

2

Kontroll på Gammal Mjukvara?

- Intervju med Jens Byléhn, EIS by Semcon och Daniel Berggren, Combitech

5

Kommunikation i Inbyggda System

6-7

Nyheter - Utbildningar

- Verifiering av Mjukvara - Design av Medicintekniska Produkter
- Applikationer i Inbyggda System - Programmering & Utveckling

8-9

ARMs Utvecklingsmiljö

- ARM Workbench, ARM RealView Debugger
- ARM Profiler - Realtids trace, Profilerad kompilering

10

En Modern Kravmiljö

- IRQA - en process där kravhantering är nyckeln till framgång

11

Embedded Fokus med Nohau

- Seminarium **Embedded Fokus** den 26-29 maj i Stockholm, Oslo, Göteborg och Kolding
- Övriga evenemang under 2009: Webinars - Konferenser - Mässor

Hur får man Kontroll på Gammal Mjukvara?

När vet man om det är värt att vidareutveckla på gammal kod? Är det dags att skriva om delar? Vad behövs för att vi ska kunna återanvända delar av vårt gamla system?

Det var några av de frågor vi ställde till Jens Byléhn, på EIS. Han arbetar med denna typ av problem ute i olika organisationer.



Det finns en kritisk punkt bortom vilken man riskerar att helt förlora förmågan till nyutveckling.

Jens Byléhn, Sr. System Arkitekt, EIS

- Man kan säga att man inte hinner ro eller täppa till hålen för att man har fullt upp med att ösa. Antalet kunder och kodens storlek samverkar, och underhållet växer kvadratisk över tiden, säger Jens.

- För att täppa till hålen måste man vid en viss nivå på underhållet fundera på ett omtag. Samtidigt kan man inte, med en konkurrensutsatt verksamhet, bara stänga butiken i väntan på en stor omdesign, säger Jens.

- När det blir uppenbart att den kritiska punkten är nådd, är det oftast försent, eftersom du har en fördröjning på 6-12 månader innan ett omtag slår igenom ute hos dina användare. Under tiden fortsätter underhållet att växa, säger Jens.

Vad gör man då för att upptäcka om man nått den kritiska punkten?

- De akuta symptomen märks i organisationen. Medarbetare gnäller på tråkiga arbetsuppgifter och personalomsättningen är stor, speciellt när konjunkturer vänder. Nya funktioner tar dessutom allt längre tid att bli klara, och releaser blir liggande för att man måste integrera nya rättningar, säger Jens.

- Men det går att se symptom tidigare. Den underliggande orsaken till problemen är att systemets komplexitet ökar sannolikheten att fel skapas, säger Jens.

Berätta om de tjänster ni erbjuder era kunder!

- Vi jobbar med effektiv mjukvaruutveckling, det omfattar koden, verktygen men även arbetssätt, säger Jens.

- Just för den här typen av problem jobbar vi med refactoring. Vid refactoring mäter och visualiserar vi komplexiteten i form av beroenden i våra kunders kodbas. Utifrån våra etablerade mönster och tekniker kan vi föreslå förändringar som gör att det skapas färre fel, säger Jens.

Daniel Berggren från Combitech berättar att han i ett projekt insåg att de måste göra koden lättare att underhålla.

Vi intervjuade ett antal personer i teamet. Det framkom att koden var svår att förstå, den var inte längre särskilt modulär och det var uppenbara kvalitetsproblem.

För att förstå var problemen egentligen låg, gjordes en arkitekturanalys med verktyget Lattix som åskådliggör kod i form av matriser som visar olika beroenden.

- Väldigt snabbt kunde jag få upp ett par bilder att visa management. Det var lätt för alla att se att arkitekturen innehöll en mängd oönskade beroenden, säger Daniel.

- Jag kunde också visa på hur en förändring på ett ställe skulle komma att påverka stora delar av systemet, säger Daniel.

- Det var nu relativt enkelt att se vilka förändringar vi måste göra för att bryta upp oönskade beroenden. Med Lattix fick vi koden mer modulär, vilket underlättar underhållet, säger Daniel.



Dessutom fick vi vår systemarkitektur dokumenterad på köpet!

Daniel Berggren, Combitech

Rhapsody + Eclipse = Sant

Den senaste versionen av Rhapsody innehåller ett antal nya spännande funktioner. Här finns förbättringar som hjälper både systemingenjörer, mjukvaruutvecklare och testare att förbättra kvalitet och dokumentation på sitt arbete.

De nya funktionerna innebär t ex att Rhapsody kan köra under Eclipse, vilket betyder att du kan arbeta både med modell och kod under Eclipse, [se fig 1](#).

Vidare har det blivit enklare att köra rapid-prototyping med de nya grafiska verktygen i Rhapsody. Något som väsentligt kan spara projekttid när du grafiskt kan åskådliggöra krav och funktionalitet.

Ytterligare nyheter:

- **IBM Rational Test RealTime Integration** – Utvecklare kan validera och verifiera sina designs genom en integration mot Rhapsody TestConductor.
- **IBM Rational Team Concert Integration** - För dig som vill dra nytta av Jazz och Eclipse-plattformarna för CM-hantering.

IBM **Rational** software

- **Grafisk verktygsfåra** – Gör det enkelt att skapa ett prototyp-GUI som simulerar, validerar och kommunicerar din designs beteende.
- **Förbättrad Reverse Engineering och Roundtripping** – Redan bra blir ännu bättre, då fler C/C++ konstruktioner hanteras i den nya versionen.
- **Mer stöd för Autosar** - Bland annat en ny diagramtyp för att specificera arkitekturer för Autosar, möjlighet att styra minnespartitionering.



Fig. 1) Rhapsody kan nu köra under Eclipse.

Demo på Eclipse-integrationen och de andra nyheterna:
<http://modeling.telelogic.com/products/rhapsody/whats-new.cfm>

Kodgranskning

Gartner-rapport rekommenderar kodgranskning och statisk analys

Att fånga programfel tidigt är den viktigaste åtgärden om du vill hålla utvecklingskostnaderna nere. Det slår en rapport från Gartner fast.

Rapporten konstaterar vidare att:

- Kodgranskning och statisk kodanalys, tillsammans med en daglig byggprocess, kommer att resultera i väsentligt högre kodkvalitet.
- Granskningar i team skapar en gemensam förståelse och förbättrat samarbetet i projekten.
- Statisk kodanalys tillför en konsistent kodningsstandard.

Rapporten rekommenderar därför att:

- Man utvecklar en daglig process som kräver statisk analys och kodgranskning i team, innan någon tillåts checka in ny kod till ett bygge.
- Man kombinerar statisk kodanalys med kodgranskning för att få bästa djup och bredd i kodanalysen

Källa: Gartner rapport G00158218



Don't check in code that doesn't work!

Gwyn Fisher, Klocwork

GPLv3 licenser ökar

Black Duck Software har samlat in tiotals miljarder rader med öppen källkod från cirka 4000 webbplatser



Black Duck Software har den mest omfattande databasen av öppen källkod och annan nedladdningsbar kod. De har upptäckt över 1400 unika programvarulicenser, men över 85% av öppen källkod är licensierad under sex licenser - [se tabell](#).

License	July 2008	January 2009
GPLv2	57.74%	53.15%
LGPLv2.1	10.70%	10.00%
Artistic	9.85%	9.10%
BSD	6.15%	6.34%
GPLv3	1.86%	4.06%
Apache	2.78%	3.62%

GNU General Public License version 3.0 (GPLv3) släpptes i juni 2007 och används i mer än 7000 projekt. Enbart i december 2008 tillkom över 400 nya projekt. Fenomenet har dock inte fått stor uppmärksamhet i media.

För några månader sedan passerade GPLv3 Apache att bli den femte vanligaste licensen. Om nuvarande ökningstakt håller i sig kommer den, inom 12-18 månader, att bli vanligare än BSD 2.0 licensen.

Under de senaste sex månaderna har användningen av GPLv2 minskat mer än 4% i open source-projekt. Istället används mer GPLv3, BSD, Apache och MIT. Branschens mest kända open-source projekt, Linux, kommer att fortsätta med GPLv2.

Copyleft-licens

GPLv3, liksom GPLv2 är en copyleft-licens. Lite grovt kan Copyleft sägas vara motsatsen till upphovsrätt, och är till för att säkerställa tillgången till källkoden, och inte skyddet av koden som privat egendom (som copyright gör).

Centralt för en copyleft-licens är en ömsesidig skyldighet. Ömsesidig innebär att om du skapar ett arbete baserat på GPL programvara, och du distribuerar den, då måste du distribuera det resulterande arbetet under GPL.

Med andra ord, du är skyldig att ge användaren källkoden för det arbete och alla rättigheter att modifiera och distribuera den kod som du fick under GPL. GPL version 2 och 3 kräver att du publicerar källkoden för bearbetningar om du distribuerar din kod.



Copyleft-licens är en ömsesidig skyldighet

Joakim Nilsson, Nohau

Flera bolag har ifrågasatts och några har även blivit stämda för att de inte efterlever kraven på GPL-licens; Cisco, Verizon, Skype, BT, D-Link, TomTom med flera.

Ny licens - Affero GPL

Utvecklare av webbapplikationer, som faktiskt inte distribuerar koden, kan ändra i öppen GPL källkod utan att behöva göra den tillgänglig för sina kunder.

För att täppa till detta kryphål har en ny licens skapats, Affero GPL (AGPL). AGPL kräver att man måste dela med sig av källkod som används i en klient-server tillämpning, så kallad Software-as-a-Service (SaaS). Det är 165 projekt som hittills har anammat Affero GPL-licens.

Mer info?

Kontakta Joakim Nilsson:
joakim.nilsson@nohau.se,
040-592208



Kommunikations- lösningar



Bluetooth, CAN, USB

Protokollanalytatorer och testare för:

- Bluetooth, Bluetooth Robustness
- CAN, Ethernet, RS-232
- USB, UWB, Wireless USB ZigBee



För industriautomation och fordonsindustrin:

- CAN, CANopen,
- DeviceNet,
- Ethernet Powerlink,
- EtherNet/IP
- FlexRay, J1939, LIN



Industriell Ethernet

Ixxats Industrial Ethernet modul är en flexibel och kostnadseffektiv lösning för integration av Industrial Ethernet i dina produkter.

Modulen erbjuder dig ett universalt interface för användning av ett flertal Industrial Ethernet teknologier. Du får lägre komponentkostnad och kan använda samma hårdvara till olika utvecklings- och produktionslinjer.

Protokoll som stöds: EtherCAT (Slave Controller), EtherNet/IP (Adapter Device), Powerlink (Controlled Node) och PROFINET RT (IO-Device).

Mer info?

Kontakta Markus Lai:
markus.lai@nohau.se, 040-592224

ARM-utvecklare?

Med RealView Development Suite får du ut det bästa av ARM-baserade processorer:

RealView
Tools by ARM

- ✓ Full support för hantering av målsystem med multicore
- ✓ Unik streaming Trace och profilering
- ✓ Höghastighets-debuggning och Trace support för CoreSight
- ✓ Full support för alla ARM-processorer

Kontakta Nohau

Markus Lai:
markus.lai@nohau.se
+46 40 59 22 24



RealView ICE och Trace 2 kopplad till en ARM plattform för ARM1176JZF-S

Nyheter - Utbildning

Verifiering av Mjukvara Applikationer i Inbyggda System Programmering & Utveckling

Develop Linux Based Embedded Systems

Kursmål

I denna kurs går vi igenom en fullständig uppsättning "Best of Breed" verktyg för att utveckla ett inbyggt Linux-system. Tonvikten ligger på att använda och konfigurera den bredd av funktioner och applikationer som finns tillgängliga för Linux, i kombination med din egenutvecklade applikation.

Deltagare

Om du snabbt och effektivt vill komma igång med utvecklingen av ett inbyggt Linux-system, då är detta en kurs för dig. Kursen är också mycket användbar för dig som vill veta mer om detta heta ämne, och som vill utvärdera om Linux är rätt alternativ för dina framtida embedded-projekt.

Praktiska övningar/verktyg

Vi använder PC som värd för utvecklingsmiljön och ansluter till ett ARM-baserat målsystem (OMAP5912 OSK). På target kör vi Embedded Linux.

Vi utvecklar och debuggar applikationen genom att använda både hårdvarudebugger (Lauterbach) och mjukvarudebugger (t ex GDB).

Pris och schemalagda datum

- Pris: 1680 Euro
- Datum & Ort: 18-20/5 Malmö

Lauterbach

Lauterbach TRACE 32 är ett utvecklingsverktyg utformat för komplexa applikationer, och finns tillgängligt för ett antal olika processorer.

Vi erbjuder följande Lauterbach-kurser:

- **Lauterbach Debugging** - *Ger dig en flygande start med ICD (In-Circuit Debugger).*
 - Datum & Ort: 28/4 Göteborg, 5/5 Stockholm
 - Pris: 780 Euro
- **Lauterbach Trace Functions** - *Fördjupa dina kunskaper och lär dig att använda TRACE32 på ett effektivt sätt.*
 - Datum & Ort: 29/4 Göteborg, 6/5 Stockholm
 - Pris: 780 Euro
- **Lauterbach for Linux Based Systems** - *Debugga ett Linux-system, där du drar nytta av både GDB och Lauterbach (mjukvaru- och hårdvarudebugger).*
 - Datum & Ort: Kontakta kursansvarig
- **Lauterbach for DSP Based Systems** - *För att debugga familjen DSP 64xx.*
 - Datum & Ort: Kontakta kursansvarig

Kontakta kursverksamheten

Marica Manecke-Jansson, mmj@nohau.se, 040-592216
Louise Vinternatt, lv@nohau.se, 040-592223
www.nohau.se/training.asp



Med ett flertal **nyheter** i vår kursportfölj, erbjuder vi nu **80 kurser**.

Skulle du inte hitta det du söker, kontakta mig.



Marica Manecke-Jansson



Objektorienterade Metoder

- Practical Modeling with UML, 3 dagar. Kursen ges på begäran.
- MBSE Architecture, 2 dagar. Nästa kurstillfälle: 2/4 samt 15/4 i Linköping.
- MBSE - Design, Execution and Code Generation, 2 dagar. Nästa kurstillfälle: 16/4 samt 29/4 i Linköping.
- MBSE Simulation and Model-Based Testing, 2 dagar. Nästa kurstillfälle: 30/4 samt 15/5 i Linköping.

Design of Medical Devices

- Basic Regulatory Training for Design Engineers, 1 dag.
- Architecture of Medical Devices Containing Embedded Systems, 2 dagar. Båda kurserna ges på begäran.

Agila Metoder

- Certified Scrum Master, 2 dagar. Ges på begäran.
- Lean and Agile for Managers. Kursen är på 2 dagar och ges på begäran.
- Scrum for Team Members, 1 dag. Ges på begäran.

Kurskalender

Programming skills

Software Architecture for Embedded Systems:

- 24-25/3 Stockholm
- 13-14 Jönköping
- Pris: 1470 Euro



Safe Programming Practices:

- 14-15/4 Stockholm
- Pris: 1400 Euro

Programming Language & Development

Reviewing C Code:

- 16-17/4 Stockholm
- Pris: 1300 Euro



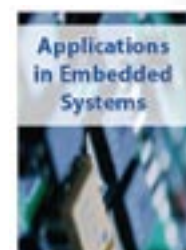
C Engineering:

- 6-8/4 Stockholm
- Pris: 1600 Euro

Applications in Embedded Systems

Develop Linux Based Embedded Systems:

- 18-20/5 Malmö
- Pris: 1600 Euro



Develop for ARM Cortex-M3:

- 7-8/4 Göteborg
- Pris: 1200 Euro

Verification of Software / Quality and Test

Lauterbach Debugging:

- 28/4 Göteborg
- 5/5 Stockholm
- Pris: 780 Euro



Alla kurser finns på:
www.nohau.se/training.asp

ARMs utvecklingsmiljö - R

Traditionellt sett har ARM som begrepp varit synonymt med en IP-modell. Så är det inte längre. Idag kan du köpa hela utvecklingsmiljöer och verktyg från ARM - från kompilering, debuggning, profilering till optimerad kompilering - där kompilatorn tweakar koden beroende på resultatet från målsystemets profilering.

I denna artikel presenterar vi ARMs verktygssvit Real-View Developer Suite (RVDS) och de moduler som finns i denna miljö.

Med RVDS får du ARMs kompletta verktygssvit för mjukvaruutveckling, med full support för ARM7, ARM9, ARM11, samt hela Cortex-serien. Verktygssviten består av tre huvudkomponenter: **Workbench**, **Debugger** och **Profiler**, se fig. 1.

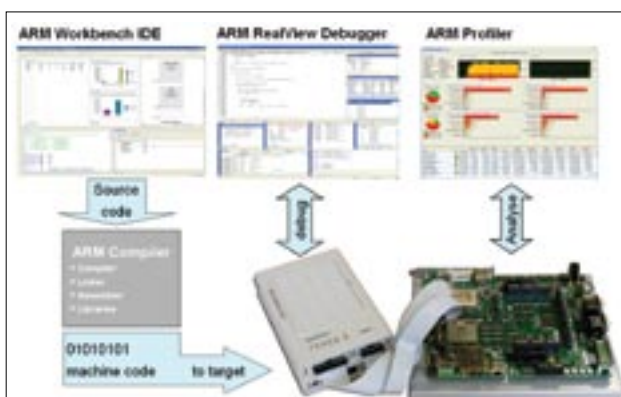


Fig. 1) De tre ingående huvudmodulerna i RVDS. Då de är integrerade, finns full transparens och stöd mellan dem. Hela miljön är eclipse-baserad, vilket gör den flexibel och generell.

ARM Workbench

Med projekthanteraren i ARM Workbench kan du som utvecklare skapa, bygga, debugga, programmera flash, profilera, traca och kontrollera C/C++ projekt för alla typer av ARM-baserade målsystem.

ARMs C/C++ kompilator är hårt optimerad och stödjer alla moderna versioner av ARM, Thumb, Thumb-2 instruktionsset, samt NEON SIMD instruktionsset med vektorserade NEON-kompilator.

Med Workbench har du full support för:

- Projekthanterare och ARM Linux projekthanterare
- Flashprogrammering
- ARM Assembler editor och ARM Profiler
- Editor länkad minnesmap
- Editor för källkodskonfigurering
- Loggning av exekverad kod på instruktionsnivå

ARM RealView Debugger

Med ARMs RealView Debugger har du full support för hantering av målsystem med multicore. Du kan koppla upp dig mot inte mindre än 64 kärnor simultant, och samtidigt öppna ett debuggfönster för var och en av dessa.

Med ARMs debugger hanterar du:

- Lågnivådebuggning
- Registerhantering
- Minnehantering och uppmappning
- Stack analys
- Variabler - lokala och globala
- Breakpoints - program och läs/skrivbrytpunkter
- Multicore debuggning
- C++ inline funktioner på header-filer
- Accelerated Image loader
- Struct hantering

RealView Developer Suite

ARM Profiler – Realtids trace

Nytt i denna svit är ARMs tillämpning av trace, där du med realtids-trace kan spela in programflöden över flera dagar. Det innebär att du kan mäta och spela in från ett verkligt system där interaktion och påverkan från användare inkluderas, [se fig. 2](#).

Tekniken bakom bygger på att data om programflödet streamas över till värddatorns hårddisk, vars storlek sätter gränsen för inspelningstiden. Exempelvis kan du övervaka och spela in systemets avbrottsrutiner och studera hur statistiken på kodtäckningen förändras. Verktöget stödjer bland annat:

- Realtidsinspelning av trace till PC hårddisk (host)
 - 1TB data => 1–10 dagar (!) Krävs alltså ingen dyr tracebuffert. Dessutom är resultatet tillgängligt direkt, utan onödig laddningstid mellan host target.
- Realtidsanalys - Du kan studera:
 - Minnesläsningar och skrivningar
 - Antal funktionsanrop per tidsenhet
 - Antal avbrott per tidsenhet
 - CPU användning och Kodtäckning

Profilerad kompilering

För att kunna hantera långtidsinspelningar av fullt optimerad ARM-kod, finns stöd för profilerad kompilering.

Genom profilerad kompilering optimeras koden beroende på hur den körs på målsystemet. Det förbättrar avsevärt programvarans prestanda och responsen hos applikationen.

Tekniken går ut på en återkoppling av resultatet på profileringen och resultatet används sedan till omkompilering av koden. Genom att iterera denna process kommer koden att anpassas allt bättre till maskinvaran och på det sätt du faktiskt exekverar mjukvaran på, [se fig. 3](#).

Ordningsgång för profilerad kompilering:

- 1) Kompilera din kod till applikationen.
- 2) Kör applikationen för att samla in statistik, baserad på just denna version av kompileringen. Observera att detta kan göras både på faktiskt target och med ett referenssystem.
- 3) Den insamlade statistiken analyseras och en rapport skapas. Denna kan matas in direkt till kompilatorn, som nu tar hänsyn till körningen och optimerar koden på de områden där de statistiskt framtagna flaskhalsarna finns.

Optimeringen sker antingen på prestanda i fråga om storlek på binär, exekveringshastighet eller på både och. Systemet kan ställas in så att detta sker automatiskt tills ingen eller väldigt liten skillnad finns i prestanda mellan två efter varandra följande körningar. ■



Fig. 2) Snippet från ARMs profileringsverktyg.

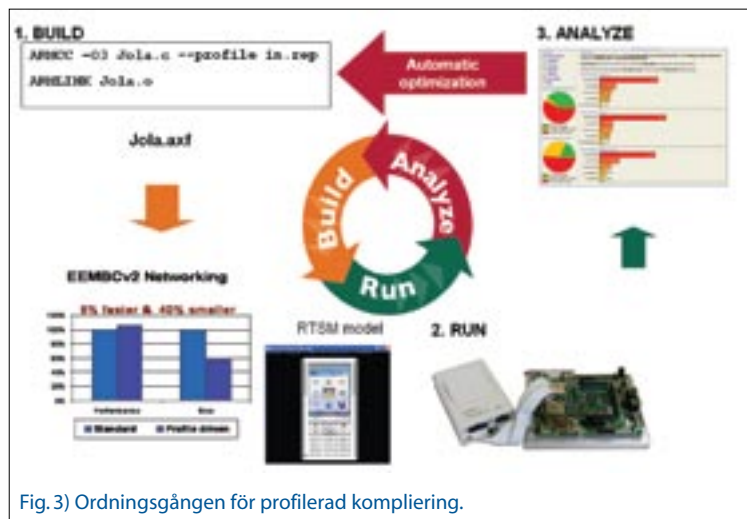


Fig. 3) Ordningsgången för profilerad kompilering.

En Modern Kravmiljö

IRQA - en process där kravhantering är nyckeln till framgång



Att beskriva det som ska utvecklas eller produceras, och sedan följa förändringarna under produktens livstid är en av det absolut viktigaste processerna inom en verksamhet.

Mängder av analytiker har under de senaste tio åren pekat på att just kravdefinition och kravfångst är de områden som måste vidareutvecklas. Tyvärr har många företag ännu inte insett hur allvarligt detta är, istället lever många kvar i ett manuellt - pappersbaserat kravarbete utan någon konkret spårbarhet.

Att utifrån detta försöka göra en snabb konsekvensanalys är inte ens tänkbart. Men, med en mer automatiserad kravmiljö och starkt verktygsstöd förändras denna bild och ger ökad flexibilitet, spårbarhet och kontinuerlig visibilitet.

Intresserad av Utbildning?

Vi erbjuder grundläggande och avancerad kurs för IRQA: Requirements Engineering with IRQA.

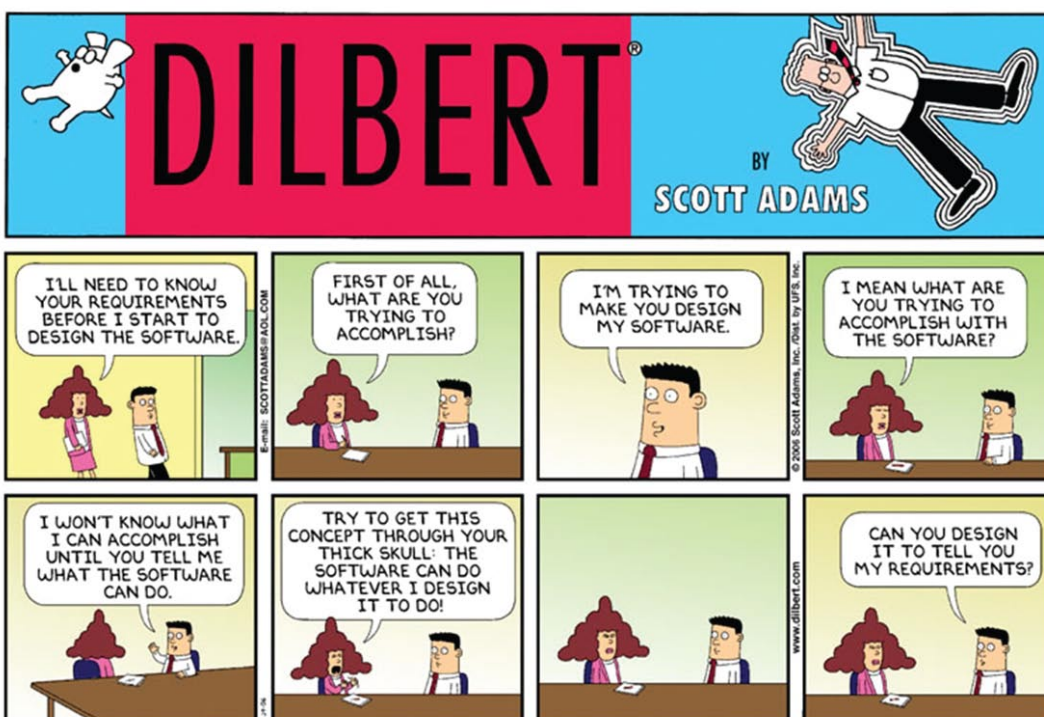
Kontakta Marica Manecke-Jansson för mer information: mmj@nohau.se, 040-592216.

Mer kunskap om IRQA?

Kontakta **Ulf Sandberg** för mer information: ulf.sandberg@nohau.se, 0707-88 04 14.



Du har väl inte missat att vi erbjuder ett flertal webinarer, samt seminariet "Get Your Requirements Right" den 22-24/4 se: www.nohau.se/events.asp.



© Scott Adams, Inc./Dist. by UFS, Inc.

Att utveckla programvara är en fortgående process med många hinder

Embedded Fokus Seminarium i maj

Nu slår vi upp portarna för seminariet Embedded Fokus

Under vecka 22, med start 26/5, besöker vi Stockholm, följt av Oslo, Göteborg och Kolding i Danmark.

Några av de ämnen som lyfts fram:

- Modellbaserad utveckling - Lyssna på projektledarnas erfarenheter från olika projekt.
- De senaste teknologierna för att underlätta debuggning.
- Refactoring - hur får jag koll på min arkitektur?
- ARM - en teknologi i förändring.
- Kvalitetssäkra från start.
- Kom igång med Embedded Linux.

Agenda, registrering och övrig information kring konferensen hittar du här:  <http://www.nohau.se/nec2009/>

Välkommen med din anmälan!

Nohau Nordic Event Calender

SEMINARIER

Embedded Fokus

- 26/5-29/5 ([se info till vänster](#))

Get Your Requirements Right ([se även sid. 10](#))

- 22/4 i Stockholm
- 23/4 i Göteborg
- 24/4 i Köpenhamn

WEBINARS

Supporting Re-usability and Product Families with IRQA

- 31/3, 16:00 (CET)

Requirements Based Testing: Control your tests with IRQA

- 28/4, 16:00 (CET)

Lattix: Architectural Analysis of Java and .NET systems

- 2/4, 15:00 (CET)

Lattix: Architectural Analysis of C/C++ embedded systems

- 23/4, 15:00 (CET)

Lattix: Dependencies Beyond Code: The Big Picture View of Enterprise Systems with Java and Oracle

- 14/5, 15:00 (CET)

MÄSSOR

Rational Day

- 21/4, Stlm, Kista Entré, Knarrarnäsgatan 7
- 28/4, IBM Göteborg, Lindholmens allé 10

EuroStar

- Stockholm 30/11- 3/12

Embedded Conference Scandinavia

- Stockholm 21/10-22/10

Alla evenemang finns på:
www.nohau.se/events.asp

Paketerade tjänster

Rådgivande Ingenjörbyrå

Ja, det skulle man kunna kalla en av våra paketerade tjänster. Vad innebär det då?

Vi har arbetat med utvecklingsmiljöer och -plattformar hos hundratals kunder. Det har givit oss en erfarenhet vi vill dela med oss av. Helt enkelt vill vi vara ditt naturliga val av rådgivare. Här följer två exempel:

1) Plattformsbyte: Gammal 8051 ska bytas ut mot ny Cortex M-3

Ett medelstort företag är osäkert på vad de ska tänka på när de ska byta plattform. De ringer oss. Vi ger råd. Det kan gälla val av processor, mjukvarans arkitektur, minnen eller införande av open source mm.

2) Prototypkort: Allt i ett

Ett företag inom automation vill snabbt komma igång med en ny produktserie. Display, CAN, Ethernet är några krav. Nohau hjälper till att hitta bästa lösningen för prototypkort, inklusive drivers och RTOS. Kunden väljer senare att även använda prototypplattformen i serieproduktion.

Vill du veta mer om våra tjänster?



Kontakta Magnus Engström:
magnus.engstrom@nohau.se,
040-592204

Open Source

LSG - Linux Support Group

Har du problem att få igång Linux på din plattform? Funderar du på vad det skulle innebära att föra in Linux i din produkt?

Vill du ha generell utbildning/workshop eller mer handfast hjälp att sätta upp just ditt system? Det kanske saknas någon komponent som du vill ha hjälp att utveckla?

Utnyttja vår tjänst, Linux Support Group (LSG) för att få igång Linux i inbyggda system. Tjänsten är till för dig som snabbt vill nå resultat tillsammans med erfarna och pedagogiska Linux-experten.

Vi har resurser för att hjälpa till i alla delar av ett Linux-införande, oberoende om du tänkt använda open-source verktyg eller kommersiella verktyg.

Mer info?

Kontakta Joakim Nilsson:
joakim.nilsson@nohau.se,
040-592208



nohau

nohau solutions ab
Box 1030
SE-212 10 Malmö Sweden
Tel: +46 40 59 22 00
Fax: +46 40 59 22 29
www.nohau.se