

# Teknisk testing (???) av kritisk infrastruktur

Alexander Tarlinder

*”Som Teknisk testare blir det ditt ansvar att hjälpa utvecklare (programmerare) att förutse och hitta avvikelser från de uppställda kraven, så att de kan säkerställa att en produkt - exempelvis en programvara eller webbplats - fungerar som tänkt i de system den ska ingå i. Du behöver kunna tillräckligt med programmering för att förstå vad utvecklarna gör.”*

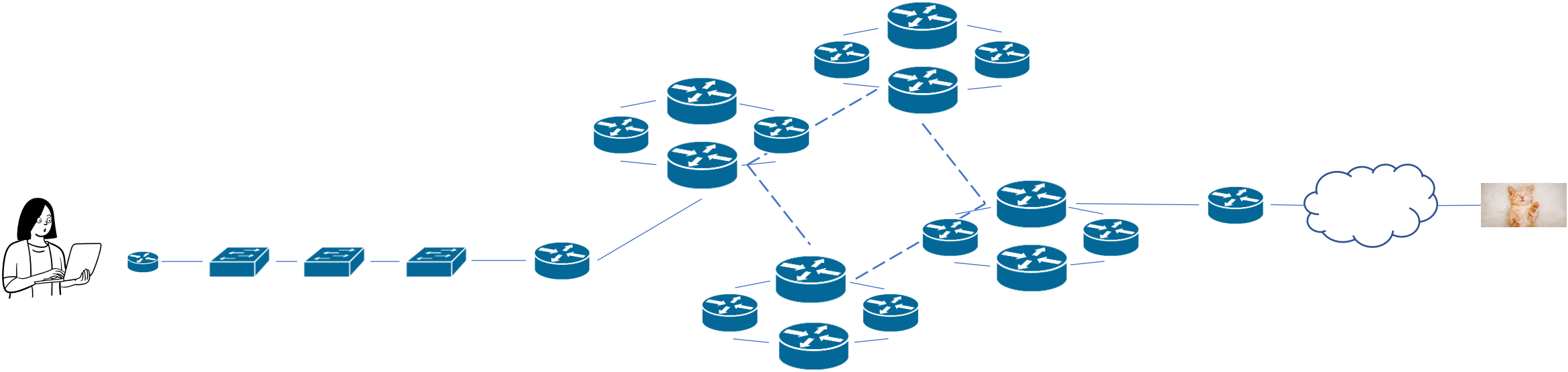
*”Din vardag består av en mix av manuella tester och automatisering med något av de etablerade verktygen på marknaden såsom Postman, SoapUI, ReadyAPI, Selenium. När det gäller manuella tester har du erfarenhet av att jobba både med att exekvera testfall och utforskande test. Inom automatisering är det inte nödvändigtvis du som satt upp ramverket, men du har skaffat dig professionell erfarenhet och jobbat med t.ex. validering.”*

# Teknisk testning ['tɛknɪsk 'tɛstnɪŋ]

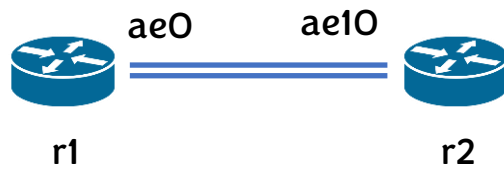
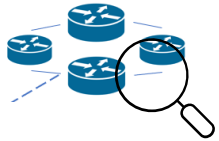
Auser testning av:

- Något tekniskt
- Något som kräver teknisk (special-)kunskap

# Vägen till kattbilden



# Konfiguration av nätelement



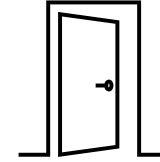
```
interfaces {
  interface ae0 {
    description r2.ae10; ← Affärsregel
    mtu 9192;
    aggregated-ether-options {
      no-flow-control;
      minimum-links 1; ← Räknas fram
      lacp {
        active;
        periodic slow;
        force-up;
      }
    }
  }
  unit 0 {
    family {
      inet {
        address 10.0.0.0/31; ← Allokeras nästans
      }
      mpls {
        maximum-labels 5;
      }
    }
  }
}
...

```

# Bra att ha



- Routing + MPLS + VPN
- Säkerhet
- Operativsystem för nätelement
- Specifika byggregler för nätet
- Andra affärsregler
- Historik



- Hårdvara
- Möjlighet att bygga/koppla om



“Buggar”

Felaktiga API-anrop



“Buggar”

Felaktig skrivning/läsning



Felaktiga API-anrop

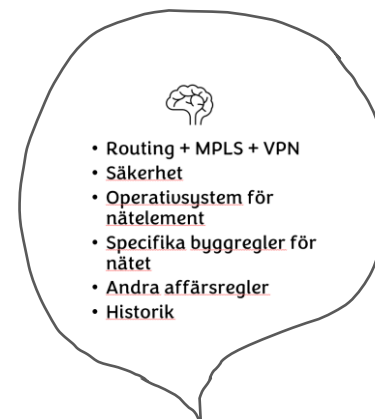


Felaktig konfiguration

Version X av drivern stödjer inte parameter Y  
“Buggar” i drivern



Konfiguration enligt spec, men det blev fel ändå  
Version X av operativsystemet stödjer inte Y



- Routing + MPLS + VPN
- Säkerhet
- Operativsystem för nätelement
- Specifika byggregler för nätet
- Andra affärsregler
- Historik



- Hårdvara
- Möjlighet att bygga/koppla om

# Kvalitetsstrategin

- Stark typing
- Hexagonal arkitektur
- Utvecklastning
- Tidig validering
- Simulering – virtualisering – labb
- (Scrum)



# Väldigt få saker är strängar eller tal

IP-adresser

Nätmasker

OSPF-areor

VLAN

Interface units

Interface-namn

Kundnummer

Hastigheter

Olika typer av id:n

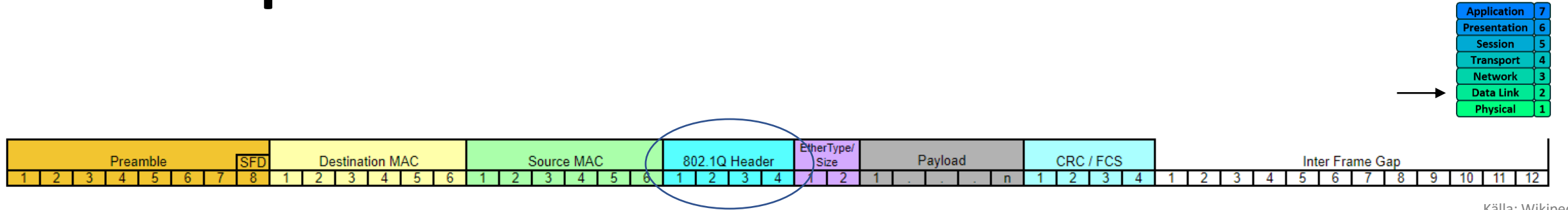
Primärnycklar

Tillstånd

Produktfamiljer

Kanaler

# Exempel: VLAN-id:n



Källa: Wikipedia

12 bitar = 0-4095  
0 = Ingen taggning  
1, 4095 = speciella  
2-4094 kvar

```
class VlanId:
    vlan_id: int

    def __init__(self, vlan_id: int):
        if vlan_id < 2 or vlan_id > 4094:
            raise ValueError(f"{vlan_id} isn't a valid VLAN id. 2-4094 are.")
        self.vlan_id = vlan_id

    def __int__(self) -> int:
        return self.vlan_id
```

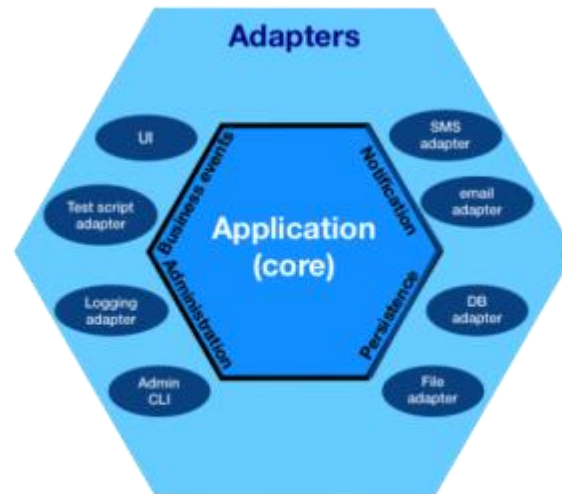
Oändligt stor

2-4094

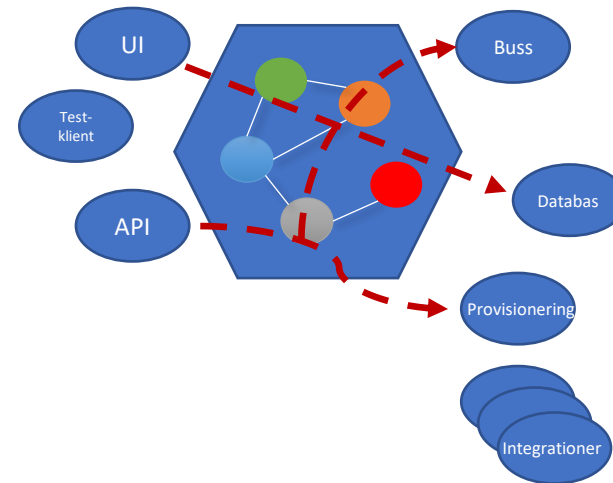
# Väldigt få saker är strängar eller tal

IP-adresser	<code>^(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\$</code>
Nätmasker	<code>/0 - /32</code>
OSPF-areor	Samma som IP-adresser
VLAN	<code>[2,4094]</code>
Interface units	<code>[0,16384]</code>
Interface-namn	Krångligt...
Kundnummer	<code>[80000000,90000000)</code>
Hastigheter	<code>{10, 40, 100, 400}</code>
Olika typer av id:n	<code>&gt;0</code> , mindre än någonting
Primärnycklar	<code>&gt;0</code>
Tillstånd	Ex: <code>{planned, provisioned}</code>
Produktfamiljer	<code>{mx, ptx}</code>
Kanaler	<code>[0,4]</code>

# Hexagonal arkitektur

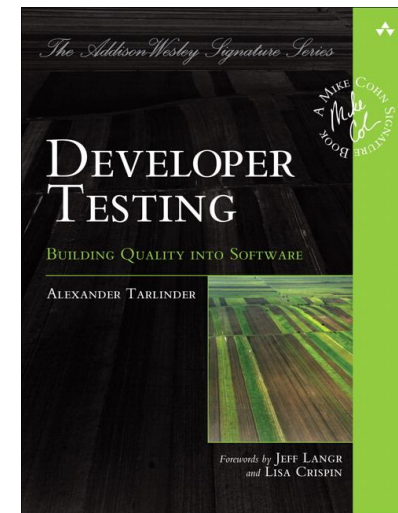
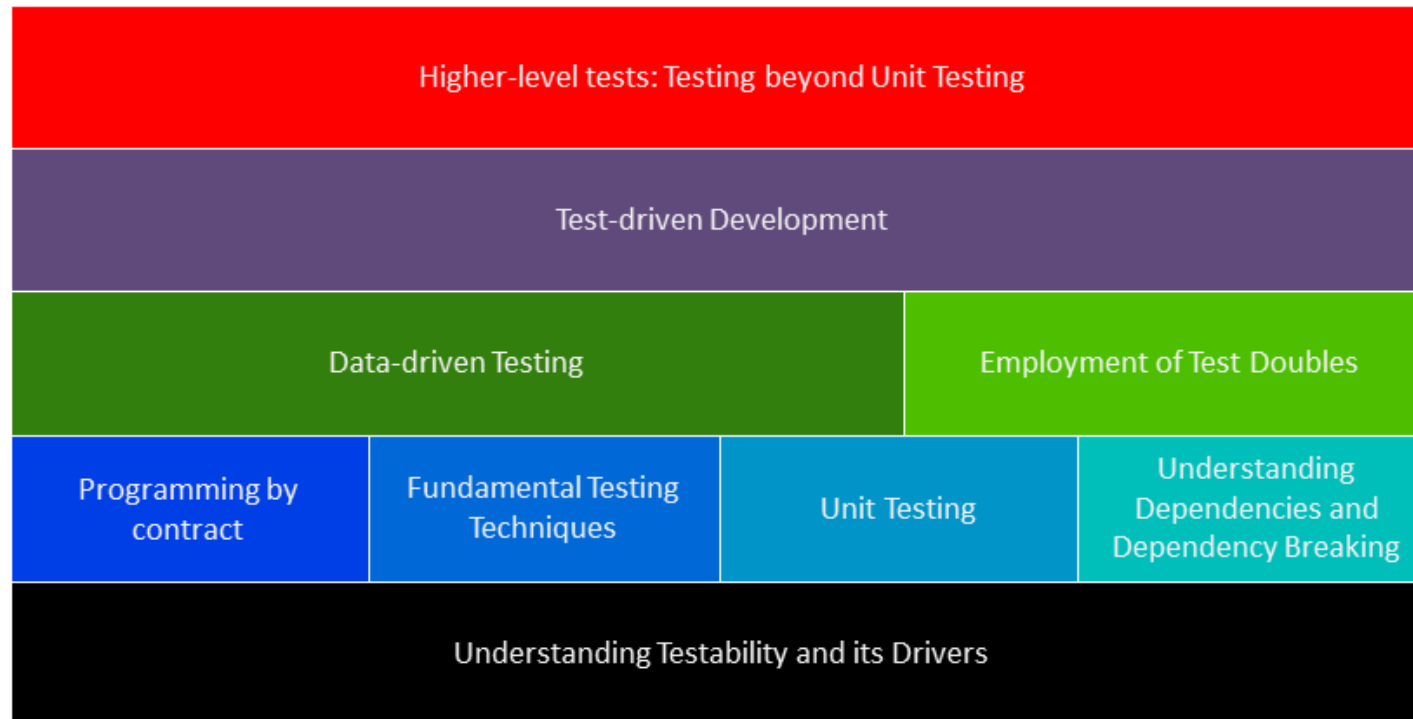


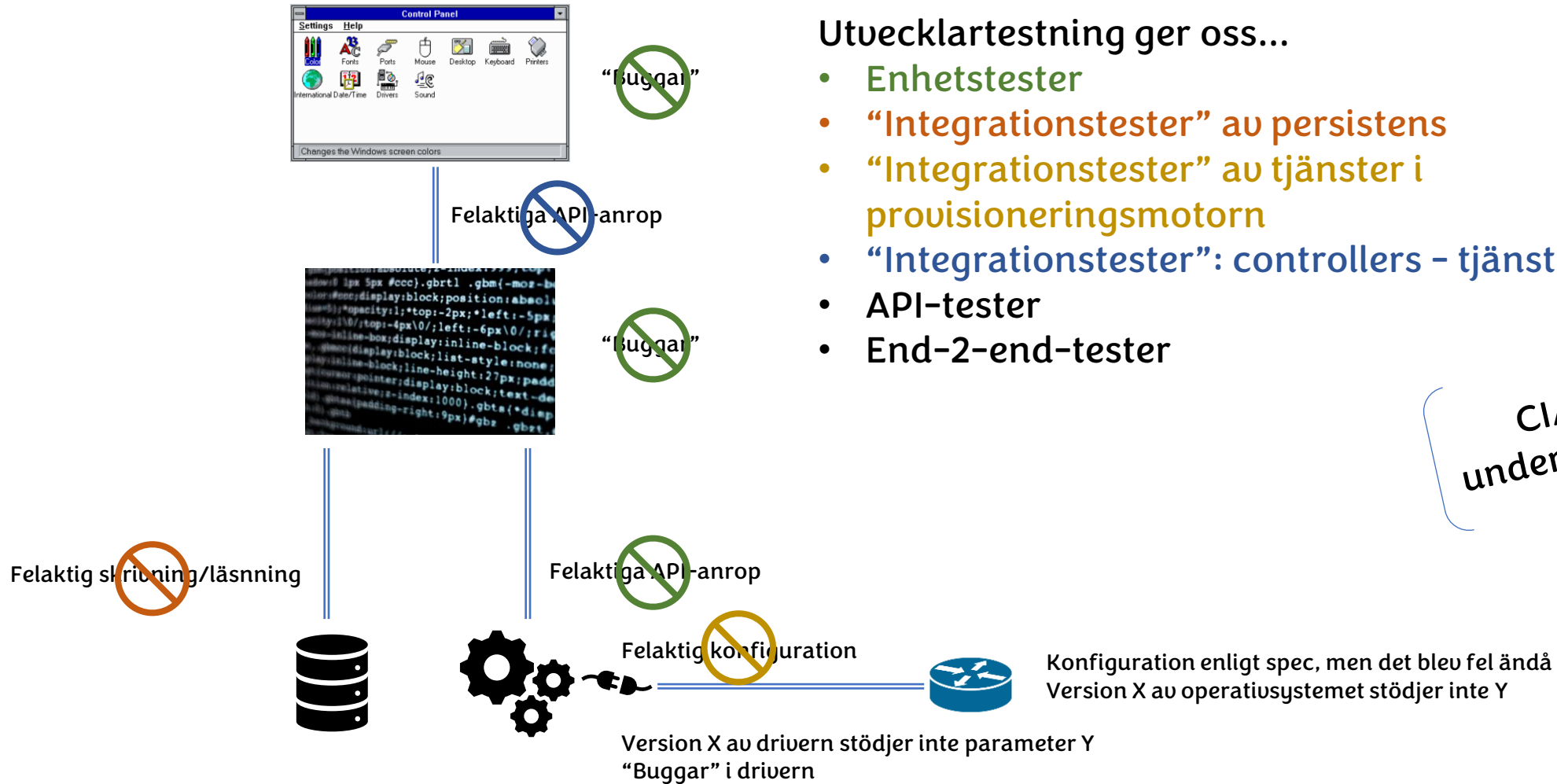
Källa: Wikipedia



# Utvecklar testning

*Developer testing* is the developers' **intentional** and **systematic** employment of testing tools and techniques to create testable and maintainable software with as few defects as possible.





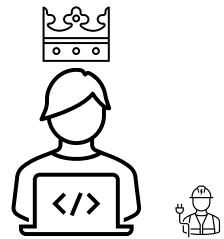
## Utvecklartestning ger oss...

- Enhetstester
- "Integrationstester" av persistens
- "Integrationstester" av tjänster i provisioneringsmotorn
- "Integrationstester": controllers - tjänstelager
- API-tester
- End-2-end-tester

# Tidig validering

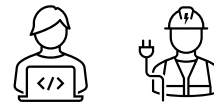
- Tidig feedback
- Billig feedback
- “Validering” på istqbiska

# Simulering – virtualisering – labb



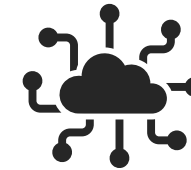
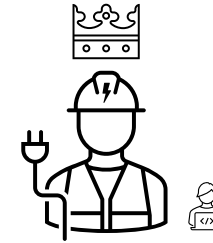
## Simulerad miljö

Provisionerings-  
produkten



## Virtualiserad miljö

Ansible  
Vmware  
ZTP  
Virtualiserade nätelement



## Fysisk miljö

Fysisk rackning  
Domänkunskap inom nät



# Bonusmaterial: Scrum

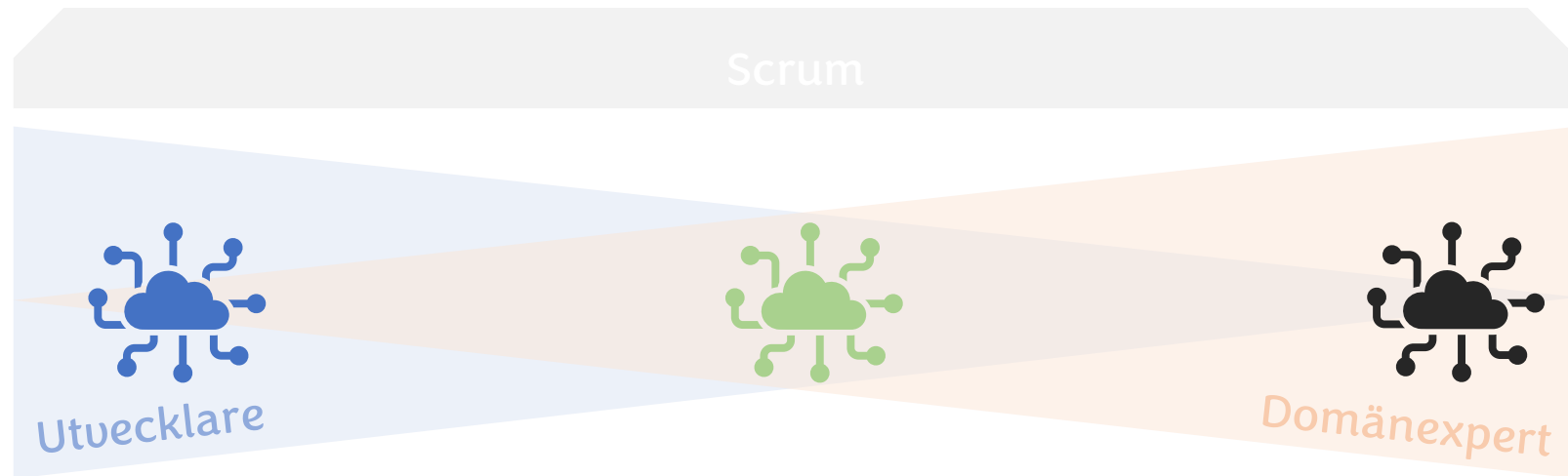
- Alla är “utvecklare”
- Definition of Done
- Sprint review



# Sammanfattning

Mål

- Förhindra alla “buggar”
- Hitta fel “krau”
- Hitta fel “i fältet”



- ✓ Stark typing
- ✓ Hexagonal arkitektur
- ✓ Utvecklartestning
- ✓ Tidig validering