

Prestandatest Processen

- Effektiv prestandatestning, teknisk verifiering, tuning, verifiera krav, förvalta prestanda

Intro

Christian Gerdes

Performance Engineering Evangelist

LIGHTS IN LINE AB

christian@lightsinline.se

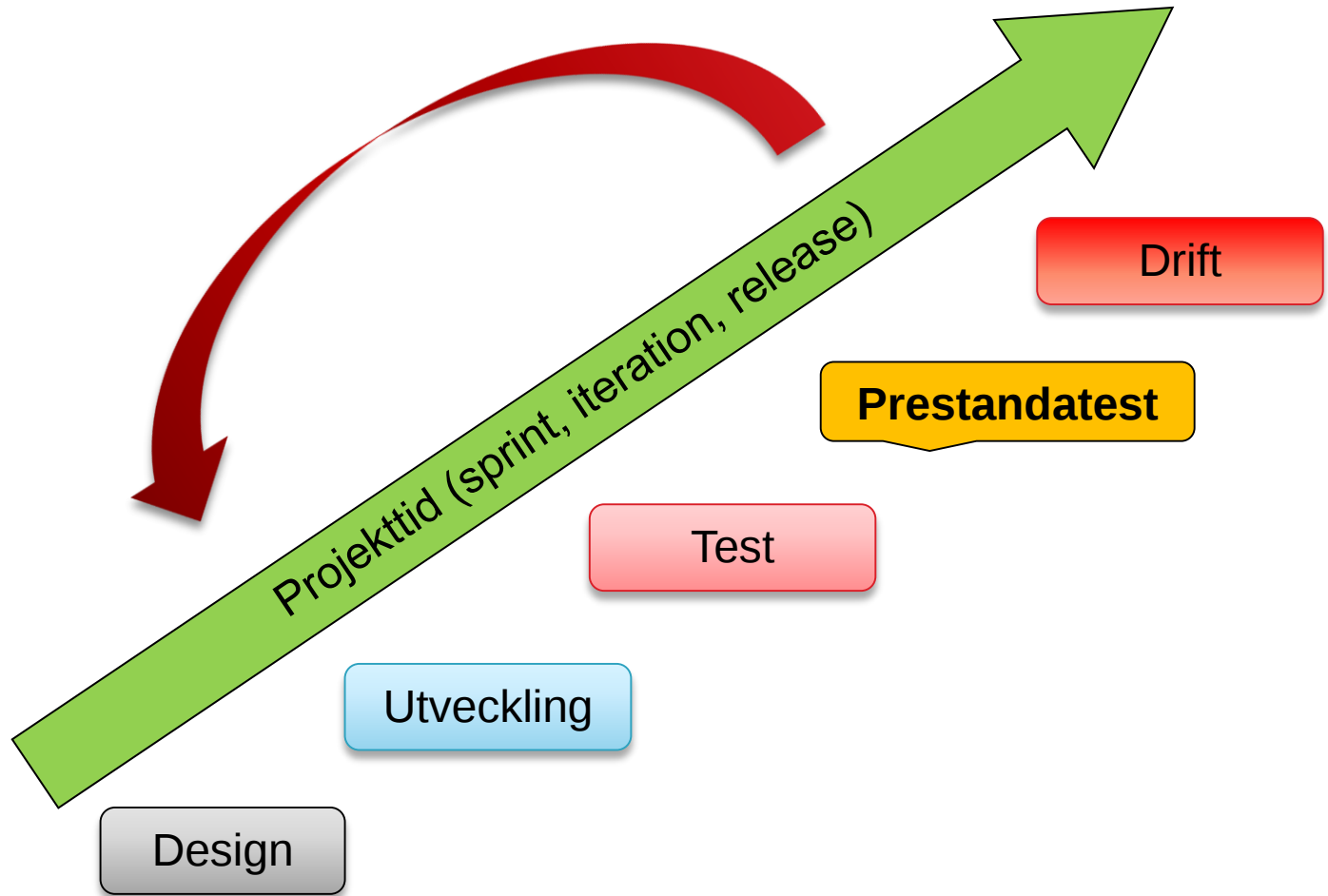
Effektiv prestandatestning

»» Varför misslyckas så många
prestandatester?

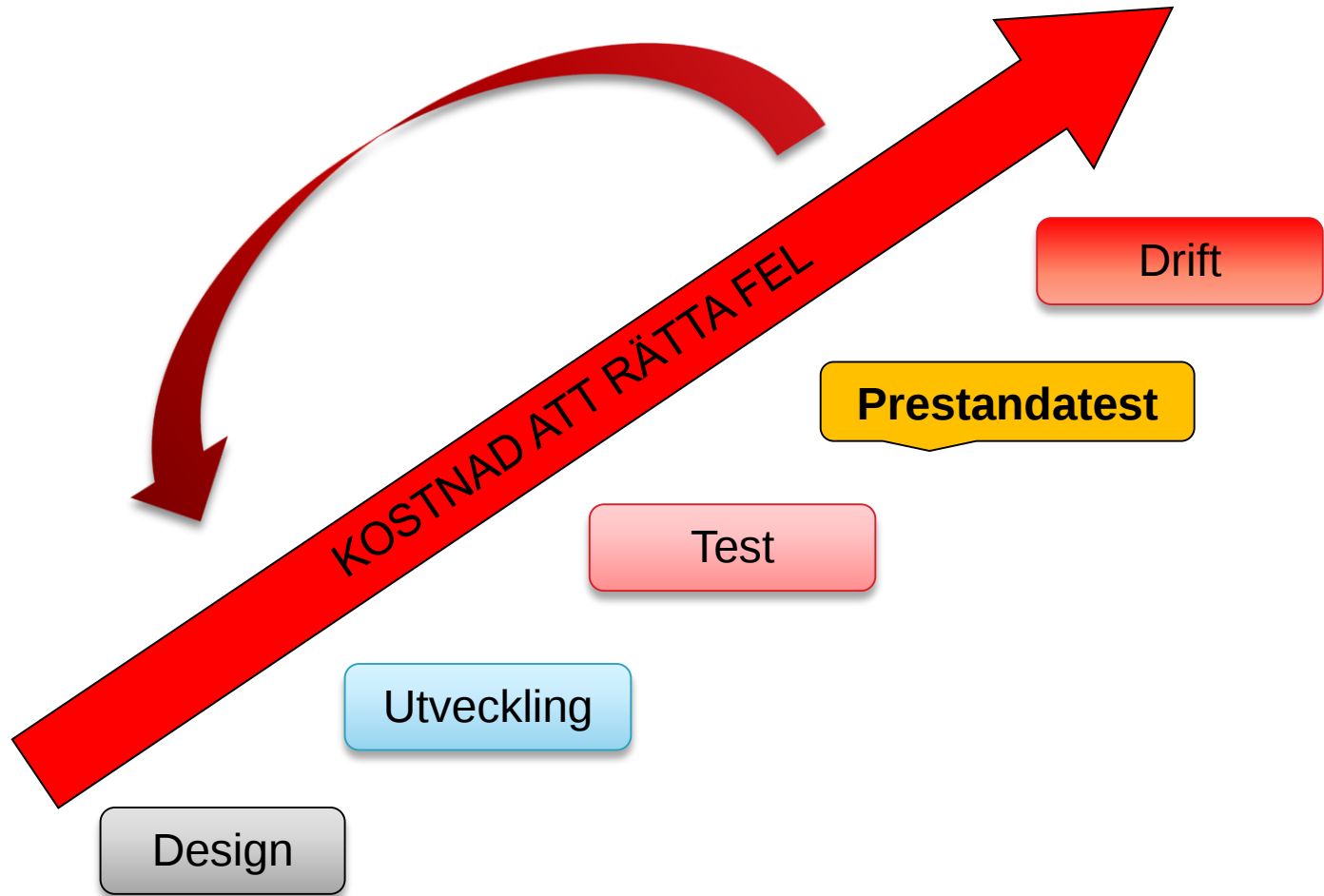
Traditionell prestandatest

- ▶ Målet är att efterlikna verkligheten
 - ▶ Försöker få med så många UC som möjligt
 - ▶ AUT ska vara färdig, funktionell och bugfri
 - ▶ Isolerad produktionslik miljö
 - ▶ Produktionslika datamängder
 - ▶ Vi måste mäta allt, HW som SW
-
- ▶ Pga detta görs det sist av allt och tar månader av förarbete...

Traditionell prestandatest



Traditionell prestandatest



Lyckat prestandatest?

- ▶ Traditionella processen försöker ha dubbla mål:
 - 1: Verifiera prestandakraven
 - Trots att man inte vet vilken miljö som krävs
 - 2: Identifiera prestandabuggar
 - Trots att det inte finns tid att rätta dem
 - Om de hittas hindrar de ofta mål 1 ovan
 - För att man måste akuträtta istället

Många misslyckas...

- ▶ Man hinner inte med alla planerade tester
 - Trots scopning
- ▶ Tidsödande felsökningar
 - Lika komplext som i produktion
- ▶ Höga kompetenskrav
 - Svårt att förstå sig på resultaten
 - Tusentals mätpunkter
- ▶ Blir ofta väldigt dyrt...

Effektiv prestandatestning

1. Teknisk Verifiering

- Förmår systemet/arkitekturen leverera hög prestanda och tillgänglighet?
- Identifiera prestandabuggar och lös dem

2. Optimering och tuning

- Iterationer för att optimera prestanda
- Identifiera kapacitetsbehov / dimensionera

3. Verifiera mot kraven i tilltänkt miljö

- Klarar vi av förväntade produktionsförhållanden?

Scopning

» Vi kommer inte hinna testa
allting... vad ska vi då ha med?

Varför scopning?

- ▶ Exempel: En internetbank
- ▶ Har över 2000 Use Cases
 - Bara att kartlägga flöden/mixar, testautomatisera dessa samt säkerställa rätt testdata skulle ta ca 1 år

Scopning: 3 tumregler

▶ 1: Volym

- De flesta applikationer har endast ett fåtal UC som står för majoriteten av trafiken/volymen.
- 90% av volymen under ett test är ett bra test

▶ 2: Tekniskt komplext

- Vissa UC hanterar mycket data, kostar mycket CPU eller låser ut andra UC. Dessa kan lätt skapa prestandaproblem trots lägre volym.

▶ 3: Affärskritiskt

- Mycket badwill om de inte har bra prestanda, typ konkurrens, marknadsföring, kampanjer

Scopning: Exempel

- ▶ Exempel: En internetbank
- ▶ Har över 2000 Use Cases
 - 50 av dessa står för 92% av volymen under månadens peak timme i drift
 - 10 tal är tekniskt komplexa
 - 10 tal är ytterst affärskritiska

Teknisk Verifiering

» 80% av jobbet, för 20% av
kostnaden

Teknisk Verifiering

- ▶ Förmår systemet/arkitekturen leverera hög prestanda och tillgänglighet?
 - Verifiera att du har "triss i äss" med 3S Metoden
 - Lös prestandabuggarna medans det finns tid
 - Utforska och lär känna systemets egenskaper/begränsningar. Var är dina flaskhalsar!
 - Kan göras tidigt och ofta, lätta att jämföra
 - Verifiera lastdelning, failover och identifiera SPOFar

3S Metoden

- ▶ Nästan alla prestandaproblem beror på något av dessa 3 förutsättningarna inte uppfylls:
 - 1: Samtidighet
 - 2: Skalbarhet
 - 3: Stabilitet
- ▶ Genom att verifiera tidigt och för individuella UC
 - Hinner vi lösa dem i tid
 - Minimerar vi risken att få dyra problem senare
 - Snabbt, tidigt, ofta

1S – Samtidighet

- ▶ Antagande:
 - För att kunna skala måste systemet förmå att hantera flera parallella / samtidiga anrop
- ▶ Vad kan hindra detta?
 - ???

1S – Samtidighet

▶ Antagande:

- För att kunna skala måste systemet förmå att hantera flera parallella / samtidiga anrop

▶ Vad kan hindra detta?

- Trådsäkerhet! Kolla dokumentationen av api:er!
- Delade resurser som skrivs över, används samtidigt av flera, sessionsobjekt blandas ihop
- Deadlocks

2S – Skalbarhet

▶ Antagande:

- Det som begränsar systemets genomströmning är dess CPU kapacitet
- Mer CPU eller fler servrar kommer att skala ut systemets kapacitet

▶ Vad kan hindra detta?

- ???

2S – Skalbarhet

▶ Antagande:

- Det som begränsar systemets genomströmning är dess CPU kapacitet
- Mer CPU eller fler servrar kommer att skala ut systemets kapacitet

▶ Vad kan hindra detta?

- Flaskhalsar där man inte förväntar sig dem (dvs inte där vi kan skala enkelt) tex nätverk, databas, disk
- Låsningar av delade resurser och applikatoriska köer (konfiguration av pooler, timeouter mm)

3S – Stabilitet

- ▶ Antagande:
 - Kapaciteten och prestandan är konstant över tiden, dvs det blir inte sämre efter längre tids användande
- ▶ Vad kan hindra detta?
 - ???

3S – Stabilitet

- ▶ Antagande:

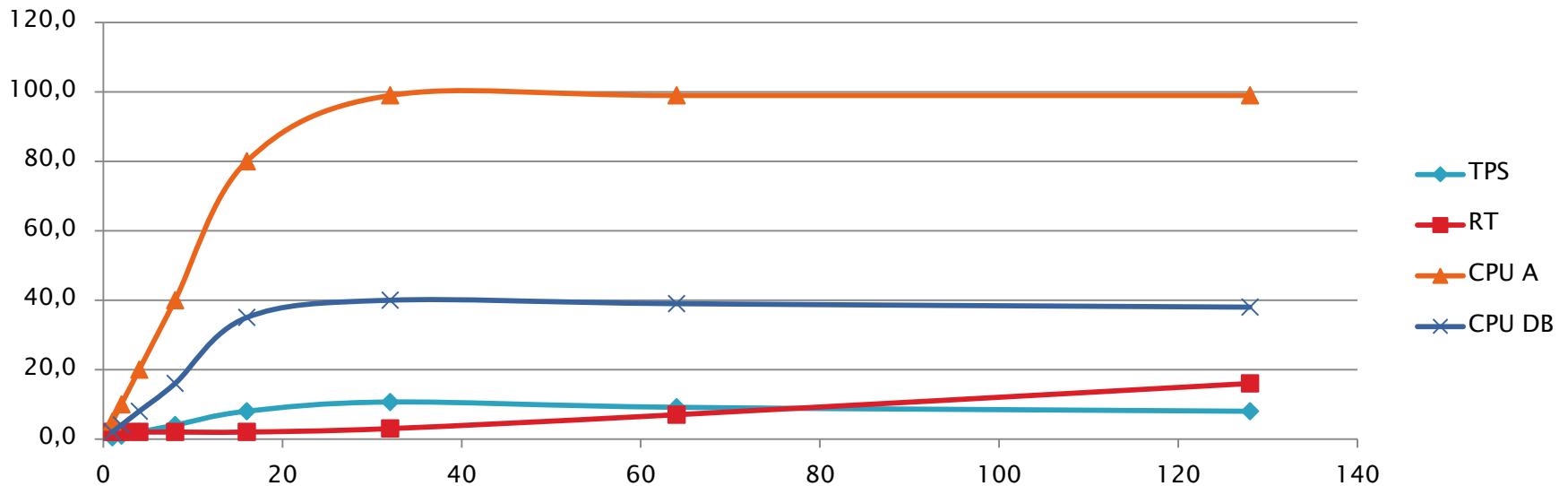
- Kapaciteten och prestandan är konstant över tiden, dvs det blir inte sämre efter längre tids användande

- ▶ Vad kan hindra detta?

- Resursläckage av olika slag, minnesläckage, trådar och delade objekt lämnas inte tillbaka, ökande datamängder utan rensning eller saknade index mm.

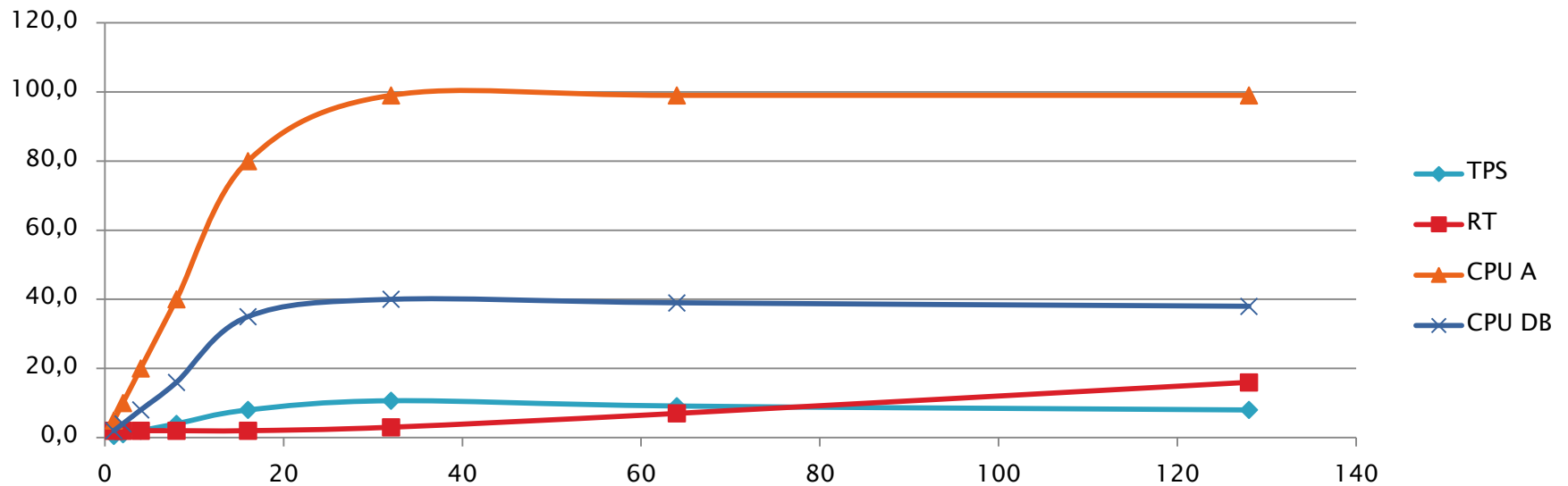
Exempel: Samtidighet och Skalbarhet

#VU	TPS	RT	CPU A	CPU DB	% FEL
1	0,5	2	5	2	0
2	1,0	2	10	4	0
4	2,0	2	20	8	0
8	4,0	2	40	16	0
16	8,0	2	80	35	0
32	10,7	3	99	40	0
64	9,1	7	99	39	0
128	8,0	16	99	38	0



Prestanda Index

PI	#VU	TPS	RT	CPU A	CPU DB
0,3	1	0,5	2	5	2
0,5	2	1,0	2	10	4
1,0	4	2,0	2	20	8
2,0	8	4,0	2	40	16
4,0	16	8,0	2	80	35
3,6	32	10,7	3	99	40
1,3	64	9,1	7	99	39
0,5	128	8,0	16	99	38



Mix Tester

- ▶ Efter att UC'n testats för sig är det dags att blanda
- ▶ Godkända UC hamnar i mixtester
- ▶ Börja vikta de olika UC'n efter mer produktionslika peakar ("värsta timmen")
- ▶ Mät i produktion eller kör beteendeanalys
 - Fördel att mäta på samma sätt i produktion samt i testmiljöer för att enkelt kunna jämföra framöver
- ▶ Google Analytics Data Sharing Program

Optimering och tuning

»» A never ending story...

Optimering och tuning

- ▶ Iterationer för att optimera prestanda
 - Vilka konfigurationer ger bästa resultat
 - Tex:
 - Utvärdera olika antal och typer av index i databasen
 - Utvärdera kombinationer av app pooler, worker threads
 - Minneshantering, heap tuning
 - Indikationer på hårdvarubehov / kapacitetsplanering
 - Underlag för hur produktionsmiljön ska dimensioneras/konfigureras

Scoping!

- ▶ Bestäm vad som är tillräckligt bra
 - Annars blir man aldrig klar...
- ▶ Prioritera det som ger mest effekt
 - Baserat på resultat från 3S testerna
- ▶ Ändra aldrig testerna eller mixen
 - Ändra endast konfigurationen eller infrastrukturen
- ▶ Ändra 1 sak i taget!
 - Annars vet du inte vad som påverkade vad.
- ▶ Kör på så länge det finns tid
 - Men ju bättre optimerat, ju lägre driftkostnader!

Verifiera kraven i rätt dimensionerad miljö

» Slutet gott, allting gott!

Verifiera mot kraven

- ▶ Klarar vi av förväntade produktionsförhållanden?
 - Verifiera tilltänkt produktionsmiljö
 - Täck ungefär 90% av volymen
 - Men inkludera även tekniskt tunga eller affärskritiska delar
 - Låg risk att fastna i tekniska problem och felsökningar (3S!)
 - Fokusera på verksamhetens krav och produktionslika tester
 - Kunskap om produktionsmiljöns gränser
 - Slutligt kvitto på att läget är gott

Releasetester (förvaltning)

- ▶ Varje slutligt prestandatest blir en baseline inför nästa release
- ▶ Nya UC tas in via
 - Först 3S
 - Nytt mixtest
 - Test med gamla och nya mixen (som blir ny baseline)
- ▶ Även patchningar, ny HW, Config eller uppgraderingar bör testas på detta sätt

Prestandatester i produktion

- ▶ Samma UC som prestandatestades bör även testas i produktion
 - Fast som aktiva monitorer / robotar
- ▶ Samma mätning avseende volym och prestanda
 - Samma verktyg (APM) som kör passiv mätning i testmiljöer som i prod för att kunna jämföra

Avrundning

»» Det är ingen rocket science
egentligen

Vad kan LIL göra?

- ▶ Marknadsledande PEC i Stockholm
 - Komplett outsourcing av ett PEC
 - Helt eller delvis
 - Egen infrastruktur, etablerad i Amazon EC2, Azure, Bahnhof ECS samt EPiServer hosting, hyra vid behov, eller VPN till kundsystem
 - Egna mobila lastgeneratorer och miljöer
 - Egna verktyg (Microsoft, opensource, m.fl.)
 - Egen övervakningsplattform för APM

Vad kan LIL göra?

- ▶ Sveriges bästa PE mentorer och konsulter
 - Produktoberoende experter (idag över 80 manår)
 - Certifierade produktkonsulter och partner med HP Software (LoadRunner/PC, BAC)
 - Inner Circle Partner och blivande ALM Silver Partner med Microsoft (VS, TFS, SCOM)
 - Leverantör till Microsoft Services
 - Gediget Open Source engagemang
 - Mentoring, rådgivning, workshops och utbildning

Tack för mig !

www.prestandatest.se

- Slides och material från idag
- Prenumerera på vårt Nyhetsbrev
- Ta kontinuerligt del av våra erfarenheter och vidareutvecklingar
- ▶ **Kommer snart:**
 - Läs mer om våra testmetoder och testprocessen, white papers, success stories, best practices, verktygsguider, m.m.
 - Bloggar från alla våra medarbetare som delar med sig av sin kunskap och erfarenheter
 - Videoguider för utvalda verktyg
 - Länkar till mer information, andra metoder, verktyg, bloggar, utbildningar mm.
- ▶ Gilla oss på **Facebook.com/lightsinline**

Diskussioner, Frågor

»» Vad är utmaningarna med
Prestandatester?