



MALMÖ HÖGSKOLA

Deployment, Delivery and Integration

Dagens agenda

- Förra veckans föreläsning
 - Testning och refaktorisering
- Deployment
- Delivery
- Integration
- Continuous Integration, Delivery & Deployment

Förra veckan: Testning

Vad är testning?

“**Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.”

Enl. Wikipedia

- Validering → Bygger vi rätt saker?
- Verifikation → Bygger vi sakerna på rätt sätt?

Lockart, *Modern PHP*. Ch 10

Wikipedia, *Software testing* - https://en.wikipedia.org/wiki/Software_testing

Varför testar vi våra applikationer?

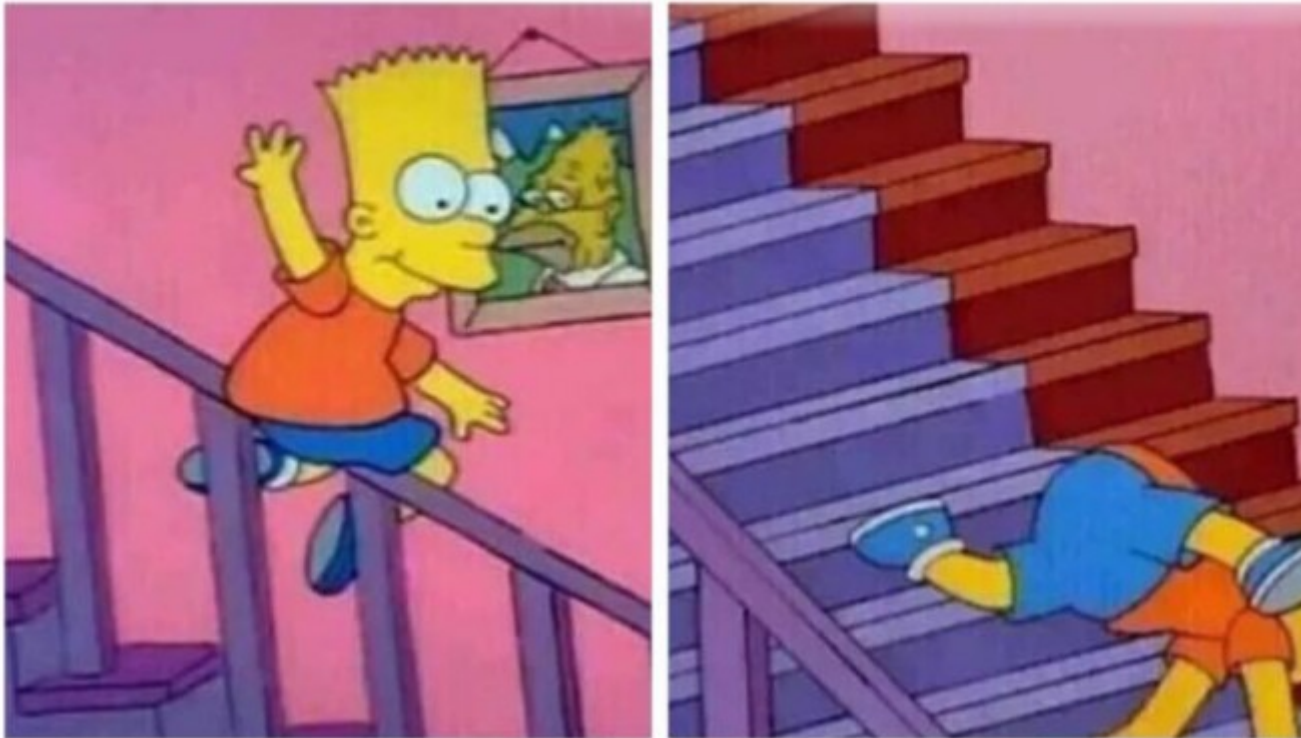
- Tekniska skäl
- Utvecklingsteamets skäl
- Ekonomiska skäl

Lockart, *Modern PHP*. Ch 10

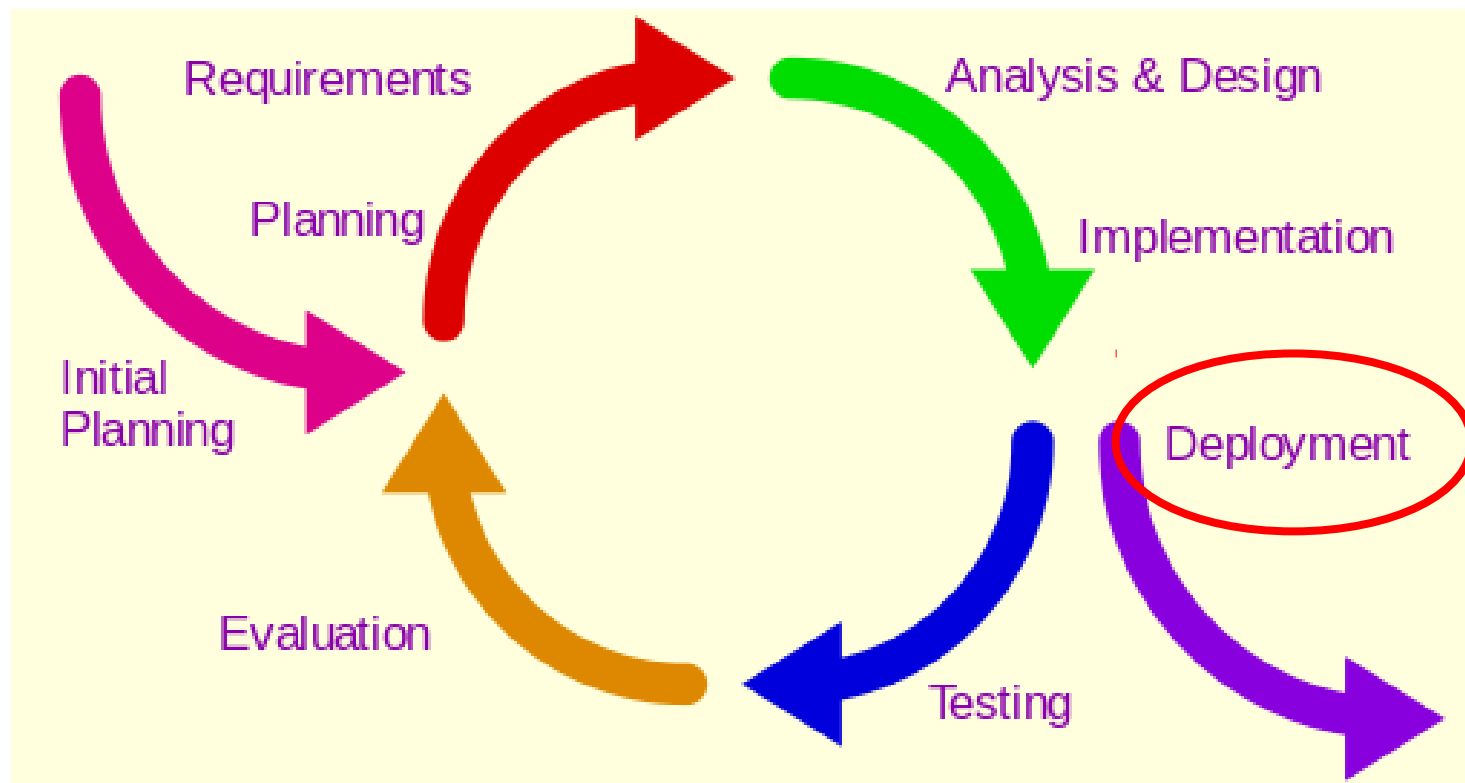
Tekniska skäl: Säkerställ funktionaliteten

My code working well on on my machine

** Deploys **



Under utvecklingen



Wikipedia, *Iterative and incremental development* - https://en.wikipedia.org/wiki/Iterative_and_incremental_development

Efter utvecklingen

Integrationstestning: Testa applikationen innan den driftsätts i produktion.

Regressionstestning: Säkerställ att “gammal” funktionalitet inte förstörs i nya releaser.

Felsökning: Dokumentera fixade buggar genom att skriva tester som visar att felet är åtgärdat.

Lockart, *Modern PHP*. Ch 10

Enhetstester med PHPUnit

- Testar de enskilda beståndsdelarna av en applikation
- Flera tester för varje enskild metod och klass
 - Testa många typer av indata
 - Välj ut relevanta typfall, ex. gränsfallen
 - Skriv testerna i samband med att koden skrivs

Refaktorisering

Vad är refaktorisering?

“**Code refactoring** is the process of restructuring existing computer code—changing the factoring—without changing its external behavior.”

Enl. Wikipedia

Typer av refaktorisering

Abstraktion: Dölj implementation för att enklare kunna utbyta den vid behov.

Nedbrytning: Bryt ner långa funktioner och klasser i mindre, mer förståeliga enheter.

Namngivning: Vettigare namngivning av klasser, variabler och metoder.

Omstrukturering: Mer logisk placering av variabler, metoder och kommentarer.

Varför refaktorerar vi vår kod?

- Tydlighet
- Förtroende
- Prestanda

Hur refaktorerar vi vår kod?

Fungerande,
vältestad kod



Välj ut en code smell
som ska åtgärdas



Deployment

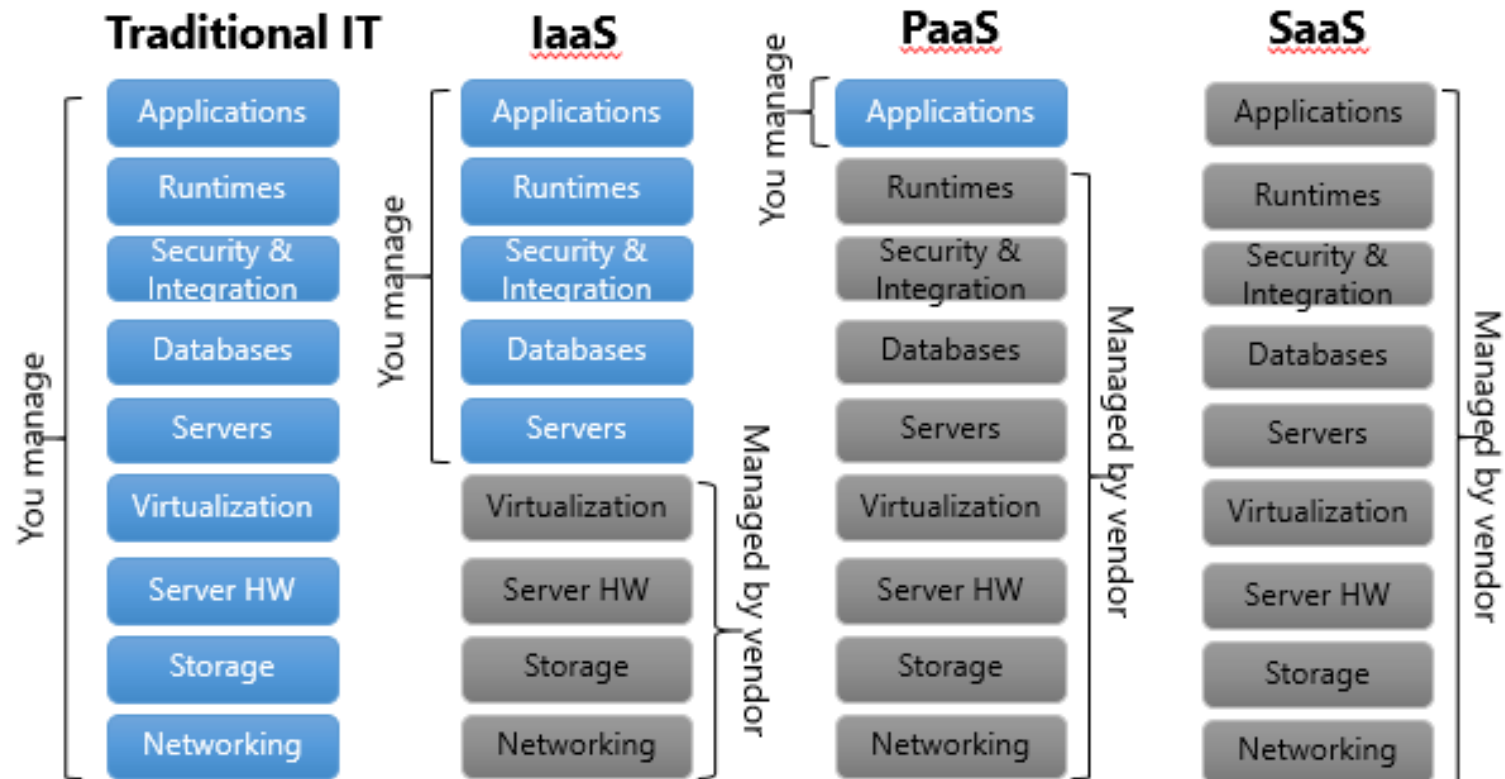


Vad är deployment?

Deployment = driftsättning

- Vi testar mjukvaran
- Vi säkerställer att allt fungerar
- Vi sätter koden i produktion
- Vi gör det aldrig på en fredag

Olika mål för driftsättning



Olika mål för driftsättning: Traditionell IT

- Driftsättning på egna servrar
- Blir mindre och mindre vanligt
- Fördelar
 - Full kontroll
- Nackdelar
 - Någon måste drifta serverna och infrastrukturen
 - Svårt att växla upp och ner
 - Mycket jobb
 - Dyrt

Olika mål för driftsättning: IaaS

- Driftsättning på virtuella servrar
- Fördelar
 - Skalfördelar – billigare än traditionell IT
 - Relativt hög kontroll – fokus på OS, runtime och applikation
- Nackdelar
 - Fortfarande mycket arbete med att drifta OS och runtime

Det är detta vi kommer att testa i veckan

Olika mål för driftsättning: PaaS

- Driftsättning med containers eller på webbhotell
- Fördelar
 - Billigt
 - Enkelt
 - Snabbt
 - Standardiserat
- Nackdelar (?)
 - Mindre kontroll över underliggande infrastruktur

Olika mål för driftsättning: SaaS

- Driftsättning på exempel Wordpress.org
- Fördelar
 - Jätteenkelt
 - Jättebilligt
- Nackdelar
 - Oflexibelt
 - Svårt att göra avancerade applikationer

Delivery



Delivery

Delivery = leverans

- Driftsättning av kod
- Publicering av dokumentation
- Publicering av testresultat

Integration



Integration

- Få alla delar av mjukvaran att fungera ihop
 - I ett mindre PHP-projekt är det sällan ett problem
 - I större projekt med många utvecklingsteam kommer problem att uppstå
 - Använd integrationstester!
 - Testa ordentligt innan driftsättning!

What's the fuzz about?

Driftsättning är läskigt

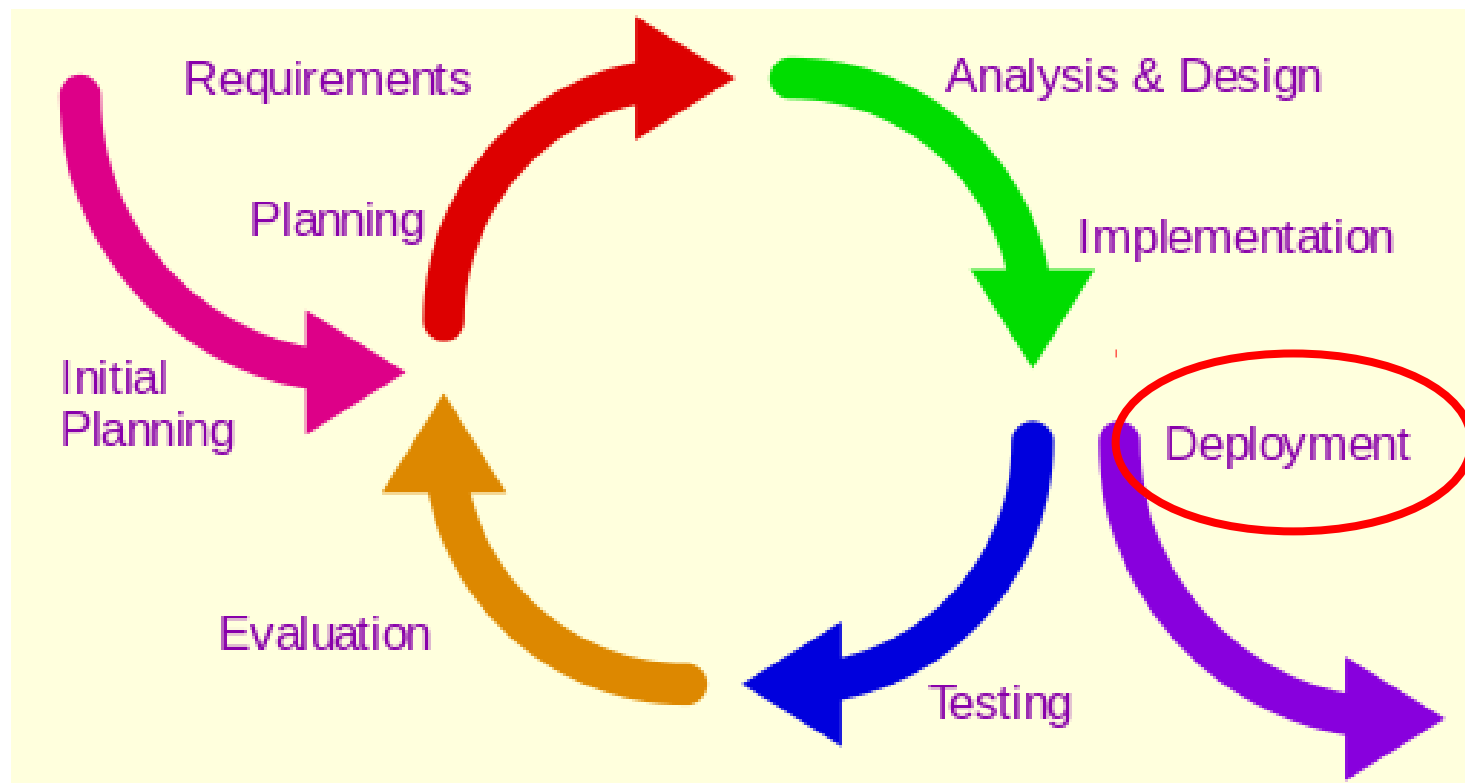
- Den naiva lösningen är manuell
- Många steg → sårbart
- Lösningen är beroende på applikationen

Kan vi förenkla det här?

What's the fuzz about?

Yes! Med automatiserad deployment!

Under utvecklingen



Wikipedia, *Iterative and incremental development* - https://en.wikipedia.org/wiki/Iterative_and_incremental_development

Deployment i två steg

- Staging
 - Här körs utvecklingsversionen av en mjukvara (develop)
- Produktion
 - Här körs mjukvaran på riktigt (master)

Continuous Integration Continuous Delivery Continuous Deployment

Continuous Integration (CI):

- Automatiserade tester vid varje push till develop
- Ser till att koden alltid är vältestad och fungerande

Continuous Delivery (CD):

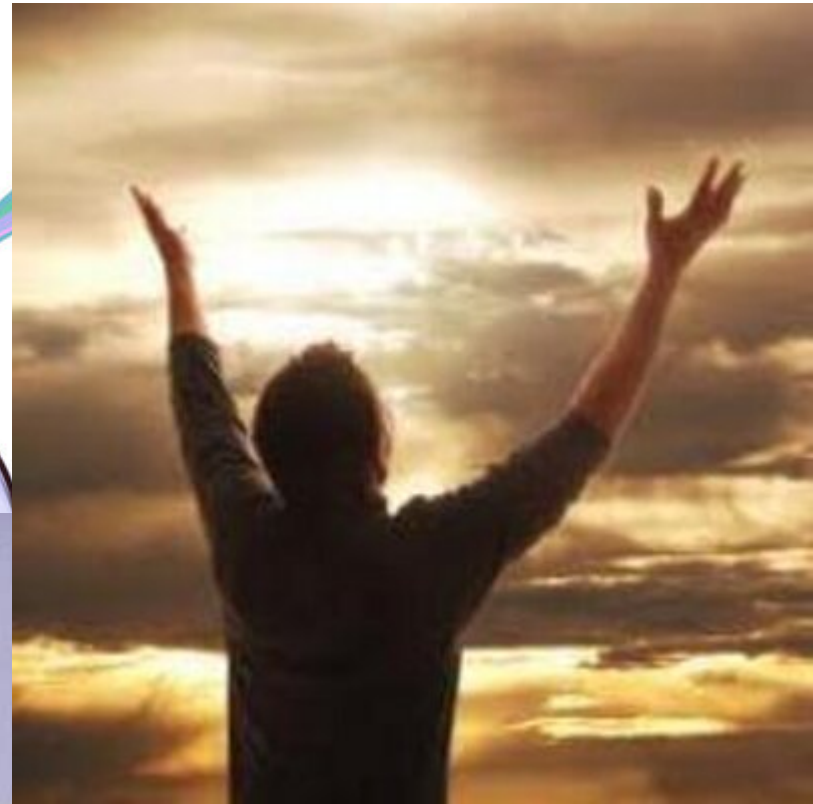
- Automatiserade tester och byggen vid varje push till master
- Ser till att applikationen alltid är levererbar

Continuous Deployment:

- Automatiserad leverans efter CI & CD
- Ser till att applikationen alltid kör den senaste versionen

Wikipedia, *Continuous integration*. https://en.wikipedia.org/wiki/Continuous_integration

Wikipedia, *Continuous delivery*. https://en.wikipedia.org/wiki/Continuous_delivery



Demo



Travis CI



Microsoft Azure