



TIGERTEAM®

Java og JEE



Udfordringer og problemstillinger

En liste over de udfordringer og problemstillinger, der er ved Java og JEE udvikling



Generelt

om

Java og JEE



Generelt, I

- ❏ Man undervurderer hvor mange Java programmører, der trænger til at få genopfrisket Objekt Orienteret tankegang
- ❏ Man tror, at en uges kursus i teknologi X er nok. Derefter er udvikleren klædt ordentligt på
- ❏ Man undlader at kigge på udviklingssupplementer, så som Aspekt Orientering eller Model Dreven Software udvikling, fordi:
 - ❏ "Det forstår vores udviklere alligevel ikke"...
 - ❏ "Det er ikke sådan vi udvikler her"...
 - ❏ "Hvordan skal vi håndtere uddannelse af nye, der skal forstå det"...
 - ❏ etc...

Generelt, II

- ❏ Man vælger et produkt eller en standard, uden at fokusere på projektets behov
- ❏ Man vælger et stort produkt eller framework, selv om enklere og bedre alternativer findes
- ❏ Man vælger frameworks, der kræver store mængder XML konfiguration
- ❏ Man vælger at kode et framework selv, på trods af, at der findes gode opensource alternativer

Generelt, III

- ❏ Man designer en arkitektur hvor alt er konfigurerbart
 - ❏ ”Man kan jo aldrig vide hvad man har behov for at ændre senere”
- ❏ Man vælger at designe hele arkitekturen færdig, før der er skrevet en eneste linie applikationskode.
 - ❏ Medfører typisk en rodet arkitektur med de forkerte abstraktioner
- ❏ Man vælger at kode applikationskode uden at have tænkt på arkitekturen og projektets behov
 - ❏ Medfører oftest en alt for hård kobling, der i det lange løb resulterer i, at projektet enten udfases eller at koden helt skal skrives om

Generelt, IV

- ❏ Man vælger at afkoble domæne modellen fra web klient laget
- ❏ Man vælger at lade domæne modellen skinne igennem WebService laget
- ❏ Man vælger, i skalerings navn, at fysisk adskille lag, der burde være co-lokerede
- ❏ Man vælger at udsulte sin domæne model for, til gengæld, at have et façade/service/transaktionslag, der er unødigt komplekst eller fragmenteret

Generelt, V

- ❏ Man vælger at bruge JDBC når ORM er et bedre valg
- ❏ Man vælger at bruge en ORM model når en JDBC drevet model er et bedre valg
- ❏ Man vælger en web teknologi, der er for kompleks eller som kun dækker 20% af behovene
- ❏ Man går overbord med Ajax fordi det er frækt, nyt og det folk tror, de vil have
- ❏ Man undervurderer effekten af at bruge Java 5's (eller senere) sprog features

Test

test

TEST



Test, test, test, I

- ❏ Man nedprioriterer unit tests
 - ❏ Mister dermed mulighederne for at refactorere og dermed mister man kernen i agile, iterativ udvikling
- ❏ Man vælger frameworks, der ikke understøtter unit test, uden opstart af store runtime miljøer, som f.eks. servere
- ❏ Mister mulighederne og værdien i at udnytte sprogets faciliteter
 - ❏ Risiko for at miste type sikkerhed

Test, test, test, II

- ❏ Man vælger for løst koblede og svagt typede teknologier
 - ❏ Uden unit tests fungerer det ikke. Man mister muligheden for at refactorere sin model/domæne
- ❏ Man vælger en arkitektur, der er for hårdt koblet
 - ❏ Besværliggør unit test (eksempel Singleton pattern)

Mulighed

for

Automatisering



Mulighed for Automatisering

- ❏ Man glemmer at fokusere på værdien af automatiseret build, test og deployment
- ❏ Man vælger at kode ting i hånden, der effektivt kan automatiseres gennem Patterns, Aspekt Orienteret Programmering (AOP) eller Model Drevne Software udvikling (MDSD)
- ❏ Der mangler folk, der er kritiske og som rettidigt er i stand til at se mønstre (patterns) og abstrahere.

Og

løsningen

ER!



Hvad kan man gøre?

- ❏ Brug interne og eksterne erfaringer til at definere en grundlæggende arkitektur, der dækker løsningens kendte behov
 - ❏ Behovsdreven arkitektur
- ❏ Lav en "deep and narrow" arkitektur i de tidlige iterationer
 - ❏ Tænk stort implementér i små bidder efter behov
- ❏ Lav proff-of-concepts, eller spikes af de største tekniske udfordringer, så tidligt som muligt
- ❏ Prioriter test og automatisering højt
- ❏ Vær kritisk overfor teknologi og produkt valg



TIGERTEAM®

LEAN THINKING

www.tigerteam.dk

Copyright 2007