet i projektet.

**Har modellerna en livslängd?**

# Strategier för test

(analogt med arkitekturstrategier)

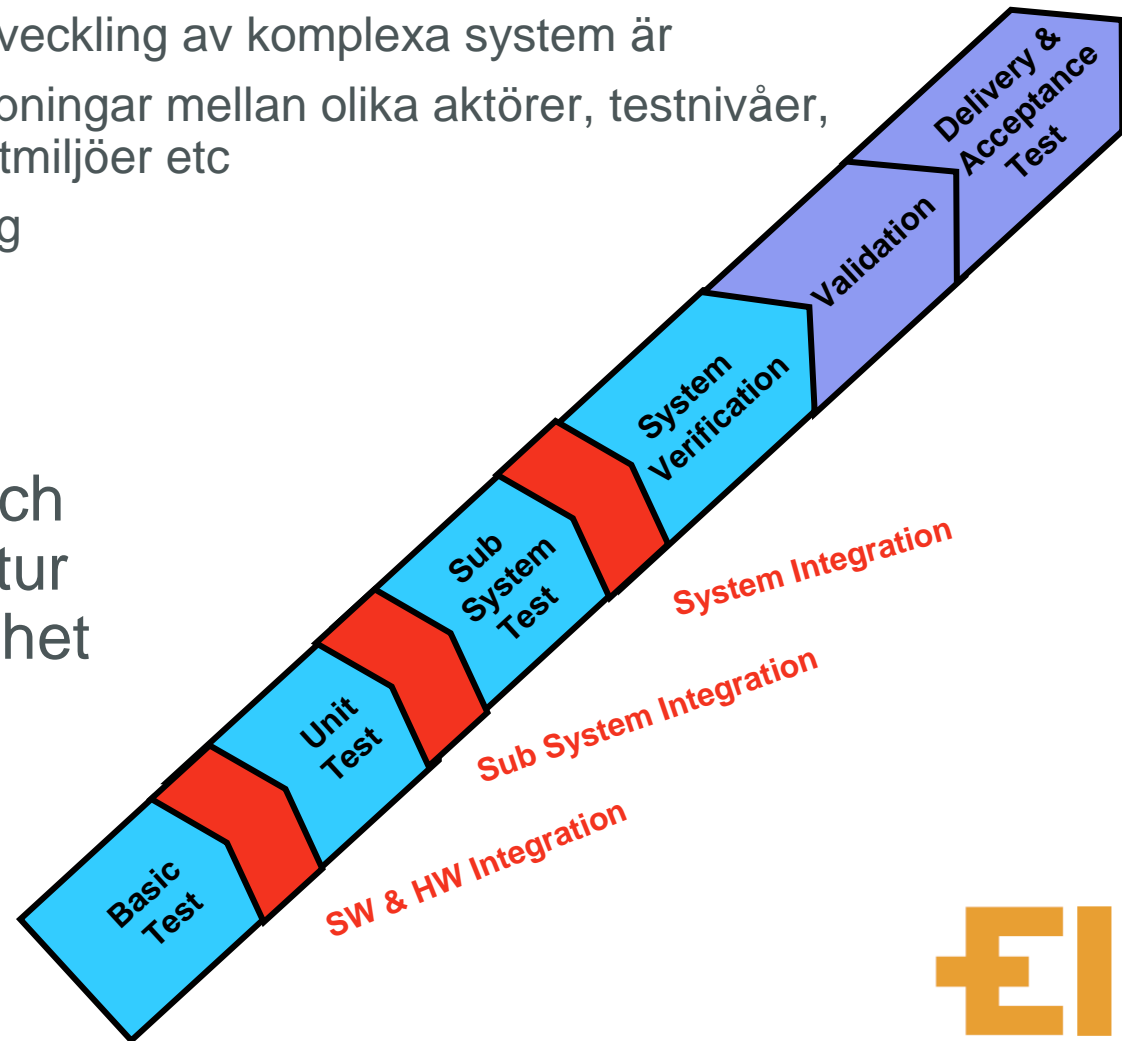
## Exempel

- Vad är det för problem som driver testverksamheten?
- Lösningen formuleras i strategierna
- **Early visual progress by incremental build and verification**
- **High reliability by user like operational profiles in test-rig**
- **Safety issues addressed in safety rig and safety test specification**
- **Maintaining system requirements by regression testing of reused parts**
- **Environments and tools for balancing of availability, drive- ability and safety**

# Modeller för test

- Stora problem inom utveckling av komplexa system är
  - Glipor och överlappningar mellan olika aktörer, testnivåer, organisationer, testmiljöer etc
  - Integrationsstyrning

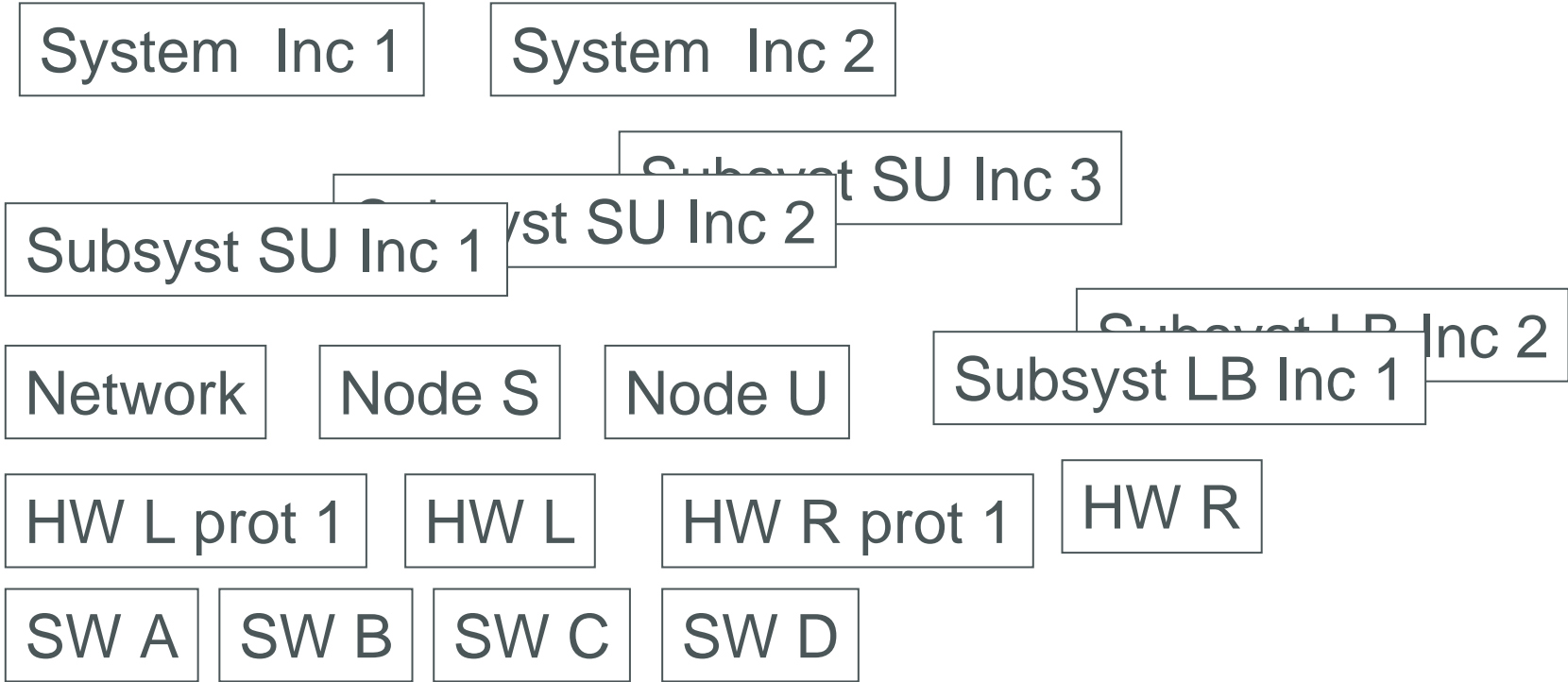
Abstraktionsnivån och  
vyerna inom arkitektur  
har ingen motsvarighet  
inom test idag



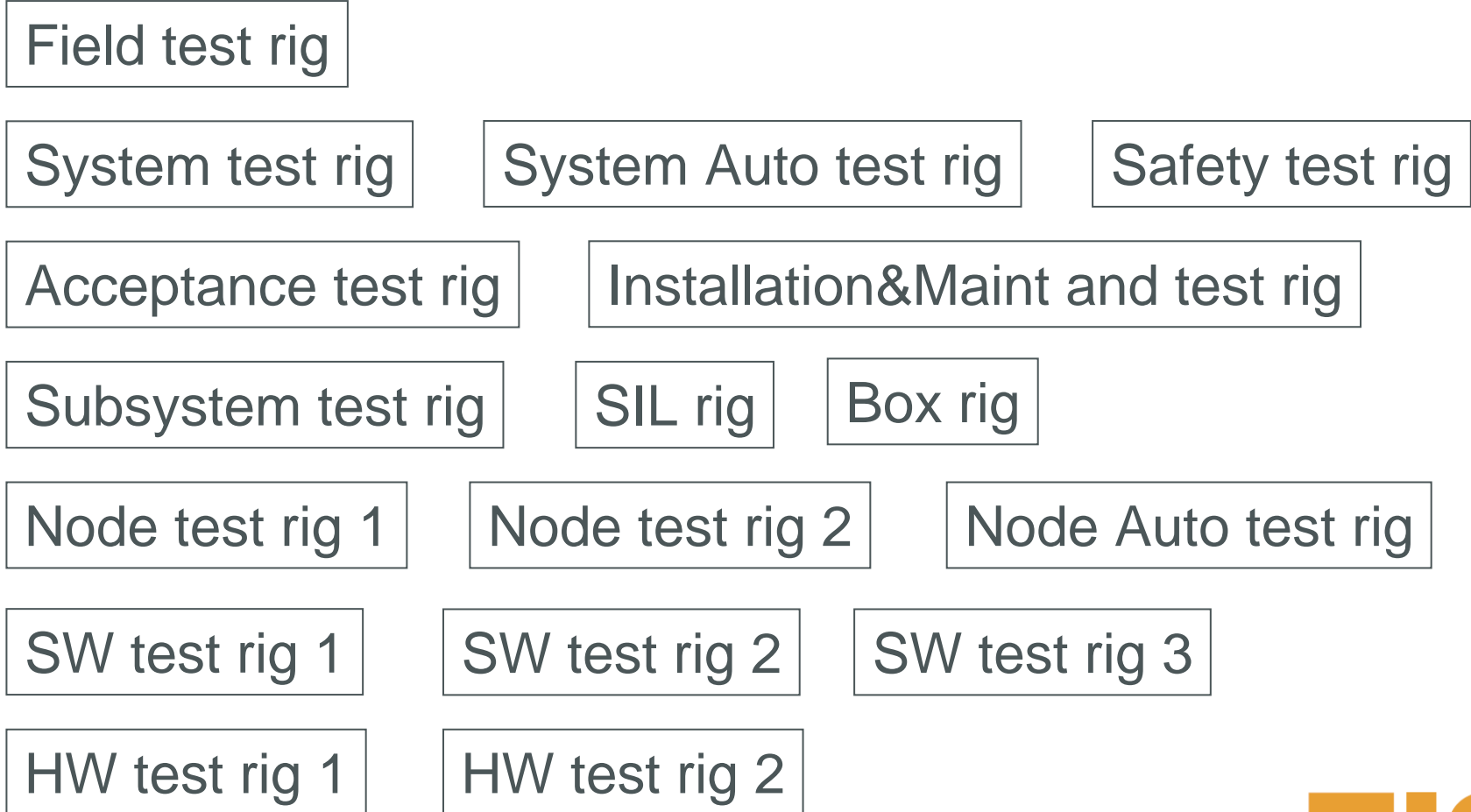
# Förslag till diagram för test

- Testobjekt
  - Testmiljöer
  - Grupperingar av testfall (testkategorier)
  - Organisatoriskt ansvar
- 
- I varje diagram anges objektens relationer och egenskaper

# Diagram av testobjekt

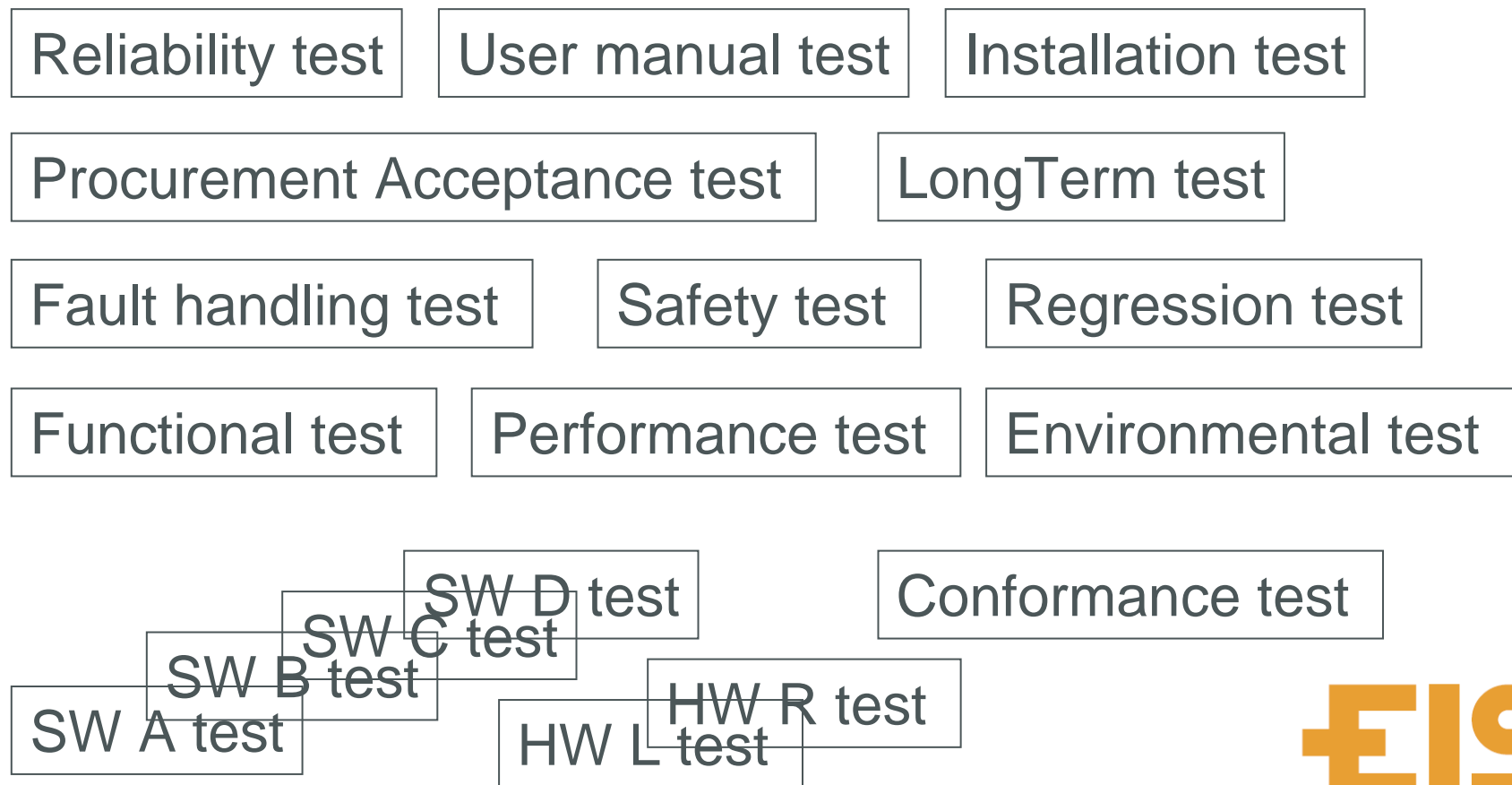


# Diagram av testmiljö

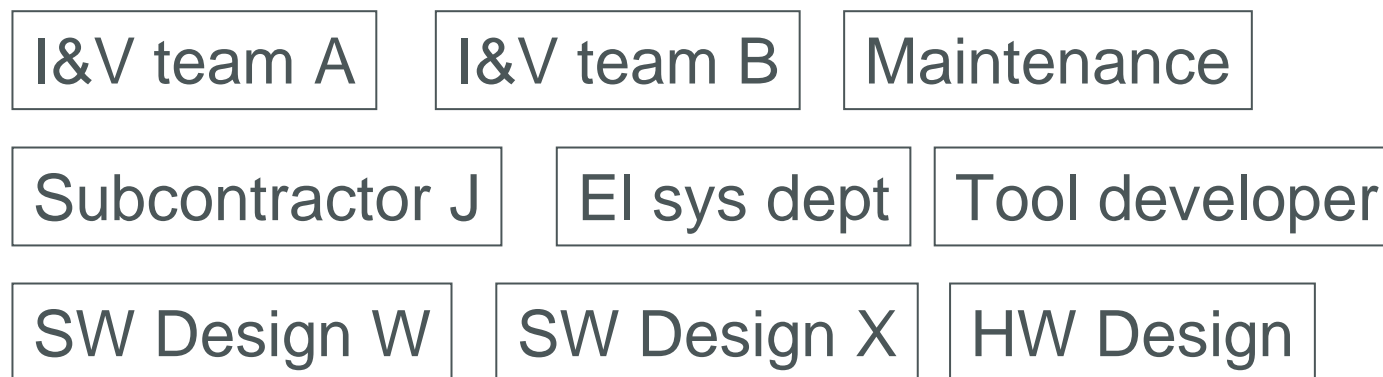




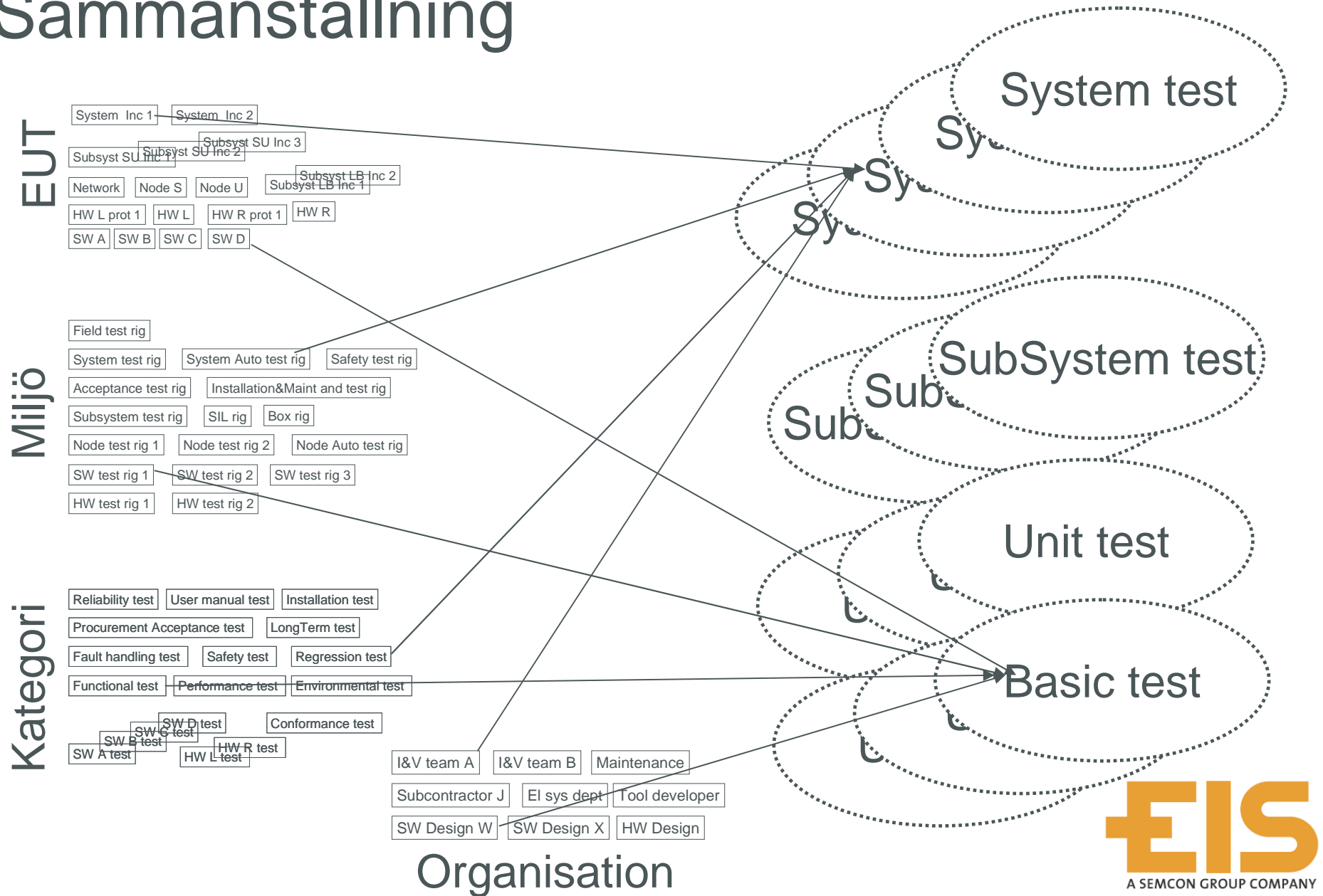
# Diagram av testkategorier



# Diagram av organisatoriskt ansvar



# Sammanställning



# Testbarhet

- Vi har möjligheten att påverka och skall bidra till egenskapen testbarhet
- Efter analys av krav och arkitektur ser vi behov av ändringar, kompletteringar etc för att öka möjligheten till effektiv test

# Systemattribut

- **Observerbarhet:** möjligheten att läsa signaler i systemet
- **Styrbarhet:** möjligheten att injicera (lägga på) signaler för att sätta systemet i ett specifikt tillstånd

# Testbarhet

- **Verksamhetsrelaterat:** Hur smidigt det är att genomföra test
  - Testbara krav.
  - Testbar design. Keep it simple.
  - Möjligheter till automatisering
- **Testtekniskt:** Sannolikheten för att olika fel detekteras under test, felens målarea för en viss testteknik

Organisationen måste  
stämma med arkitekturen  
(eller omvänt)

# Exempel på Testbarhet

## Test av felhantering

- Vi lägger på ett fel på testobjektet under exekvering. Detta benämns ofta fault injection. Felen kan vara relaterade till HW eller SW.
  - Test på hög systemnivå ger bra förståelse för felyttringen
  - Vi behöver då tillgång till interna gränssnitt

Syftet med att ta in  
arkitektur i  
testverksamheten är att  
hantera komplexitet så att  
vi får täckande och effektiv  
test