Færøsk migrering med kultursammenstød og timestamp-problemer

Proof of concept for mainframe-migrering virkede, men hvordan går det på Færøerne, når hele skattesystemet skal konverteres til en Windows-platform?

Dan Mygind tip@version2.dk Mandag, 6. januar 2020 - 5:03 16

Dette er den tredje og sidste del i Version2's fortælling om det store færøske migreringsprojekt væk fra mainframen.

Et lille agilt udviklerteam har arbejdet koncentreret på at få udviklet de nødvendige værktøjer og softwarekomponenter, så det færøske skattevæsen Taks' mainframe-baserede systemer kan blive konverteret til en Windows-platform.

Et tidligere migreringsprojekt blev indstillet grundet manglende fremdrift, men efter det lille teams vellykkede proof of concept bliver der givet grønt lys til at genstarte Færøernes største migreringsprojekt.

Nu er det alle PL/1-programmerne, som skal transformeres, og derfor inddrages Elektrons PL/1-udviklere i arbejdet, da det skal sikres, at transformeringen ikke introducerer fejl i de samfundskritiske systemer.

PL/1-udviklernes arbejdsform ligger milevidt fra det lille agile teams, hvilket kommer til at give visse gnidninger.

Læs også: Skattesystem omlægges fra PL/1 på mainframe til C# på Windows

Kultursammenstød

De mere end 40.000 PL/1-programmer indeholder blandt andet filer med definitioner af konstanter og datastrukturer, som en præprocessor sammensætter til et egentligt PL/1-program.

Da en mængde PL/1-programmer transformeres første gang, er PL/1-udviklerne forfærdede, da de ser resultatet.

I de konverterede C#-programmer indgår de samme datastrukturer igen og igen i mange forskellige programmer.

Det betyder, at hvis der skal laves en ændring i en datastruktur, skal PL/1-udviklerne rette hundredvis af steder. Det er uholdbart vedligeholdelsesmæssigt.

PL/1-udviklerne kan heller ikke finde kommentarer fra PL/1-koden i C#-programmerne, hvilket er med til at underminere PL/1-udviklernes tro på projektet.

For det lille agile team har øvelsen imidlertid været en succes, da der er kommet kørende kode ud af transformeringen. De har valgt en depth-first udviklingsfilosofi for at komme hele vejen igennem alle lagene og se, at tingene virker, selvom der er meget, som ikke er perfekt.

»Vi forsøgte at køre agilt, iterativt, og vi var i research-mode. Vi ønskede så hurtigt som muligt at komme igennem med noget kode, så koden var ikke så pæn, men det hjælper ikke, hvis vi

bruger et halvt år på at få koden til at se pæn ud, og så finder ud af, at vi ikke kan få det hele til at virke,« forklarer Torbjørn Lisberg.

Den slags hændelser var del af det kulturchok, som opstod, da de to grupper begyndte at arbejde sammen.

Som direktør for Elektron Ulla Joensen udtrykker det:

»Der var et vidt spænd mellem de to grupper, så det gjaldt om at få dækket spændet mellem folk, som var begejstrede, og folk, der var bekymrede. Det gælder om at få alle til at rumme hinanden i processen. Normalt er mainframe-udvikling meget stringent og velstruktureret. Der er nogle forretningsregler, som skal implementeres i et stabilt driftsmiljø. Mainframe-folkene har mange systemer i drift, som skal vedligeholdes, så det var forståeligt, at de var bekymrede.«

Torbjørn Lisberg var lead developer på kodetransformatoren P2C. (Illustration: Dan Mygind)

Projektleder med migrerings-erfaring

For at skabe de rette rammer for det store migreringsprojekt bliver Frodi Hansen hentet ind som projektleder. Han var ansat på Elektron fra 1995 til 2007 og har derfor kendskab til systemer og organisationen.

Desuden har han tidligere haft ansvaret for store projekter, blandt andet migrering af alle de færøske pengeinstitutter fra Elektron til SDC i 2010.

»Jeg kunne gå ind og se på projektet som helhed og identificere risici med den erfaring, jeg havde med migreringsprojekter. Det havde været et meget teknik-drevet projekt hidtil, så det var svært at få et overblik. Jeg fik lidt realisme ind i projektet, så det var klart, at det ikke kun handlede om, at der skulle laves en kode-transformator,« siger Frodi Hansen og fortsætter:

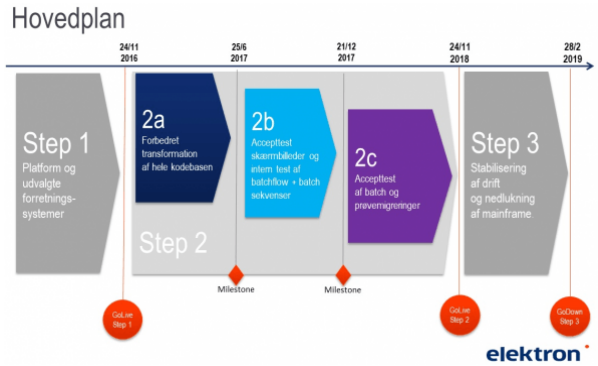
»På det tidspunkt, jeg overtog projektet, havde, det været meget teknik-drevet, og kløften mellem den lille agile udviklergruppe og mainframe-udviklerne var stor – meget stor,« siger Frodi Hansen og understreger samtidig, at de forskellige persontyper var nødvendige for at få projektet i mål.

»Der er nogen, der kommer med en vild idé. Det er sådan, man kommer til Månen. Jeg kom så ind og hjalp med at samle nogle ting op, som gjorde, at det kunne komme i mål,« siger Frodi Hansen, der starter på projektet i april 2016.

Der udarbejdes en ny plan, hvor projektet deles op i 2 faser. Step 1 var at gå i produktion med en delmængde af det samlede systemkompleks – ca. 4 pct. af systemer, men 85 pct. af den underliggende teknologi.

Formålet er at konstatere, at projektet er muligt at gennemføre, platformen er moden, og driftsorganisationen i stand til at drifte løsningen.

Frodi Hansen arbejder tæt sammen med PL/1-udvikleren Ingi Hentze, der kommer til at fungere som testmanager, da det står klart, at en



Det store migreringsprojekt blev opdelt i 3 hovedtrin. (Illustration: Elektron)

grundig test er nødvendig for at undgå, at fejl kommer til at lamme det færøske samfund.

Detaljeret systemkortlægning

Hvor Frodi Hansen udarbejder de overordnede planer, går Ingi Hentze i gang med en detaljeret kortlægning af de enkelte delsystemers opbygning og afhængigheder.

»Først skitserede vi det overordnede, og derefter gik vi detaljeret ned i de omkring 40 forskellige delsystemer og deres forskellige programmer. Hvad består batch-delen af, og hvad består online-delen af ? Hvad er input, og hvad er output? Hvilke systemer læses der fra, og hvilke eksterne systemer skrives der til? Hvilke eksterne interessenter er der? Vi kortlagde de sammenhænge, der var internt i vores systemer og eksternt med andre virksomheders og organisationers systemer.«

Det er et kæmpe arbejde, der kræver involvering af en mængde personer.

»Alle delsystemer har en ansvarlig udvikler, så de vidste det meste, og så havde vi møder med driftsfolk, kundeansvarlige og udviklere. Hvad ser kunden? Hvilke output, lister og rapporter skal vi tjekke er korrekte efter konverteringen? Hvilke data sendes til eksterne interessenter som SDC, der skal tjekke, at data er korrekte? Vi fik de rigtige folk på banen, så vi kunne identificere, hvad der kan gå galt, og hvad vi skal være obs på,« forklarer Ingi Hentze.

Elektron har mapper af systemdokumentation skrevet gennem de seneste 20-30 år, så den detaljerede systemkortlægning bliver en mulighed for at genopfriske, hvordan systemerne hænger sammen, og få opdateret systemdokumentationen med den detaljerede systemkortlægning.

Migrerings-step 1 – tre små pengeinstitutsystemer, som ikke er flyttet til SDC, og et barselssystem fra Taks-systemet – sættes i gang.

Hvor skal vi rette?

Prøvemigreringen går overordnet set fint og giver tillid til, at konverteringsprocessen virker. Den viser dog også, at transformationen til C# ikke er perfekt.

Det fører til overvejelser om, hvorvidt det bedst kan betale sig at ændre i transformeren, i PL/1-koden, eller om der eventuelt skal ændres i den konverterede C#-kode.

Den sidste mulighed er stort set no-go, da eventuel re-transformering af PL/1-koden vil overskrive den tilrettede C#-kode.

Bliver der ændret i transformeren, er det nødvendigt at transformere alle PL/1-programmerne igen, for at sikre at der ikke er opstået uforudsete sideeffekter ved en ændring, og her anvendes unit test og regressionstest i stor stil. Det er en lang og omstændelig proces, som giver migreringsprojektet øgenavnet migræneprojektet.

»Der gik næsten 2 år, fra vi gik live med step 1, til vi går live med resten,« siger Frodi Hansen, der undervejs skal sørge for at opretholde et godt samarbejde mellem det lille agile udviklerteam, der fokuserer på at få fungerende kode, og så mainframe-udviklerne, der har stort fokus på vedligehold og drift.

To år er en lang proces, hvor gnidninger ikke kan undgås.

Frustrationsråbende projektleder

C#-udvikleren Torbjørn Lisberg kommer til at sidde mellem to stole i den proces.

Dels arbejder han med den lille kernegruppe på at få alle de nødvendige komponenter til at spille på Windows-platformen og få transformeren til at spytte C#-kode ud; dels skal han prøve at opfylde mainframe-folkenes ønsker om en pæn, velstruktureret og velkommenteret kode.

»Det er personer, som er meget kompetente og ser på verden ud fra deres synspunkt,« siger Frodi Hansen, som skal fremstå som den rolige og velafbalancerede projektleder, der skaber harmoni i en projektgruppe med mennesker, der har vidt forskellige baggrunde og erfaringer.

Frustrationerne bygger sig dog op, og Frodi Hansen er blevet set gående udenfor Elektrons kontorbygning, der grænser op til et naturskønt område, hvor får græsser fredeligt, for at råbe sine frustrationer ud.

Ingen får blev skadet.
[>>media:44561:Det store migreringsprojekt blev opdelt i tre hovedtrin.]
Trods frustrationerne kommer transformeren efterhånden til at virke til alles tilfredshed, også for PL/1-udviklerne, der får pæn C#-kode inklusive kommentarer fra PL/1-programmerne.

Nyfødte disrupter prøvemigrering

I sensommeren 2018 er migreringsprocessen og alle komponenterne klar til den endelige migrering.

De mange delsystemer har vist sig at være for sammenkoblede til, at man kan migrere ét system ad gangen, så der planlægges med en big bang-migrering, hvilket ikke er en ideel situation, men den eneste farbare vej frem.

»Det ville være meget svært at dele det op, så vi besluttede at tage resten i ét hug,« siger Frodi Hansen.

Det vurderes samtidig, at det efter to dages kørsel på den nye platform ikke vil være muligt at rulle systemerne tilbage til den gamle platform.

Det er et højrisiko-migreringsprojekt, og derfor bliver der holdt to generalprøver på den store migrering.

Det går overvejende godt, men ikke helt.

To nyfødte gav problemer, forklarer Ingi Hentze.

»Første gang vi havde en prøvemigrering, var data ikke ens mellem det gamle system og det nye. Det viste sig, at to nyfødte havde skabt problemer: Den del, der har med Folkeregistret at gøre, ligger inde i Taks-systemet, mens det nye Folkeregistersystem opdaterer ind i maven på det gamle system, og vi havde ikke fået lukket den adgang,« siger Ingi Hentze.

Ellers er alt planlagt ned til sidste detalje i testdrejebogen.

»I drejebogen stod det hele. Selv hvem der skulle lave morgenmad, og hvem der skulle sende pressemeddelelse ud og så selvfølgelig alle de systemmæssige tests,« siger Ingi Hentze, der dog nævner en enkelt svipser:

»Det første forløb viste, at vi havde glemt én ting: Dørene til Elektron er låst i weekenden, så testerne kunne ikke komme ind. Det kom så med i drejebogen til næste gang,« siger Ingi Hentze.

Konspirationsteori om migrering

Weekenden 23–25. november er sandhedens time.

Det er stort set hele det færøske samfund, der på den ene eller anden måde er i berøring med systemerne, så omkring 70 repræsentanter fra Skat, pengeinstitutter, virksomheder og andre interessenter mødes i Elektrons lokaler for at være med til at teste, at migreringen foregår korrekt.

Elektron havde udsendt pressemeddelelse om migreringsprojektet, men der er ikke den store mediebevågenhed om projektet.

Som symptom på informationssamfundets tendens til historier baseret på halve sandheder opstår der dog også en konspirationsteori om migreringen.

Det offentlige udbetaler løn, pension og ydelser en uge tidligere end normalt for at minimere risikoen for en katastrofe, hvor færingerne ikke kan få udbetalt penge, hvis migreringen fejler.

Det vil give Elektron en lille måned til at rette op på tingene, i værste fald.

Da det falder sammen med Black Friday, bliver der blandt almindelige færinger spekuleret i, at det offentlige udbetaler penge tidligere end normalt for at støtte Black Friday.

Det er dog ikke noget der optager de mange forsamlede i Elektrons kontorer på Staravegur 9 i Thorshavn, hvor de arbejder koncentreret gennem en nervepirrende weekend.

Migreringen går over al forventning, og det er en række lettede medarbejdere, der sammen med direktøren for Elektron, Ulla Joensen, kan se tilbage på et hårdt og opslidende, men også succesfuldt projekt.

»Det var et projekt, der handlede om vores egen overlevelse. Hvis vi ikke havde gjort noget, så ville vi have mistet vores kunder, så vi var nødt til at gøre noget – det var et spørgsmål om liv eller død,« siger Ulla Joensen.

PL/1-udviklerens oplevelse

Den 67-årige PL/1-udvikler Páll Beder har i dag rettet i mange af C#-programmerne, og overgangen fra PL/1 til C# har ikke været for besværlig, selvom han stadig føler sig mest hjemme i PL/1.

»Det er ikke så nemt som PL/1, men jeg har også arbejdet med PL/1 i mange år. Jeg synes ikke, C#-koden er så læsbar som PL/1, men her et år efter er det ikke så slemt,« siger Páll Beder, der startede med at arbejde med skattesystemet tilbage i 1984.

Han fremhæver, at transformeren laver en – for PL/1-udviklere – venlig C#-kode.

»Vi kan finde procedure-navne fra PL/1 i koden, og så har de beholdt datastrukturer, hvilket gør det nemmere at forholde sig til,« siger Páll Beder.

Hos Elektron kalder man det for PL# grundet de bibeholdte datastrukturer fra PL/1 og en kode uden C#-shortcuts. Det er C#-programmer, som enhver C#-programmør vil genkende, selvom de måske vil gøre tingene lidt anderledes.

»Vi har haft en C#-programmør inde og lave ændringer i C#-programmerne, og det er svært for os at forstå den kode. Der er anvendt shortcuts som '?' i stedet for if...else. Det skal man selvfølgelig bare vide, men der er mange af den slags ting i hans kode, der er forskellig fra PL#,« siger Páll Beder.

Páll Beder er i dag glad for, at Elektron migrerede, da udviklingsmiljøet i Visual Studio er markant bedre end mainframen.

»Det er en meget stor fordel, at vi kan udvikle og køre lokalt. Vi laver et program lokalt og kan hele tiden se, hvad der sker. Vi kan klikke på en variabel og se indholdet af den. Det har vi slet ikke i PL/1. Der kører vi det hele igennem og laver en masse udskrifter, så vi kan se, hvor der sker noget. Nu kan vi lave en trace lokalt og i test-miljøet, inden det

Hvordan laves goto i C#?

Det var en udfordring at transformere PL/1's goto til C#. Der findes goto i C#, men ikke med så vilde muligheder som PL/1's.

Med Goto i PL/1 kan du springe helt ud af kontekst, fra en PL/1-kodeblok til en anden. I C# bliver du indenfor scope.

Størstedelen af PL/1 labels var dog indenfor scope og er derfor lavet om til helt almindelige C# labels, som det ses i nedenstående kode-eksempel

»Der var dog nogle goto, som vi blev nødt til at løse med en ikke så pæn løsning. Vi laver en try-catch, hvor konverteren laver exceptions svarende til goto-labelen, og så kan vi smide en goto-exception som catches,« siger Torbjørn Lisberg.

kommer over i produktion. Det er en meget stor forskel,« siger Páll Beder.

Vejen dertil har dog været lang.

»Det var en meget lang proces. Det tog vel omkring seks år inklusive Raincode-projektet og vores egen in-house-udvikling. Vi var langt nede. Jeg var træt af at tale om konvertering, konvertering, konvertering,« siger Páll Beder og fortsætter:

»Jeg sagde til den årlige medarbejdersamtale med Ulla: 'Vi må se at få den konvertering ud af verden, jeg gider ikke mere'. Det var forfærdeligt, vi syntes, det gik for langsomt.«

Den erfarne systemprogrammør

En anden erfaren Elektron-medarbejder er systemprogrammøren Olavur Lydersen, som havde en central rolle i det store migreringsprojekt.

Olavur Lydersen har arbejdet hos Elektron siden 1979 og er en af den slags systemprogrammører, som ikke gør meget væsen af sig selv, men som er afgørende for, at en infrastruktur fungerer.

Brænder system-lokummet, går man til ham.

Han har udviklet i Assembler, PL/1 og har et indgående kendskab til CICS, JCL, CA Scheduler og andre mainframekomponenter, som han hjalp det lille agile udviklingsteam med at forstå og få tilsvarende komponenter til at virke på Windows-platformen.

```
//CODE SNIPPET: PL1 DO LOOP WITH LEAVE STATEMENT
for (J.Set(1); (J <= 10); J.Set(J + 1))
{
    for (I.Set(30); (I >= 1); I.Set(I + -1))
    {
        if (Builtin.Substr(lyk[J].TAINCLY2.LYKIL, (I), 1) != " ".ToChar() &&
            Builtin.Substr(lyk[J].TAINCLY2.LYKIL, (I), 1) != Builtin.Low(1))
            goto leave; // PL1 LEAVE
    }
    leave:;
}
```

Kodeeksempel fra det færøske migreringsprojekt. (Illustration: screenshot)

Batchjobs er en vigtig del af et mainframe-driftsmiljø, og i forbindelse med Raincode-projektet var man begyndt at arbejde med produkterne JAMS og NEOBatch, der modsvarer henholdsvis CA Scheduler og JCL på mainframen.

»Vi analyserede alle job, og deres afhængigheder blev beskrevet i XML-filer. Heldigvis var der allerede lavet en del forarbejde i forbindelse med Raincode-projektet,« siger Olavur Lydersen, der har svært ved at svare på, hvad den største udfordring var for ham.

»Der var så mange ting, så mange komponenter,« siger Olavur Lydersen og nævner konvertering af karaktersæt fra mainframens EBCDIC til Windows-platformen som et problem, der blev ved med at dukke op i forskellige sammenhænge.

Aftestningen af batchjob tog også tid, ligesom der var en del print-udfordringer. Her blev Olavur nødt til at udvikle små print-handlere til både Cypress-print-produktet og Neobatch.

Hurtigere svartider og timestamps som nøgler

Elektron har nu kørt over et år på den nye platform, der består af omkring 45 Windows-servere, som er delt op i et test-miljø, et præproduktionsmiljø og så et egentligt produktionsmiljø.

»Det kører meget stabilt, og det har gode svartider,« siger Olavur Lydersen, der med lettere affektion i stemmen tilføjer, at den gamle mainframe fra 2006 også havde gode svartider, selvom det kneb lidt på det seneste, da Elektron valgte ikke at opgradere den.

Faktisk kører de konverterede systemer markant hurtigere, hvilket gav problemer grundet den eksisterende datamodel, hvor timestamps blev anvendt som nøgler i databasen.

Tilbage i 1984 havde man ikke problemer med at anvende en timestamp, hvor mindste tidsenhed er millisekunder som nøgle for transaktioner. Den forbedrede performance skaber dog problemer, da flere records nu forsøges at blive gemt under den samme entydige nøgle.

Løsningen bliver at lægge kunstige forsinkelser ind i systemerne.

Elektron planlægger sammen med Taks at starte et nyt projekt i 2020, hvor et af de gamle systemer gennemgås, gen-modelleres og får ny brugergrænseflade.

»Nu kan kunderne begynde at udvikle mere moderne og fremadrettet,« siger Ulla Joensen, der også glæder sig over besparelsen i softwarelicenser.

```
//CODE SNIPPET: HANDLE PL1 GOTO LABEL WITH C# EXCEPTION, WHEN LABEL IS LOCATED IN A CALLING PL1 FUNCTION
void Run()
{
    try
    {
        DB2ERROR()
    }
    catch (END)
    {
        // PL1 "END" LABEL
    }
}

void DB2ERROR()
{
    throw new END(); //GOTO PL1 "END" LABEL
}
```

Kode-eksempel på implementering på goto-funktion i C#. (Illustration: screenshot)

Vi vil hjælpe andre

»Vi sparer omkring fire millioner kroner år på licenser alene. Dertil kommer besparelser som vi har opnået, da det ikke har været nødvendigt at købe opgraderinger til mainframen,« siger Ulla Joensen og oplyser, at hele projektet, inklusive Raincode-projektet, har kostet 39 millioner kroner.

Færingerne har fået krigssår og projekt-traumer som konsekvens af det mangeårige projekt, men også hvad der tegner til at være et solidt fundament for migrering af mainframe-programmer til Windows.

Det er oplagt at spørge, om de vil kommercialisere værktøjerne så andre kan få glæde af deres indsats.

»Lige efter migreringen skulle vi lige trække vejret. Nu er der gået over et år, og vi føler os trygge og klar til at se på den mulighed. Det kunne være sjovt at finde en kunde og lave et proof of concept sammen med dem,« siger Ulla Joensen.

De involverede fremhæver, at det ikke er et produkt, man blot installerer og så trykker på en knap, hvorpå transformeringen foregår automatisk. Det er en række veltestede værktøjer, som skal sælges med konsulentydelser. Færingerne har taget de store slag, så de forventer ikke at der vil være tale om en så lang proces som de selv har været igennem.

Taler man mainframeprogrammeringssprog kan man ikke komme udenom COBOL. Har færingerne nogen planer om at lave en Cobol-transformator?

»Lad os først prøve med nogen, der har PL/1 i stedet for at bruge en masse krudt på Cobol. Det vil være sjovt, hvis vi kan hjælpe andre, som det har hjulpet os selv,« siger Ulla Joensen.

Version2 var inviteret til Færøerne af Elektron.