

Frank Thomsen
Nyvangsvej 22
8462 Harlev

Projekter

Følgende er en liste og kort opsummering af de projekter jeg har deltaget i.

CSC Scandihealth (tidligere afdeling af KMD Kommunedata)

Januar 2000 – april 2001

Ansæt som systemudvikler på et projekt, der lavede en elektronisk patientjournal. Dette system havde en tredjeparts klient, som fortolkede data og definitioner lagt i databasen via den forretningslogik vi skrev i SQL.

AI udvikling foregik i SQL på serverne, og al forretningslogik, GUI-definitioner og GUI-opførsel blev udviklet på serveren. Efter nogle få måneder blev jeg en af kernekompetencerne inden for logikken og virkemåden af systemet, og fik de sværeste opgaver, samtidig med at jeg hjalp andre udviklere med deres problemer. Samtidig fik jeg ansvaret for et medicinmodul i dette system.

Teknologien var et tredjeparts produkt, der var udviklet i Norge, og al udvikling foregik som nævnt i SQL på Microsoft SQL-servere. Dette gav mig en unik chance for at komme rundt i krogene af SQL, samt få databasedesign "ind under neglene".

Det stod ret hurtigt klart for mig, at det ikke var på den teknologi jeg skulle fortsætte. Så jeg begyndte at lære mig selv C++ for at udvide mine muligheder for at komme på andre projekter, og fordi det interesserede mig.

Maj 2001 – maj 2003

Dette resulterede i, at jeg i maj 2001 fik produktansvaret for en af CSC's større overbygninger oven på deres mainframesystemer. Med produktansvar menes der øverste ansvarlige udvikler. Antallet af udviklere på projektet svingede mellem 1 og 5.

Produktet var et grafisk framework skrevet udelukkende i C++. Det sørgede for at implementere nogle forretningsregler, men vigtigst af alt skabte det et framework til andre udviklere (uden for projektet) til at lave skræddersyede systemer, der arbejder på data i mainframen. Databindingen og kontakten til mainframen foregik via et sideløbende projekt, der lavede middleware-delen.

Der er reelt set 4 sideløbende projekter:

- Det grafiske framework med integrationen, som jeg havde ansvaret for.
- Mainframeudviklerne
- Middlewareudviklerne
- De systemudviklere, der anvender det grafiske framework til at udvikle kundespecifikke og -orienterede systemer med.

Disse 4 projektteams er hele tiden i tæt forbindelse, og samarbejder om ændringer.

Juni 2002 – februar 2003

Det stod ret hurtigt klart, at ambitionsniveauet for dette grafiske framework har været skudt langt over målet; kompleksiteten stod slet ikke mål med det nuværende og fremtidige anvendelsesniveau, og ændringer og fejlrettelser tog uforholdsmæssigt lang tid at lave. Så jeg begyndte at presse på, godt bakket op af de andre udviklere, projektledere samt produktspecialister, for at få det skrevet om. Vi opstillede nogle konsekvenser ved at fortsætte med det gamle system, sammenholdt med den langt mindre kodebase en omskrivning ville resultere i. Vores time-to-market i forbindelse med rettelser og udvidelser ville blive markant mindre. Dette, sammenholdt med vores estimater for at have et nyt system med tilsvarende funktionalitet, gjorde at vi fik lov til at starte omskrivningen i juni 2002. I mellemtiden blev arbejdet på det gamle system neddroset til udelukkende fejlrettelser.

På det tidspunkt var Microsoft lige kommet med version 1.0 af deres .NET platform. Vi brugte nogle uger på at komme "ind under huden" på denne teknologi, og det stod ret hurtigt klart, at vi skulle anvende C# frem for Java eller C++. Samtidig fik middleware-gruppen blod på tanden, og påbegyndte en omskrivning af deres projekt, også i C#.

Projektet lykkedes næsten; da vi havde knap 3 uger tilbage på udviklingsplanen, og ellers havde overholdt vores estimater, skete der organisatoriske ændringer, og den nye mellemlidelse fik kolde fødder. Resultatet var desværre, at efter cirka 8 måneders udviklingstid stod vi med et næsten køreklart projekt med rigtig gode fremtidsudsigter, men projektet blev lukket.

Juni 2003 – oktober 2004

Da CSC så påbegyndte deres nye, store satsning inden for næste generation af kliniske systemer, blev jeg headhunted over i det projekt. Teknologien her var Java – J2SE på client, J2EE, Oracle Enterprise Suite og OC4J på server – og der skulle bygges en ny generation af kliniske systemer oven på Oracles teknologi.

Oracle var i gang med at lave noget nyt inden for healthcare, og vi valgte at anvende denne teknologi. Dette skete fordi Oracle var i udviklingsfasen af dette og behøvede professionelle partnere inden for domænet. Samtidig havde CSC så en mulighed for at påvirke Oracle så systemet bedre passede til vores (og europæiske) formål.

Men jeg kom med i et core-team af arkitekter, der skulle designe og implementere client og server oven på dette system. I starten var jeg med til at lave nogle proof-of-concepts ned mod serveren, da det var helt ny teknologi. Blandt andet hvordan vi på et overordnet plan skulle håndtere optimistisk låsning, clusterservere samt caching af data mellem server og client.

Efter en måned gik jeg sammen med to andre arkitekter om at designe og implementere frameworket til klienten. Vi satte os det mål at lave frameworket således, at de udviklere der senere skulle udvikle grænsefladerne og forretningslogikken, overhovedet ikke skulle bekymre sig om adgangen til server/database, udvikling af GUI-kontroller, etc. Nøgleordene var:

- database- og servertransparens: for en udvikler så det ud som om alle data var på klienten.
- en uniform og nem adgang til data.
- Ingen udvikling af specialkontroller – dem tog vi os af, og hvordan de interagerede med resten af frameworket.
- Styring af layout samt skærbilleder.
- Bruger- og anden konteksthåndtering (heriblandt adgangskontrol til de enkelte data).
- udviklerne, der anvendte frameworket, skulle udelukkende spekulere på forretningslogik og GUI-design

Projekteamet var på sit højdepunkt på over 40 mand.

November 2004 – maj 2005

Da frameworket var på plads, og systemudviklerne kunne komme i gang, blev jeg flyttet over på et andet projekt. Dette projekt er et medicineringsmodul skrevet i C# til .NET 1.1, og jeg blev i kraft af mine erfaringer med .NET platformen, samt erfaring generelt, sat ind som brandslukker, da projektet var kronisk bagude. Jeg skulle blandt andet også hjælpe med at se, om der var arkitektoniske aspekter, der kunne laves om fordi de havde problemer med performance.

Kamstrup A/S

Juni 2005 – august 2006

Hos Kamstrup A/S kom jeg på et projekt, hvor jeg i første omgang fungerede som systemudvikler; projektteamet havde ikke så meget erfaring med GUI-programmering og performanceoptimeringer som jeg havde, så det blev min opgave at lære fra mig, og rette fejl og uhensigtsmæssigheder. Da softwaren skulle køre på Windows CE var performanceforbedringerne både i forbindelse med hastighed, men også med hensyn til hukommelsesforbrug.

Projektet er et system der kører på PDA'er, og er skrevet i .NET Compact Framework v. 1.1. Grundlæggende er det en applikation, der kan anvendes til at aflæse målere (primært varme- og elmålere). Disse data importeres så i et af vores andre systemer til videre behandling. Kommunikation til målere foregår via radio eller optisk kommunikation, og integrationen til andre systemer via GPRS eller Microsoft ActiveSync. ActiveSync har dog nogle problemer med transport af store datamængder over TCP/IP-protokollen, så jeg implementerede vores egen protokoloverbygning på TCP/IP der kunne omgå disse problemer.

Det oprindelige projekt bar præg af forhastede beslutninger, minimal styring, og diverse knopskydninger. Desuden havde der været en del krybende krav der var målrettede en bestemt kunde, og alle var enige om at det burde refaktoreres. Det blev mit ansvar som arkitekt på dette projekt at sørge for at designe og implementere et framework, samt sørge for maksimalt kodegenbrug og nem integrering med vores andre systemer. I oktober 2005 påbegyndte jeg designet, og udviklerne og jeg begyndte implementeringen i januar 2006 (designet kørte jeg sideløbende resten af tiden).

På dette tidspunkt var Microsoft lige kommet med .NET version 2.0, og jeg besluttede med det samme, at projektet skulle laves i dette. Dette skete fordi jeg kunne se en masse fordele i det nye framework, hvor udviklere blev mere produktive, samt vi fik mulighed for at komme over på Microsoft PocketPC platformen (dette var ikke muligt før grundet vores leverandør af PDA'er). Projektet er blevet færdigt til tiden, og bliver i denne tid solgt til kunder i hele Europa.

Samtidigt med dette har jeg haft ansvar for at koordinere udviklingen af et submodul med vores udviklingsafdeling i Norge. Dette submodul bliver anvendt i det norske system, i PDA'en, samt det store aflæsesystem, som bruges til fjernaflæsning af værker med adskillige tusinde målere. Modulet præsenterer, efter redesign og reimplementering, et ensartet interface til alle de målerprotokoller vi understøtter – både åbne standarder, samt proprietære protokoller.

September 2006 – december 2006

I dette tidsrum har jeg været ansvarlig for omskrivning af et af vores submoduler, som førnævnte modul anvender. Det er en implementation af en bestemt type transportprotokol, og omskrivningen skulle sikre, at modulet passede ind i vores nye modultankegang. Samtidig udviklede jeg en overbygning, der kunne drage fordel af nogle nyudviklede ting i hardwaren.

Januar 2007 og fremefter

Nu sidder jeg så på Kamstrups store satsning – et nyt aflæsesystem der skal afløse alle de tidligere, bortset fra PDA'systemet - som udvikler. Her er jeg blandt andet ude ved kunder for at høre dem om ønsker, og for at se hvordan de anvender de nuværende systemer. Dette har resulteret i den første version af en prototype på grænsefladen.

Sideløbende hos Kamstrup

Sideløbende med mine “reelle” opgaver hos Kamstrup A/S, har jeg:

- Været med til indførelse af en dokumentationsmodel (en standard 3-lags dokumentationsmodel).
- Været drivende faktor i indførelsen af automatiske unittests i samtlige projekter i system-software-afdelingen. Samtidig indførte jeg en fælles standard for unittests (NUnit) på tværs af projekterne, så der var mere ensartethed.
- Indført kodestandarder – på et pragmatisk niveau – således at vi får mere ensartethed i sourcekoden på tværs af projekterne. Dette er vigtigt, da man ofte springer mellem projekterne, og derfor er det en fordel hvis sourcekoden har en ensartethed på tværs af projekter.
- Indført et bugtrackingsystem, og står for den daglige drift af dette.
- Været med til at komme med inputs til vores nye modultankegang. Tidligere havde vi en smule kodegenbrug, men dette var jeg med til at standardisere.
- Været med til at skabe erfaring med .NET 2.0; det projekt jeg sad på var det første, der anvendte denne version, og alle projekter – bortset fra et enkelt nicheprojekt – er nu opgraderede.
- Valgt og indført værktøj til UML-diagrammering på tværs af projekterne, og står som ansvarlig for at opgradere eller udskifte hvis behovet opstår.
- Været ansvarlig for et par andre submoduler, som implementerer ofte benyttet funktionalitet på tværs af projekterne.
- Presset på for at skabe mere og bedre samarbejde mellem hardware- og softwareudvikling, så der fanges flere fejl så tidligt som muligt.

Privat

I privat regi arbejder jeg med følgende problemstillinger, fordi emnerne og kompleksiteten interesserer mig:

- neurale netværk
- genetiske algoritmer og maskinel evolutionær programmering
- datakomprimering og data-entropi

I forbindelse med genetiske algoritmer er jeg i gang med at definere mit eget programmeringssprog, som er nemt for computeren selv at programmere i. Sproget er et funktionelt programmeringssprog i stil med Haskell og F# (F-Sharp - et funktionelt programmeringssprog til .NET-plattformen. Det er indtil videre ikke kommet videre end Microsofts research-afdeling, men har dog et lille community online).

Jeg har lavet en implementation af NUnit (automatiske unittests) til .NET 2.0 compact frameworket, da et sådan ikke eksisterer.

Jeg er i øjeblikket i gang med at lave mit eget revisionskontrollsystem (i stil med SubVersion og Microsoft Source Safe), mest fordi jeg ikke har prøvet noget lignende tidligere, og der er nogle interessante problemstillinger.

Jeg har også lavet software til et Ph.D.-forskningsprojekt på Århus Universitet. Softwaren præsenterer nogle inputs for en forsøgsperson, der ligger i en MR-scanner (magnetic resonance scanner). Forsøgspersonen skal foretage et valg ud fra disse inputs, og softwaren måler meget præcist responstiderne og svarene. Disse data kan eksporteres, og sammenholdes med resultaterne fra scanningen. Formålet er, kort fortalt, at se hvilke områder af hjernen der er aktive, mens forsøgspersonen bliver præsenteret for inputs og overvejer svarmulighederne.

Professoren, der lavede dette forsøg, har nu fået bevilliget en større pose penge, og kan derfor fortsætte sin forskning i emnet. Dette vil afstedekomme, at jeg skal til at arbejde sammen med ham igen for at skrive nyt software. Planen er at øge præcisionen i softwaren, samt lave noget databehandling i selve softwaren. Den oprindelige software er skrevet i C++, men jeg overvejer mulighederne for at anvende C# i næste omgang, da præcise tidsmålinger via Windows API'et er blevet bedre i .NET 2.0 end det var i tidligere versioner.