Deployment, Delivery and Integration

DA288A – Molnbaserade Webbapplikationer

Johan Holmberg 2018-05-09

MALMÖ UNIVERSITET

Dagens agenda

- Repetition från förra veckan
  - Testning och refaktorisering
- Deployment
- Delivery
- Integration
- Continous Integration, Delivery & Deployment

MALMÖ UNIVERSITET

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

MALMÖ UNIVERSITET

Varför testar vi våra applikationer?

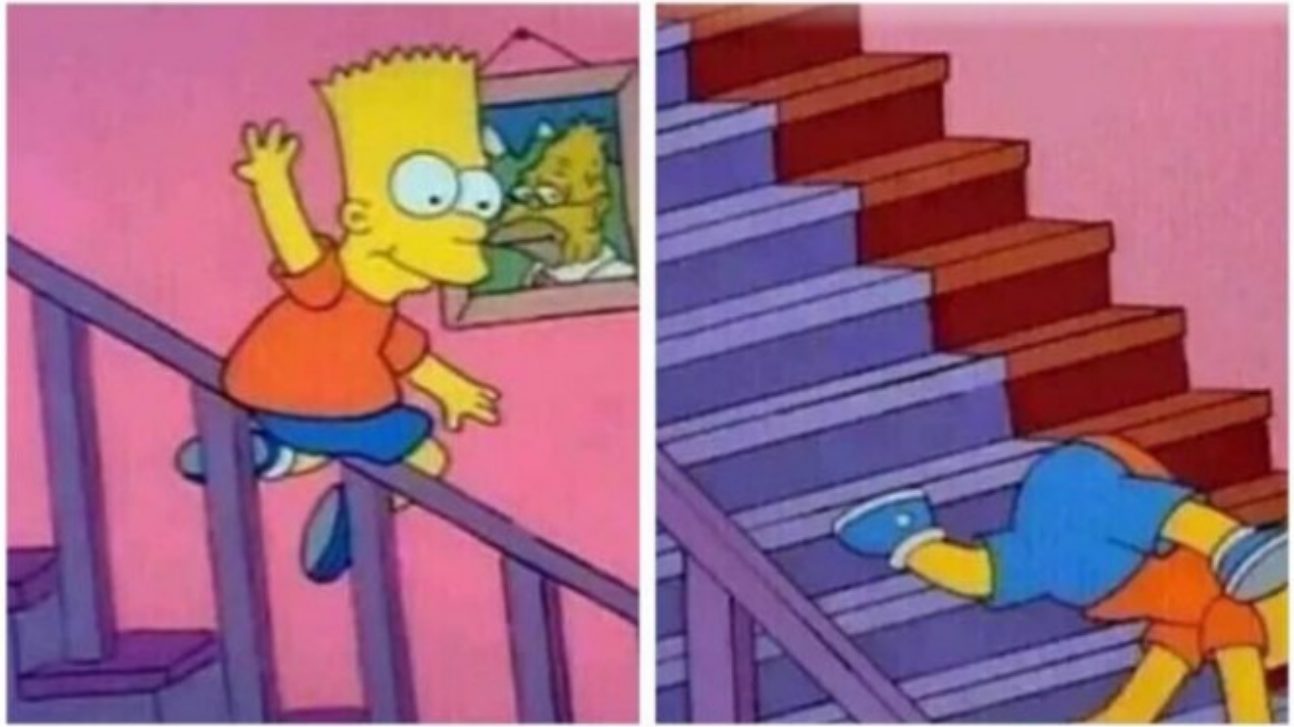
- Tekniska skäl
- Utvecklingsteamets skäl
- Ekonomiska skäl

Lockart, *Modern PHP*. Ch 10

MALMÖ UNIVERSITET

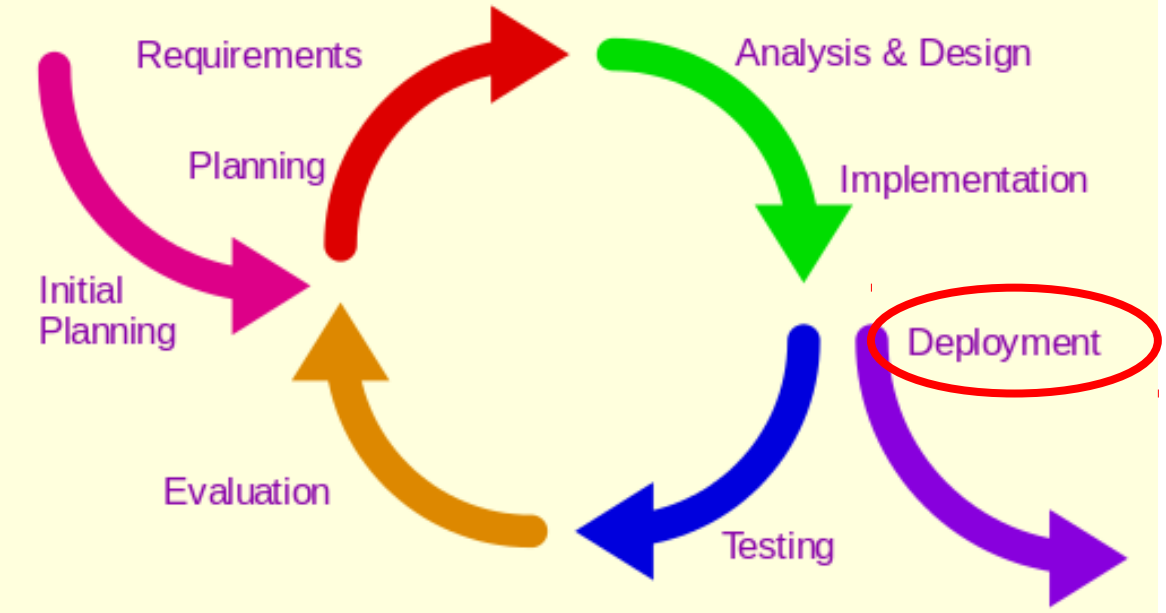
My code working well on on my machine

* *Deploys* *



MALMÖ UNIVERSITET

Under utvecklingen



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

MALMÖ UNIVERSITET

Efter utvecklingen

Integrationstestning: Testa applikationen i sin helhet 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

MALMÖ UNIVERSITET

Integrationstester med Story BDD

- *Story BDD* står för *Story Behaviour-Driven Design*
- Testar applikationen i sin helhet
- Flera tester som visar hur applikationen ska fungera
  - Testerna skrivs i “berättelseform”:
    “As a paying customer, I need to log in”
  - Skriv testerna innan koden – med BDD får du hjälp med att skapa koden du behöver!

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, exempelvis gränsfallen
  - Skriv testerna i samband med att koden skrivs

Vad är refaktorisering?

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

Enl. Wikipedia

Wikipedia, *Code refactoring* – https://en.wikipedia.org/wiki/Code_refactoring

Typer av refaktorisering

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

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

Namngivning: Vettigare namngivning av klasser, variabler och metoder

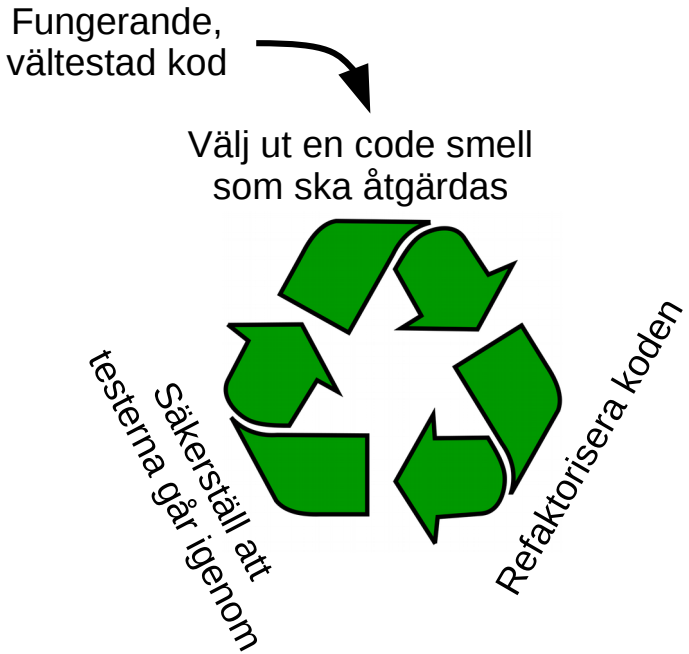
Omstrukturering: Mer logisk placering av variabler, metoder och kommentarer.

Varför refaktoriserar vi vår kod?

- Tydlighet
- Förtroende
- Prestanda

MALMÖ UNIVERSITET

Hur refaktoriserar vi vår kod?



MALMÖ UNIVERSITET

Deployment

MALMÖ UNIVERSITET

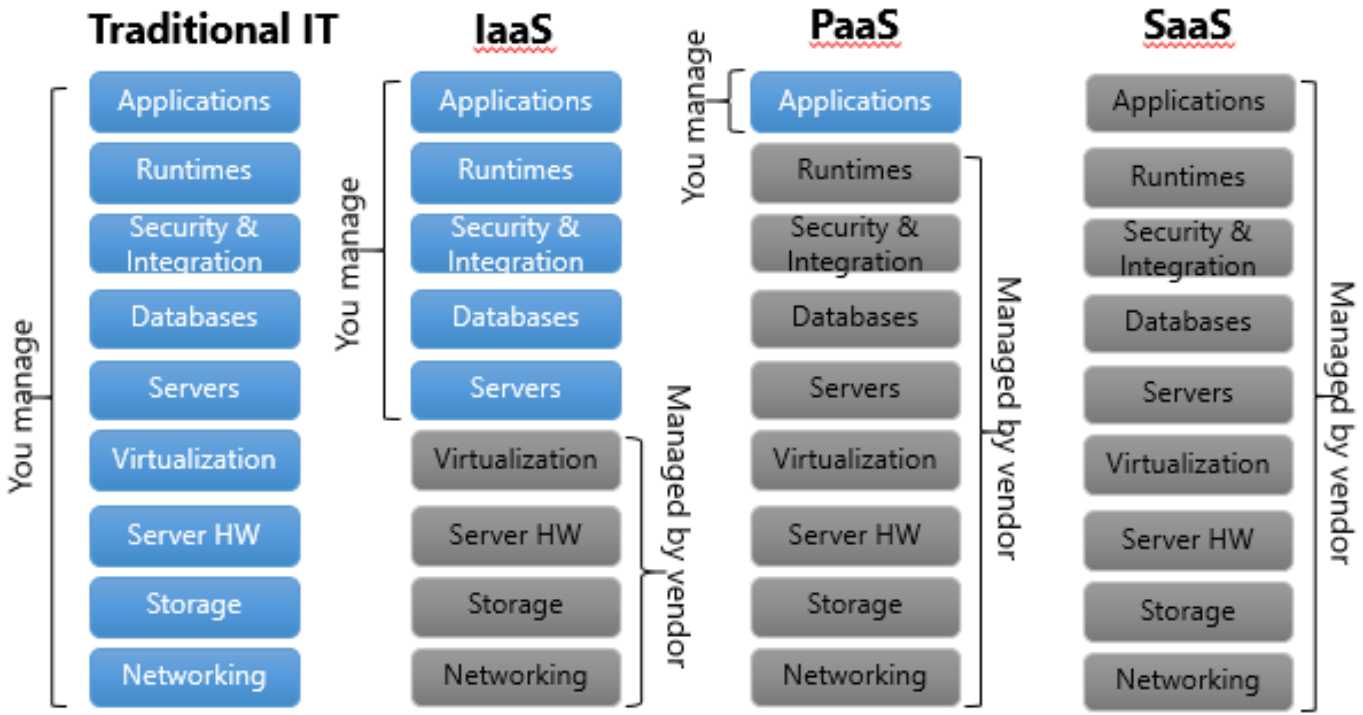
Vad är deployment?

Deployment = driftsättning

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

MALMÖ UNIVERSITET

Driftmöjligheter



MALMÖ UNIVERSITET

Traditionell IT

- Driftsättning på egna servrar
- Blir mindre och mindre vanligt
  - De flesta startups använder inte egen infrastruktur
  - De flesta företag mixar med molntjänster
- Fördelar
  - Full kontroll
- Nackdelar
  - Någon måste drifta servrarna och infrastrukturen
  - Svårt att växla upp och ner
  - Mycket jobb
  - Relativt dyrt

MALMÖ UNIVERSITET

IaaS – Infrastructure as a Service

- Driftsättning på virtuella servrar
  - Virtuella maskiner – du kan använda samma mjukvara på din utvecklingsmaskin som i driftmiljön!
- 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

PaaS – Platform as a Service

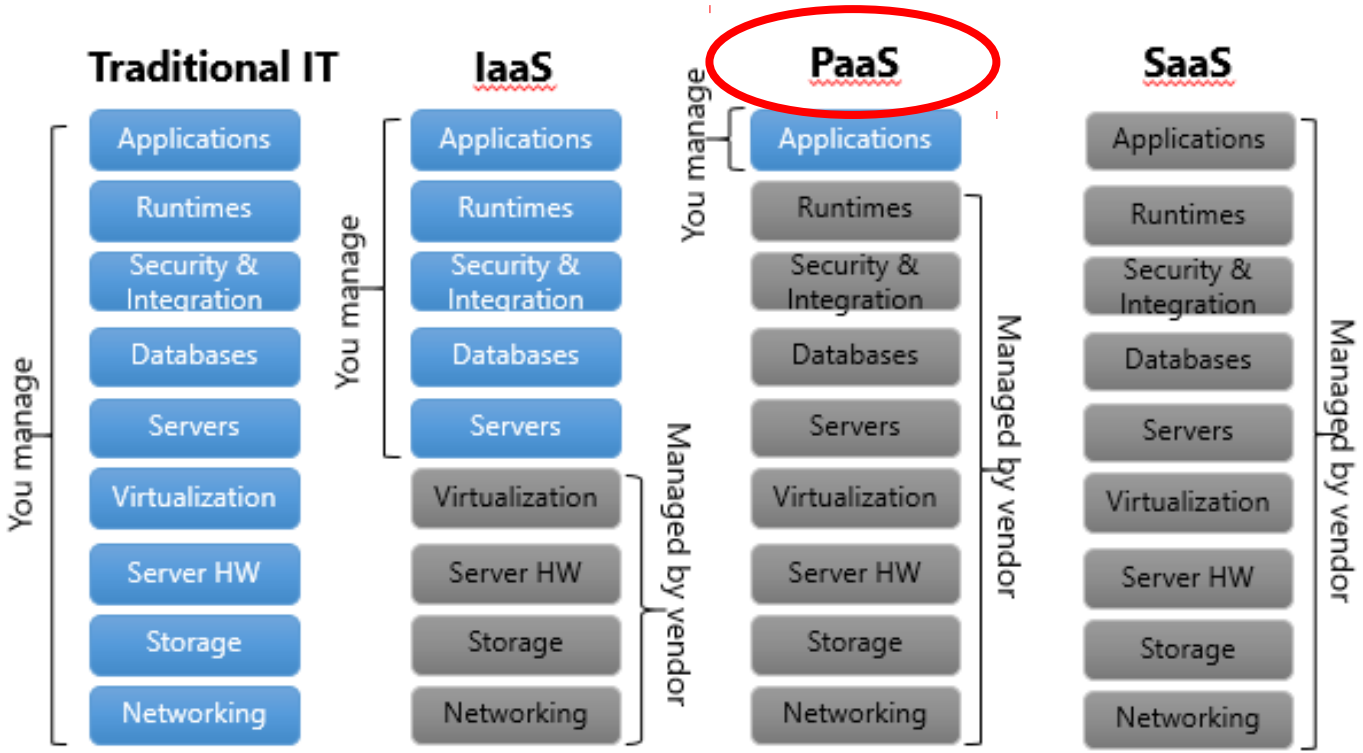
- Driftsättning på en plattform som kontrolleras av en leverantör
  - Webbhotell
  - Container-hosting
  - Specialiserade plattformar, exempelvis för IoT
- Fördelar
  - Billigt
  - Enkelt
  - Snabbt
  - Standardiserat
- Nackdelar (?)
  - Mindre kontroll över underliggande infrastruktur

SaaS – Software as a Service

- Någon annan driftar mjukvaran som du använder
  - Exempel:
    - Wordpress.org
    - Its learning
- Fördelar
  - Jätteenkelt
  - Jättebilligt
- Nackdelar
  - Oflexibelt
  - Svårt att göra avancerade applikationer

MALMÖ UNIVERSITET

Driftmöjligheter



MALMÖ UNIVERSITET

Några IaaS- och PaaS-leverantörer

Google Cloud

IBM Cloud

Azure

MALMÖ UNIVERSITET

Delivery

MALMÖ UNIVERSITET

Vad är delivery?

Delivery = leverans

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

MALMÖ UNIVERSITET

Integration

MALMÖ UNIVERSITET

Integration

- Få alla delar av mjukvaran att fungera ihop
  - I ett mindre PHP-projekt är detta 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 (som med FTP)
- Många steg → sårbart
- Lösningen är beroende av applikationen

Kan vi förenkla det här?

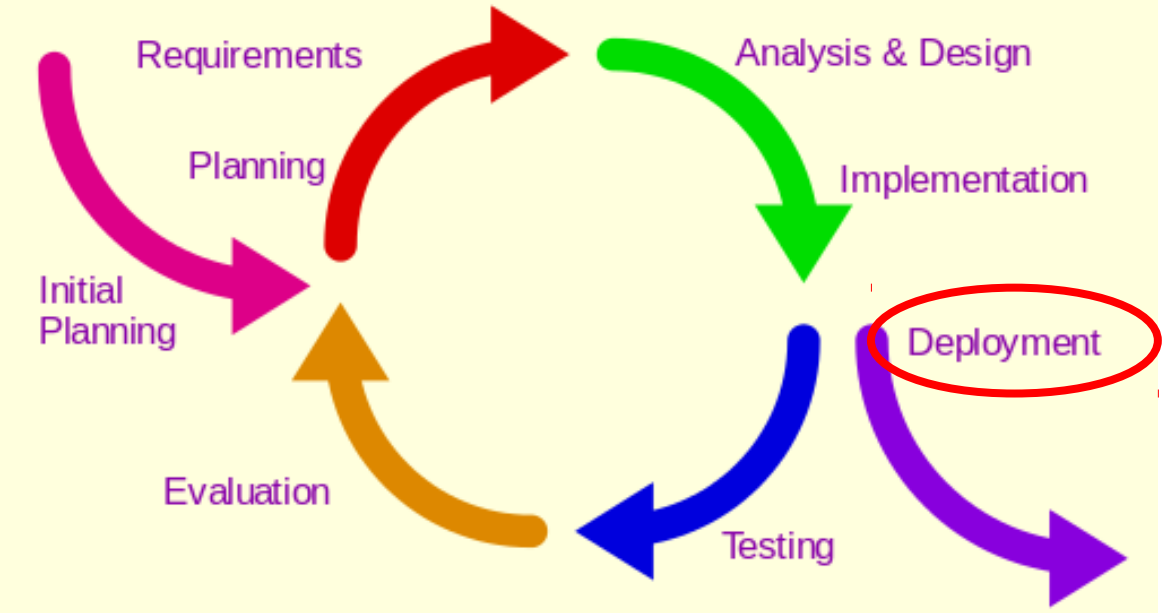
MALMÖ UNIVERSITET

What’s the fuzz about?

Yes! Med automatiserad deployment!

MALMÖ UNIVERSITET

Under utvecklingen



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

MALMÖ UNIVERSITET

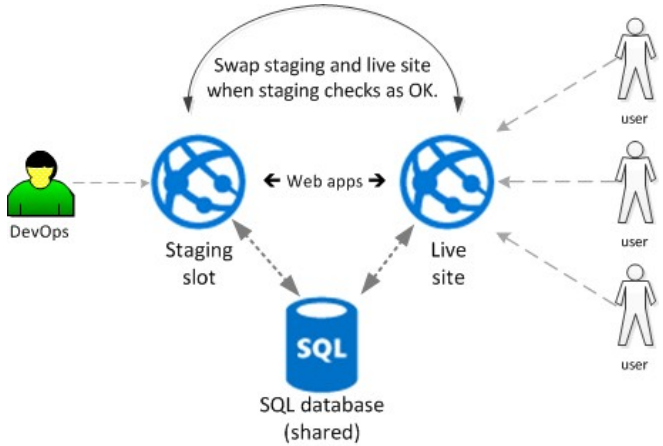
Deployment i två steg

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

MALMÖ UNIVERSITET

Driftsättning med slots

Använder två “applikationsplatser”, som växlar mellan att vara staging och produktion:



Redgate Software, *Deploying an Entity Framework Database into Production*.
https://www.red-gate.com/simple-talk/dotnet/net-framework/deploying-an-entity-framework-database-into-production/

MALMÖ UNIVERSITET

CI, CD och CI/CD

Continous Integration (CI)

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

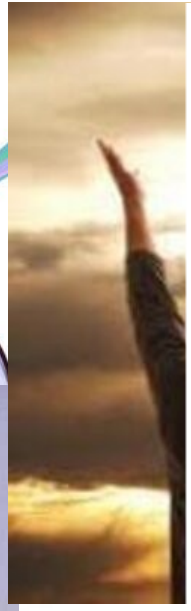
Continous Delivery (CD)

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

Continous Deployment (CI/CD)

- Automatiserad leverans efter CI och CD
- Ser till att applikationen alltid kör den senaste versionen av koden

Wikipedia, *Continous integration*. https://en.wikipedia.org/wiki/Continous_integration
Wikipedia, *Continous delivery*. https://en.wikipedia.org/wiki/Continous_delivery



D)

Cl o

kör

Con

ntin

Wikipedia, C
Wikipedia, C

MALMÖ UNIVERSITET

Demo

Travis CI

MALMÖ UNIVERSITET