

# Dansker gør webprogrammører glade

David Heinemeier Hansson har udviklet frameworket Ruby on Rails. Målet er at gøre webprogrammering til en oplevelse.

Af Stig Andersen, prosabladet@prosa.dk

Danske David Heinemeier Hansen har gjort det, mange udviklere må nøjes med at drømme om. Han har udviklet teknologi, som har en væsentlig indflydelse på det daglige arbejde for tusindvis af denne verdens programmører.

David er 29 år, københavn-  
ner bosiddende i

Chicago, medejer af udviklingsvirksomheden 37signals og ikke mindst skaber af frameworket Ruby on Rails (RoR). For David Heinemeier Hansson handler programmering ikke bare om at få udviklet nogle applikationer eller få nogle websites til at køre.

– Det vigtige er den oplevelse, man har ved at bruge værktøjet. Om man synes, den kode, man laver, er smuk og elegant, eller om man føler, man hurtigt bliver trukket ind i noget, der er quick and dirty eller besværligt, siger han

For fem år siden fik David Heinemeier Hansson helt frie hænder til selv at vælge værktøjer, da han som eneste programmør på projektet skulle udvikle produktet Basecamp for 37signals. Ef-

ter at have brugt en uges tid på at sætte sig ind i programmeringssproget Ruby, var han solgt.

– Det var en helt anderledes oplevelse, end jeg nogensinde havde haft med andre programmeringssprog. Jeg blev især meget inspireret af selve baggrunden for Ruby, som er at gøre programmører glade og tilfredse. Jeg besluttede derfor at tage de gode idéer fra PHP og Java, fjerne nogle af de ting, der irriterede mig, og så implementere det i Ruby. På den måde ville jeg sørge for, at Ruby også blev optimeret for glade webprogrammører, fortæller han.

### Konkrete problemstillinger

Udviklingen af RoR tager udelukkende afsæt i konkrete problemstillinger, som webprogrammører møder i dagligdagen. Ifølge David Heinemeier Hansson er det helt afgørende, at man har fat i det konkrete, før man begynder at lave generaliseringer. David Heinemeier Hansson arbejder stadig rigtig meget i RoR, og stort set alle forbedringer og nye features er uddraget af konkret produktionskode.

David Heinemeier Hansson havde ikke fra starten en klar forestilling om, hvad RoR skulle udvikle sig til. Og han havde egentlig heller ikke planlagt at udgive noget. Han havde blot løbende udviklet en række generelle komponenter, som gjorde programmeringsarbejdet nemmere for ham. Så han tog beslutningen om at stille værktøjerne til rådighed for andre. Hvis de værktøjer, han havde udviklet, havde gjort ham mere tilfreds og produktiv, hvorfor så ikke lade andre få glæde af dem?

– Jeg havde så at sige opdaget ”den

hemmelige guldgrube” i form af Ruby. Så RoR var et bidrag til at udbrede Ruby generelt ved at give udviklerne noget konkret, som virkelig kunne få dem op af stolene, fortæller han.

Hvad angår visioner og planer for RoR, er det stadig en helt grundlæggende pointe, at der ikke er en masse fremtidsvisioner om, hvordan frameworket ser ud om fem år.

– Vil lægger planer for, hvad vi har tænkt os at lave lige nu. Og det udspringer af, hvilke problemer vi sidder med lige nu og ikke hvilke problemer andre et eller andet sted måske kunne have. Så alle tager udgangspunkt i de applikationer, de arbejder med i det daglige. Og hvis de løber ind i et konkret problem mere end en gang, så begynder de at tænke i at udvikle en generaliseret løsning, siger David Heinemeier Hansson.

Som med alle andre nyere teknologier er der også livlig diskussion i diverse fora om

fordele og ulemper ved RoR. Det sociale netværkssite

Twitter,

som kører på RoR, havde for cirka et år siden alvorlige driftsproblemer, og der blev stillet spørgsmålstejn ved RoR's evne til at skalere. David Heinemeier Hansson bliver meget fast i mælet, da han afviser påstanden:

– At hævde at noget ikke skalerer, er ganske enkelt den bedste måde at så tvivl og bekymring om en teknologi uden at blive konkret. Men det har ikke hold i virkeligheden. Alle frameworks i dag skalerer. Så det er absolut muligt at skalere en løsning i RoR, helt på samme måde som man skalerer i PHP, Python, osv. Men det kan være et utroligt svært budskab at komme igennem med, selvom der er

meget store websites, som for eksempel Yellowpages.com, der kører på RoR.

### Plads til gode idéer

Sommetider har man indtryk af, at frontløberne inden for softwarebranchen inklusive open source-miljøet næsten får gurustatus. Selvom David Heinemeier Hansson jævnligt optræder på store konferencer og har figureret på lister over mest indflydelsesrige personer, har han ikke et billede af sig selv som en softwareguru.

– Det eneste, jeg havde, var nogle gode idéer, som jeg kunne præsentere på blogs og konferencer. Det gode ved open source er, at det giver gode idéer en utrolig nem måde at boble op på. Jeg sad for eksempel i en lejlighed på Amager og udviklede noget til RoR, som har haft stor indflydelse på, hvordan tusindvis af mennesker udvikler software. Og når det er sådan, at enkelt personer forholdsvis nemt kan komme frem med nye idéer, så kan der nogle gange komme en vis personfiksering, og det har jeg en lidt ambivalent holdning til. Men det er nok sådan i dag, at det er nemmere at få gode idéer ud, hvis de er knyttet til en persons historie, siger han.

Tech-branchen udmærker sig ifølge David Heinemeier Hansson ved at være en meget isoleret niche. Selvom man er et kendt ansigt i branchen, er der ingen problemer med at gå i fred på gaden uden for conferencecentrene. Og rådet fra David Heinemeier Hansson til kommende profiler i open source-miljøet er klart.

– Man skal endelig ikke skal have som mål i sig selv at ”blive berømt”. Open source-miljøet er et meget fagligt orienteret miljø, så den eneste måde at profilere sig på er at have bedre idéer end andre. Og det gode er, at man ikke behøver at opfinde helt nye og revolutionerende teknologier. Det er helt fint at komme frem med noget, der gør et eller andet 20-30 procent bedre, end man har kunnet før, siger han. ■

**Det gode ved open source er, at det giver gode idéer en utrolig nem måde at boble op på.**

David Heinemeier Hansen, skaber af Ruby on Rails

Foto: Andreas Offesson



# Når programmering bliver en oplevelse

Programmeringsæstetik og produktivitet går hånd i hånd i Ruby on Rails. Frameworket til webudvikling er skabt til at gøre programmering til en oplevelse.

Af Stig Andersen, Prosabladet@prosa.dk

Lenio i Århus er et af de efterhånden mange udviklingshuse i Danmark, der har satset på open source-frameworket Ruby on Rails (RoR) til udvikling af webapplikationer.

– Vi plejer at sige, at vi har 100 forskellige webframeworks, men ingen af dem gør os rigtig lykkelige. Så da RoR kom, var det virkelig en wow-oplevelse, fordi det løser et helt specifikt problem, nemlig at lave webapplikationer. Tingene bliver bundet sammen på en måde, så det virker naturligt, og det er let og sjovt at arbejde med, fortæller udviklingschef Tommy Dejbjerg Pedersen.

Programmeringssproget Ruby blev i midten af 1990'erne udviklet af japaneren Yukihiro Matsumoto, men det var først, da danskeren David Heinemeier Hansson udviklede frameworket Rails, at det virkelig lattede. Først i USA, og i de senere år også her i Danmark.

RoR er baseret på principperne om "convention over configuration" og DRY (Don't Repeat Yourself). Og som noget enestående i forhold til andre sprog og frameworks udviklet med det erklærede mål at gøre selve programmeringen til en oplevelse. Christian Dalager, udvikler hos Eksponent, siger:

– Det handler faktisk om, hvad man kunne kalde programmøræstetik. Det

er enkelt og smukt med gode sprogkonstruktioner, og man kan få udført rigtig meget med få linjer kode. Og som vi siger: It reads out well.

Oplevelsen og æstetikken er dog ikke det endegyldige mål for RoR. Filosofien er, at hvis motivationen og programmeringsglæden er stor nok, så bliver den enkelte programmør mere produktiv. Og det er almindelig anerkendt, at man i RoR typisk kan nøjes med færre kodelinjer end i eksempelvis Java og PHP.

RoR er specielt udbredt i virksomheder, der praktiserer Agile Software Development, hvor man satser på iterativ udvikling i tæt samarbejde med kunden. Og frameworket er så specifikt rettet mod udvikling af webapplikationer, at man meget hurtigt kan få omsat idéer på whiteboardet til en kørende prototype. Jesper Rønn-Jensen, udvikler hos Cap Gemini, fortæller:

– Vi bruger RoR til sammen med kunden at afklare behovene og afprøve mulige løsninger. Det er meget nemmere at få en fornemmelse af kundernes forretning, hvis de i udviklingsfasen kan forholde sig til noget konkret og visuelt. Og så er man jo allerede i gang med at lave løsningen. Samtidig kan det også være en udfordring, at kunderne kan blive ved med at foreslå ændringer, for det virker

meget ukompliceret at justere prototypen løbende.

Den hurtige vej fra idé til kørende system er ifølge Torben Dejbjerg Pedersen et kæmpeplus, men rummer også en udfordring. – Det er meget nemt hurtigt at få noget op at køre, men erfaringen viser, at på et eller andet tidspunkt får man brug for at kunne dykke helt ned under motorhjelmene og tweeke et eller andet. Det kan for eksempel være i en applikation, hvor det viser sig, at JavaScript performer dårligt i en bestemt browser. Sådan noget skal man selv løse, medmindre man kan finde et tool, der løser problemet, fortæller han.

RoR er ikke længere det eneste webframework på markedet. Microsoft er på vej med virksomhedens .NET MVC framework, og til Java-verdenen er Groovy on Grails et spændende bud. Som navngivningen afslører, er det dog tydeligt, hvor inspirationen kommer fra. ■

## Eksempler fra kodesamlingen

**Henrik Løvborg, udvikler hos Lenio har sammensat en række kodeeksempler, som viser styrkerne ved Ruby on Rails sammenlignet med Java. Se [www.prosa.dk/link/208](http://www.prosa.dk/link/208)**