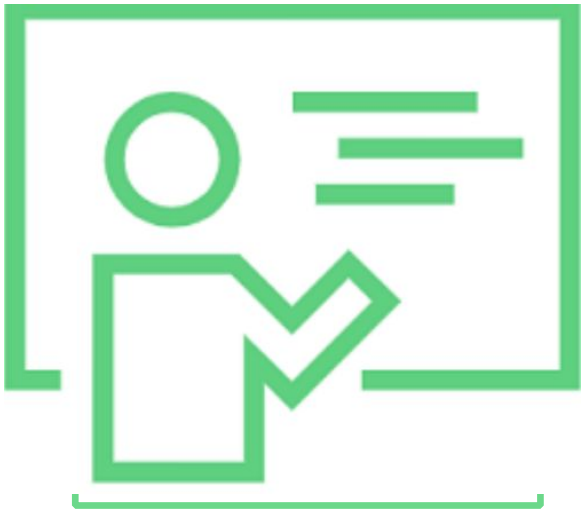


Integrerad systemtest och teknisk kommunikation

Joakim Englund, Systementor AB

Repetition från dag 12

- Källa -> krav -> kod -> test -> bevis -> drift
- Testnamn är inte samma sak som testbevis
- Dokumentation ska göra arbetet reproducerbart
- En känd begränsning ska vara synlig

Dagens mål

- Testa ett helt användbart systemflöde
- Skilja observation, hypotes och slutsats
- Ge konkret och respektfull kamratrespons
- Förklara tekniska beslut på svenska och engelska
- Förbereda en kort systemgenomgång

Fyra testnivåer

Nivå	***Fokus***
Enhet	En liten kodsnutt, t.ex. en funktion
Komponent	En tjänst
Integration	Komponentgränser
System	Ett helt användbart flöde

Vad gör systemtest annorlunda?

Sensor → Nät → Gateway → Broker/API → Konsument

- Fler verkliga gränser
- Mer realistisk konfiguration
- Observerbart resultat utifrån

Börja med gränserna

- Vad ingår?
- Vad simuleras?
- Vilka externa beroenden finns?
- Vilken fysisk eller logisk enhet kör varje komponent?
- Var observeras resultatet?

Ett reproducerbart testfall

- Krav-ID och test-ID
- Miljö och starttillstånd
- Testdata och steg
- Förväntat och faktiskt resultat
- Bevis och återställning

Testorakel

Egentligen bara ett annat namn för förväntade resultat från tester.

Observerat resultat + tydlig regel -> PASS eller FAIL

- En regel måste kunna avgöra resultatet
- Testoraklet kan vara kontrakt, krav, referensvärde, oberoende mätning, eller om det gäller något mer subjektivt - testaren själv.

Positiva och negativa tester

- Normal data genom hela flödet - “Happy path”
- Ogiltig data avvisas
- Beroende otillgängligt
- Timeout och återanslutning
- Behörighet och certifikatfel

Felinjektion med kontroll

1. Verifiera baslinjen
2. Injicera ett känt fel
3. Observera symptom och loggar
4. Återställ
5. Kör om baslinjen - verifiera att det fortfarande fungerar som det ska

Observation vs hypotes

Typ	***Exempel***
Observation	Klienten fick HTTP 503
Hypotes	API:t är överbelastat
Test	Kontrollera serverlogg och resursmått
Slutsats	Accepteras först efter stöd

Korrelations-ID följer flödet

test_id=SYS-004
request_id=demo-42

- Samma ID i klient, gateway och API
- Logga inte token eller privata nycklar

Demo

- Terminal 1: Gateway/API på logisk gateway-enhet
- Terminal 2: Systemtest på test-/konsumentdator

Demo: verifiera baslinjen

```
PASS health endpoint
PASS valid reading contract
PASS invalid reading is rejected
PASS unavailable dependency is visible
RESULT: PASS (4/4 tests)
```

Demo: injicera fel

```
server --fault missing-unit
…
FAIL valid reading contract
```

- HTTP 200 men kontraktet är brutet
- Grundorsaken syns vid komponentgränsen

Felsökningskedjan

Symptom → Observation → Hypotes → Test → Orsak → Åtgärd → Omtest

Strukturen gör resonemanget betydligt enklare att granska.

Konstruktiv kamratrespons

- Observation: Vad såg jag?
- Konsekvens: Vilken inverkan har observationen och varför spelar det roll?
- Förslag eller fråga: Vad kan ändras?
- Undvik personpåhopp.
- Undvik vaga kommentarer t.ex. “gör bättre/tydligare”

Prioritera återkopplingen

1. Säkerhet och dataförlust
2. Felaktigt eller ej reproducerbart resultat
3. Otydliga kontrakt och driftsteg
4. Läsbarhet och presentation

Systemgenomgångens röda tråd

- Problem och systemgräns
- Arkitektur och dataflöde
- Säkerhets- eller robusthetsbeslut
- Testbevis och felsökningsspår
- Begränsning och nästa steg

Exempel teknisk engelska

- System purpose
- Integration decision
- Verified test result
- Known limitation

Ex: “The gateway rejects readings without a unit.”

Laborationens definition of done

- Hela normalflödet är testat
- Minst två relevanta felvägar är observerade
- Resultaten är kopplade till krav och bevis
- Kamratrespons är prioriterad och beslutad
- Systemgenomgången har genomförts på både svenska och engelska