

RENDEZVOUS

Nr 1 april 2001

Innehåll

Ordföranden har ordet	3
SESAM-medlem får utmärkelsen "2000 Computer Pioneer Award"	4
Kort introduktion till pekare	6
TA-13 Progressive Acquisition reviderad	6
Software Technology Conference – konferenssäsongs höjdpunkt inom militär programvaruteknik	7
Riksdagen godkänner "sexnationsavtalet"	9
Ett systems uppgifter som underlag för kravarbete	10
Vilken väg går kvalitetssäkringen?	13
Utredningarna om försvarsteknikens framtid nu klara	15
Systems Engineering standarden 15288 närmar sig målnöret	16
Seminarium ISO/IEC 15288 System Life Cycle Processes	18
SNART och Högskolan i Halmstad arrangerar RTiS2001	19
Hemsidor	20
Kalendern	20

Försvarssektorns Adaintressenters Användargrupp för Software Engineering

SESAM

Vad är SESAM?

SESAM har tillkommit för att organisera och stimulera samarbete och samverkan inom programvaruområdet mellan försvarsindustrin, FMV och FOA.

Det avtalsfästa syftet med SESAM är "att genom organiserat samarbete mellan användargruppens medlemmar främja tillförlitlighet och effektivitet i utveckling och vidmakthållande av programvarusystem i Ada inom försvarssektorn". Inom ramen härför skall SESAM även anpassa, profilera och förnya sin verksamhet med hänsyn till ändrade tekniska och andra omständigheter av betydelse för intresseområdet.

Följande kommer att ske under den närmaste 2-3-årsperioden.

1. SESAM skall allmänt verka för att sprida information om faktorer som påverkar möjligheterna till tillförlitlig och effektiv utveckling och vidmakthållande av programvarusystem. Särskilt skall härvid Adas betydelse i sammanhanget klargöras.

2. SESAM skall i sin verksamhet fortlöpande bevaka möjligheterna att samla, skapa och sprida information om objektiva mät- och andra resultat och erfarenheter vunna vid användning av "software engineering"-principer och Ada.

3. SESAM behandlar tillvägagångssättet vid utveckling och vidmakthållande av programsystem. Implicit i detta ligger givetvis att använda processer skall tillförsäkra de resulterande produkterna efterfrågade egenskaper. Produktens egenskaper som påverkas av processerna är därför av primärt intresse att bevaka i SESAMs verksamhet.

4. SESAM skall i sin verksamhet fästa stor vikt vid att underlätta samexistens mellan Ada-program och programvara skriven i andra språk. Speciellt skall aspekter vid användning av COTS beaktas.

5. SESAM skall där så är möjligt sätta konkretiserade och mätbara mål för sin verksamhet under avgränsade tidsperioder.

SESAM styrs av ett Råd med representanter för gruppens medlemmar. Rådet har till sin hjälp ett Verkställande Utskott (VU) och ett sekretariat.

Rådets ordförande är Claes Wadsten, AerotechTelub, tel 013-231652 .

VU

Andersson Tommy, Ericsson Microwave Systems AB
tommy.andersson@emw.ericsson.se

Bengtsson Christopher, FMV
chben@fmv.se

Brandt Roger, FMV
robra@fmv.se

Carlsson Ingemar, adjungerad
ingemar.carlsson@mbox2.swipnet.se

Folkesson Dag, Saab Aerospace AB
dag.folkesson@saab.se

Johansson Billy, CelsiusTech Electronics AB
bijo@infomatics.saab.se

Merkell Curt, Saab Bofors Dynamics AB
curt.merkell@dynamics.saab.se

Wadsten Claes, AerotechTelub AB
claes.wadsten@aerotechtelub.se

Arbetet utförs i två arbetsgrupper:

Ag Metodik

Håkan Edler, CTH/Datorteknik
edler@ce.chalmers.se

Ag Teknik

Lars Asplund, Uppsala Universitet
asplund@docs.uu.se

Vilka kan vara med i SESAM?

Medlemmarna i SESAM är svenska företag, organisationer och myndigheter (förvaltningar, utbildningsinstitutioner etc) med anknytning till försvarssektorn. Medlemmarna indelas i följande kategorier

- ordinarie medlemmar
- arbetsgruppsmedlemmar
- informationsmedlemmar.

Enskild person kan endast komma ifråga som informationsmedlem.

Inträde i SESAM

För samtliga medlemskategorier gäller att inträde beslutas av Rådet.

För inträde som ordinarie- och arbetsgruppsmedlem krävs status som leverantör till FMV. Dessutom krävs en skriftlig förbindelse att uppfylla åtagande som ordinarie- och arbetsgruppsmedlem.

För inträde som informationsmedlem (erhåller endast informationsbladet) krävs status som leverantör till FMV eller status som myndighet inom totalförsvaret. Rådet kan emellertid anta annan part som informationsmedlem.

För ansökan om medlemskap i SESAM vänd er till sekretariatet.

SESAM-Sekretariatet

AerotechTelub AB
c/o Kåsjös Kontor
Ytterspåret 14
187 54 TÄBY

Ordföranden har ordet

Vi har nu startat ett nytt verksamhetsår med höga ambitioner. Vi hoppas att seminarier, projekt och "mikroprojekt" skall leda till nytta för våra medlemmar. Vi har dessutom fått ett antal "tung" nya medlemmar till detta år.

Trots våra höga ambitioner så har vi inte fått den rivstart på verksamheten som vore önskvärd. Detta på grund av att vårt avtal med deltagande industrier är inriktade så att finansieringen av eget arbete är vår medlemsavgift och att försvarsmakten bekostar kansli och vissa projektmedel.

Då vi inte fått tillgång till dessa medel finns ett visst avvaktande från industrin som drar ned på det arbete som redan nu borde startats.

Projektet omvärldsbevakning blir också mycket lidande på grund av dessa förse-ningar då vi inte kan planera deltagande i mässor och konferenser utan får lita på att medlemmarna besöker dessa i sin ordinarie verksamhet. Då SESAM normalt brukar bidra med viss finansiering för att få tillgång till redovisningen så kan denna utebli tills finansieringsfrågan är löst. Vi har nu förlorat 3 månader och kommer att förlora ytterligare tid för detta.

Jag hoppas att vi snart får en lösning då FMV sagt att SESAM:s verksamhet är viktig och att man på alla sätt skall stödja verksamheten både finansiellt och med ett aktivt deltagande. Jag hoppas att vi snart får detta bekräftat så att inte mer tid går förlorad.

Med detta uttalande från FMV så hoppas jag att medlemmarna snarast sätter ingång med de projekt inom Metodik- och Teknikgrupperna som ändå skall göras inom ramen för medlemskapet för att vi skall få intressanta kontakter mellan deltagande företag samt ett välfyllt seminarie i höst.

Som ordförande vill jag därför be alla deltagande företag att starta sina projekt inom ramen för medlemskapet. Kontakta därför ordförande i respektive grupp om detta inte är gjort med förslag på mikroprojekt så att vi får "snurr" på verksamheten. Allt tyder på att FMV:s del i SESAM kommer att lösas inom en snar framtid.

Med hälsningar
Claes Wadsten
Ordf. SESAM

Björn Källberg avled på långfredagen efter en längre tids kamp mot cancer

Björn var den främste Ada-expert vi haft i landet och hade en stor del i försvarsindustrins framgångar på programvaru- och Adaområdena. Björn var också Sveriges främsta ansikte utåt i Ada-sammanhang, med ett långvarigt engagemang som svensk representant i de standardiseringsorgan som arbetat fram de successiva Ada-standarderna. Förutom sina insatser i SESAM, där han var rådsordförande när sjukdomen drabbade honom, och i Ada i Sverige, var han styrelseledamot i Ada Europe och medlem i olika internationella arbetsgrupper och organisationskommittéer. Hans kompetens, skicklighet och förmåga att sakligt motivera sina åsikter, var starkt bidragande till att Sverige blev erkänt och respekterat i Ada-världen.

Björns frånfälle kom inte direkt oväntat, men är lika smärtsamt för det. Vi kommer att minnas honom med stor tacksamhet och saknad.

SESAM-medlem får utmärkelsen ”2000 Computer Pioneer Award”

“to recognize and honor the vision of those people whose efforts resulted in the creation and continued vitality of the electronic computer industry”.

Som säkert många av Er har noterat i andra sammanhang, kan (Harold) Bud Lawson, medarbetare i Rendezvous och talare vid SESAM-seminarier, till sina tidigare utmärkelser, som bl a fellow i både ACM och IEEE, nu lägga titeln Computer Pioneer. IEEE Computer Socitey har tilldelat Bud ”2000 Computer Pioneer Award”. Utmärkelsen delas ut årligen till “outstanding individuals whose main contribution to the concepts and development of the computer field was made at least fifteen years earlier”.

Utmärkelsen delades ut den 1 februari i Ft Lauderdale, Florida i samband med en stor IEEE-konferens.

Bud får utmärkelsen för sin uppfinning av pekarvariabeln. I en följande essä berättar Bengt Asker, en av pionjärerna på hemmaplan, om pekarvariabelns innebörd och betydelse.

Vi återger här först tillkännagivandet från IEEE i sin helhet:

Harold W. (“Bud”) Lawson has, by the Board of Governors of the Institute of Electrical and Electronic Engineering’s Computer Society, been selected to receive the year 2000 Computer Pioneer Award in honor of his seminal contributions to the field of computer science.

This award is given for significant contributions to early concepts and developments in the electronic computer field which have clearly advanced the state-of-the-art in computing. The contributions must have taken place fifteen or more years earlier. The award consists

He joins a select group of approximately 100 pioneer award recipients. Previous recipients have included J. Presper Eckert and John Mauchly, the inventors of the ENIAC,; Gene Amdahl and Fred Brooks, inventors of the IBM System/360; Grace Murray Hopper, inventor of higher level programming languages and compilers; Seymour Cray for Cray Supercomputers; as well as Jack S. Kilby the winner of the year 2000 Nobel Prize in Physics for his invention of the integrated circuit.

The motivation for the award is as follows:

“For inventing the pointer variable and introducing this concept into PL/I, thus providing for the first time, the capability to flexibly treat linked lists in a general-purpose high level language”



of a bronze metal. The award ceremony will take place in Fort Lauderdale, Florida in February.

The work on the pointer variable concept was performed during the years 1964-65. At that time Bud Lawson was employed by the International Business Machines Corporation. After the invention of the pointer variable and its integration into PL/I, he authored the first professional publication of these concepts in the article "PL/I List Processing" which appeared in the June, 1967 issue of the Communications of the Association for Computing Machinery.

As with many of the pioneering inventions, such as the integrated circuit, the pointer variable has been an important enabler. The integrated circuit has enabled the development of advanced computer and telecommunications hardware. The pointer variable has enabled programmers to effectively create higher level language programs to solve many complex problems in, amongst other areas, computer graphics and systems programs such as compilers and operating systems. Unquestionably the pointer variable concept, later adapted from PL/I to various programming languages such as C, Pascal, C++, and Ada has had a major impact upon the field of computing.

When properly utilized, pointer variables have provided for elegant solutions to complex computational tasks. One of the most notable successes in this regard was the development of what many consider to be the best operating system ever developed; namely MULTICS which was developed jointly by the MIT (Massachusetts Institute of Technology) and Bell Telephone Laboratories in the late 1960s. The Operating System was pro-

grammed using PL/I with heavy reliance upon the pointer variable facilities. During its commercial life with General Electric and later Honeywell, MULTICS proved to be one of the most understandable and maintainable operating systems ever.

When utilized in a non-professional manner, pointer variables have led to software reliability problems. Such non-professional usage is evident in enormous software composites containing significant unnecessary complexity such as the Windows operating system. There are undoubtedly many thousand pointer variables utilized in that system; whereas, there were only a few hundred in MULTICS. By not properly engineering software, complex pointer relationships and status can arise that are difficult to predict and trace thus leading to difficulties.

When will we ever learn that engineering discipline is required to produce high quality software!!! Pointer variables when properly utilized have provided hundreds of thousands of software engineers with an essential enabling tool of the trade.

A description of the Computer Pioneer Award as well as a complete list of previous recipients can be found at: www.computer.org.

For further information contact:

Harold W. Lawson
Lawson Konsult AB
Björnvägen 7
181 33 Lidingö
SWEDEN

Telephone and Fax +46 8 7677831

E-mail bud@lawson.se

Buds utmärkelse hör givetvis till det finaste en datormänniska kan få. Rendezvous sällar sig till gratulanterna och är stolt att ha Bud bland sina medarbetare.

Kort introduktion till pekare

En pekare är precis vad ordet säger, det är en variabel som pekar på någonting. Enkelt uttryckt en minnesadress. I de första programmeringsspråken (Fortran, Algol) hanterade man större datamängder i minnet med arrayer vilket är mycket klumpigt och inte minst otydligt och svår-genomskådat. Den första förbättringen kom med introduktionen av poster, dvs gruppering av samhörande data. Typfall: Namn, Adress, Telefonnummer. Det kom med Cobol och var ett väsentligt inslag i Algol-Genius. Bud Lawsons pekare introducerade ett sätt att dynamiskt bygga upp datastrukturer i minnet. Genom att låta varje post innehålla en pekare till nästa post åstadkommer man s k länkade listor. Det gör det enkelt att skapa olika strukturer såsom stackar, köer, sorterade listor, träd etc. En finess är att detta sker utan att behöva flytta posterna, det är endast länkarna (pekarna) som behöver ändras.

PL/1 fick aldrig något riktigt genomslag utanför IBM-världen, men i det språk

som länge dominerade all systemprogrammering, C, spelar pekarna en väsentlig roll. En pekare i C får peka på vad som helst (i stort sett i alla fall) vilket gör att det går att skapa oerhört effektiva och kraftfulla program. Men all flexibilitet har ett pris. När en pekare kan peka på allt kan den också lätt peka fel. När ni råkar ut för "Programmet har utfört en förbjuden åtgärd och kommer att avslutas" är det med stor sannolikhet ett pekarfel. Typfallet är att programmet försöker använda en pekare som inte blivit ifylld och ofta pekar på adressen noll.

I nyare språk har reglerna för hur en pekare får användas därför skärpts. I Java användes inte ordet, man talar om referenser mellan objekt, men grundprincipen är densamma. Pekare är sedan länge ett omistligt inslag i all programmering.

Bengt Asker



TA-13 Progressive Acquisition reviderad

FMV med bistånd av försvarsindustrin och konsulter, har under ett antal år medverkat i ett WEAG-projekt med syfte att förbättra anskaffningsprocessen för "Defence Information Systems". Projektet resulterade i sin första fas bl a i ett förslag till sådan process kallad Progressive Acquisition. Resultaten med tillhörande handbok "Guidance on the Use of Progressive Acquisition" presenterades på ett seminarium som FMV anordnade i november 97. Projektet har sedan dess fortsatt i en andra fas som avslutades i december, där bl a teorierna i denna handbok prövats i praktiskt bruk. Med anledning av erfarenheter och kommentarer har handboken sedan uppdateras och föreligger nu i en version 2, november 2000.

Täckningsområdet för denna handbok är bredare än för bara informationssystem, varför de flesta SESAM-iter bör kunna ha behållning och nytta av den, inte minst i det klimat av ökat europeiskt samarbete som påbjudits, tex enligt sexnationsavtalet. Handboken finns att ladda ner från SESAM hemsida.

Software Technology Conference – konferenssäsongsens höjdpunkt inom militär programvaruteknik

Ingmar Ögren åker till Salt Lake City och har lovat rapportera till Rendezvous



Software Technology Conference i Salt Lake City 29 april - 4 maj 2001

2001 blir det fjärde gången för mig, varav den tredje som föredragshållare vid "Software Technology Conference" (STC) i Salt Lake City. Kan alltså vara dags att berätta lite för vännerna i SESAM om vad detta är och vad man kan förvänta sig om man åker dit. Det jag kan berätta här blir av naturliga skäl kortfattat, varför den intresserade även bör studera webbplatsen www.stsc.hill.af.mil och därifrån klicka vidare till STC under logon som syns ovan.

Varför åka till STC?

För oss som arbetar med programvara och programvaruintensiva system för försvaret finns det alltid anledning att tillbringa månadsskiftet april-maj vid STC i Salt Lake City. Beroende på inriktning kan man ha olika intressen, varav några kan vara:

- **Utbildning**

Här ingår en dags "tutorial" i konferensavgiften och man får chansen att lyssna på vad som är "det senaste" i USA.

Vidare får man se själva konferensen som ett tillfälle att lära mera om de mest skiftande ämnen inom programvaruområdet och få en uppfattning om vem som är expert på vad.

- **Träffa folk**

Det är här man får chansen att träffa folk, som man bara hört talas om eller som har

varit på snabbvisiter i Sverige. Adaexperten Curry Colcet, Dr. Barry Boehm, Joyce Tokar från DDCI och många, många fler. Dessutom är diskussionen lättsam i utställningen och runt föredragen så det finns goda chanser till nya bekansnader.

Man träffar för övrigt folk inte bara från USA utan även från Europa, Asien, Australien osv. Totalt handlar det om flera tusen besökare!

- **Kolla marknaden**

Själv har jag aldrig varit på en så stor utställning som vid STC och även ställt ut Tofs två gånger. Storleken beror på en medveten policy med måttliga kostnader för utställarna. Dessutom en väldigt professionellt genomförd utställning med servering av gratis kvällsmat i utställningslokalen.

Här ges chansen att skapa sig en uppfattning om läget på marknaden.

Årets tema

Årets tema handlar om att styra kostnader, tidsplaner och kvalitet. Således något som är nog så aktuellt även för oss i Sverige!

Inledande "tutorial"

Även om en del utställningar sätts upp redan på söndagen så är måndagen första egentliga dagen med tio parallella "tutorials", som håller på hela dagen. Ämnena anknyter väl

till årets tema med ämnen som processförbättring, ISO 9001, COTS, CMM, kravarbete, mätning, projektarbete mm.

Den första timmen efter lunch är det avbrott för öppningssessionen med Dr. Garber som är "Director of Interoperability" vid försvarsdepartementet.

Många parallella spår

Tisdag - torsdag pågår själva konferensen. De många parallella spåren gör det naturligtvis omöjligt att följa allt, men ger i stället en utmärkt chans att hitta något som passar just de egna intressena. Vad handlar då årets sessioner om:

- Samverkan (interoperability) med arkitektur, datahantering och organisatorisk påverkan.
- Processförbättring med organisatoriska aspekter, modern hantering av granskningar och genomgångar, lärdomar från bedömningar.
- Processmodeller, framför allt med utvecklingen av CMM och om hur man bör hantera förändring.
- COTS med erfarenheter, behov bedömningar, etc.
- Informationsteknologi med sekretess och skyddsaspekter, feltolerans, problem med objektorientering mm.
- CAC (Common Access Card)
- Projekthantering med sådant som att bygga arbetslag, distribuerad utveckling och om hur kaos leder till misslyckande.
- Livscykel där man återkommer till en del processfrågor och där även mitt eget föredrag om att utnyttja ett systems uppdrag (missions) som ett underlag för kravarbete är med.
- Programvaruunderhåll med underhållsaspekter av COTS, hur man hanterar system av system, utvärdering av provningsbarhet för krav mm.
- Web-hantering med B2Bi (Business to Business integration), hur man använder webben i försvaret, utvärdering av AWACS-arkitekturen, mm.
- Programvaruteknologi med kostnadsuppskattningar, riskhantering, kvalitet, mm.
- Datahantering med fusionsnätverk för dataintegration, realtidsaspekter mm.

- Ett spår, sponsrat av värden STSC, som handlar om utveckla programvara utan förhöjt blodtryck och förtida gråa hår.
- Ett spår om standarder och policy med bl. a. de senaste uppdateringarna av ISO/IEC 12207.
- Ett spår om "intjänat värde" som handlar om sådant som att forska i gamla system efter mönster och som diskuterar olika aspekter på "outsourcing".
- Provning med programvaruorienterad provning och utvärdering, provning av "svarta" och "vita" boxar, provningskvalitet och principer för provning.
- Metrik med hur man integrerar sin metrik, paneldiskussion mm.

Extra presentationer

Förutom huvudprogrammet finns de inledande och avslutande sessionerna där generaler och andra högt uppsatta personer talar om hur viktigt det är med programvaruutveckling och om hur försvarsdepartementet prioriterar detta arbete.

Sedan finns det en uppsättning leverantörspresentationer från utställarna där man kan få en närmare diskussion med presumtiva leverantörer eller samarbetspartners.

Vidare finns det lunchtalare att lyssna till när de talar om temat att styra kostnader, tidsplaner och kvalitet.

Tisdag morgon håller de militära med-sponsorerna en paneldiskussion om hur deras organisationer skall hantera programvara.

Dessutom finns möjlighet till diverse "sidomöten", vars innehåll inte direkt framgår av nuvarande underlag, men som säkert kommer att ingå i den utdelade informationen.

Miljön

Miljön vid STC är speciell och det första man märker dagligen är daglig "kontinental" frukost i amerikansk stil med kakor och giffjar. Frukosten ingår i konferensavgiften och sparar således ett tiotal dollar för hotellfrukost.

Konferensmiljön är imponerande med en jättelik utställningshall, många och stora föreläsningssalar, både i konferenscentret och i angränsande hotell. Strax intill konferenscentret finns flera bra hotell med en prisnivå på drygt 100 dollar/dag (plus några procent moms till Utah)

Intill konferenscentret finns också ett stort köpcenter med ett typiskt amerikanskt restaurangarrangemang med ett antal "stånd" från McDonald till grekiska, mexikanska och andra specialiteter samt ett stort område med bord och stolar, som är gemensamt för de olika restaurangerna.

Mitt emot köpcentret ligger sedan mormonernas världscenter med det genealogiska biblioteket där man kan släktforska, något som många passar på att göra (vid mitt förra besök fick jag hjälpa till att översätta sådant som "husförhörslängd" och "häradshövding" från svenska kyrkböcker). Man kan tycka olika om den mormonska läran, men parken är vacker och kören sjunger förträffligt en kväll varje vecka.

Salt Lake City är omgiven av stora fjäll

och skidsäsongen pågår fortfarande i början av maj. Den som vill hyra bil är uppe bland liftar och snö inom en timme.

En annan attraktion, för den som vill hyra bil., är Utahs nationalparker med helt otrolig natur och rustika men bekväma stugor till rimliga priser. Kolla dock in på Internet och förboka stuga eftersom det ofta är fullt.

Vad kostar det då? Konferensavgiften är \$ 685 - 770, men gratis om man är talare. Hotell, drygt \$ 100/dag och flyg tycks i år kosta ca 7000:- kr med ap-biljett. Traktamente är nu 537/dag.

Väl mött i Salt Lake City
Ingmar Ögren
0176-54580
iog@toolforsystems.com

Riksdagen godkänner "sexnationsavtalet"

Riksdagen godkände den 28 mars det s k sexnationsavtalet, d v s en överenskommelse mellan Frankrike, Italien, Spanien, Storbritannien, Tyskland och Sverige betitlad: Ett ramavtal om åtgärder för att underlätta omstrukturering och drift av den europeiska försvarsindustrin (Proposition 2000/01:49).

Avtalet har föregåtts av en gemensam avsiktsförklaring (LoI) den 6 juli 1998.

Sammanfattningsvis anförs i regeringens proposition "välkomnar regeringen den försvarsindustriella förändringsprocess som inletts i Europa. Försvarsindustrins internationalisering är en förutsättning för att vi skall kunna bevara industrikapacitet i Sverige och säkra vår försvarsmakts materielförsörjning med bibehållande av vår traditionella militära alliansfrihet. Internationellt samarbete ger också de bästa förutsättningarna för en ekonomiskt effektiv försvarsindustri, vilket inte minst är viktigt för att tillse att försvarets resurser räcker så långt som möjligt. Regeringen bedömer också att europeiskt industrisamarbete kommer att underlätta och påskynda avvecklingen av den exportdrivande överkapacitet som idag föreligger i europeisk försvarsindustri. Av såväl säkerhetspolitiska som försvars- och industripolitiska skäl är det följaktligen av stort värde att Sverige deltar i den omstrukturering som måste äga rum i Europa."

Var och en kan bilda sig en uppfattning om frågan genom att studera propositionen som finns att ladda ner från <http://forsvar.regeringen.se/propositionermm/propositioner/index.htm> och se redogörelsen över riksdagsbehandlingen, vilken finns på <http://www.riksdagen.se/bik/> (ange UU8 som sökord).

Ett systems uppgifter som underlag för kravarbete

För konferensen STC 2001 har jag fått ett föredrag antaget med rubriken "System missions as a basis for requirements' engineering". Vid STC är det så att den som vill hålla ett föredrag får lämna in ett sammandrag på en sida för bedömning och om det blir antaget är det sedan dags att börja arbeta på allvar. Den här gången tänkte jag inleda det arbetet med detta, något förlängda, sammandrag på svenska i förhoppning att det intresserar några av vännerna i Sesam. Eventuella synpunkter är naturligtvis mycket välkomna!

Det är lätt att acceptera att alla militära system skapas för att genomföra någon uppgift, eller flera uppgifter, och att systemet behöver en uppsättning förmågor för att genomföra uppgiften eller uppgifterna. Det är också uppenbart att alla, som är inblandade i en systemutveckling, måste ha en god förståelse för de uppgifter som skall genomföras. Vår tekniska tradition har gett oss många sätt att beskriva struktur och beteende för system. Vi har sådant som mekaniska ritningar, elektronicschemor, funktionella och objektorienterade diagram etc. Problemet är att inget av dessa beskrivnings-sätt klarar att beskriva ett systems uppgifter och förmågor, möjligen med undantag för "användningsfall". Användningsfallen begränsas dock till en informell beskrivning av gränssnittet mellan ett system och dess användare, vilket innebär att inte heller dessa egentligen klarar av att beskriva systemets uppgifter och förmågor.

"Uppgift" är ett substantiv, vilket innebär att man kan se en uppgift som ett objekt. En förmåga innebär att uppgiftsobjektet kan göra någonting. Följaktligen kan man se förmågor som metoder eller aktioner i relation till uppgiftsobjekt.

När man väl har etablerat uppgifterna som objekt, innebär det att man kan rita dem som objekt (eller om man så vill klasser) i UML (Unified Modeling Language) komponentdiagram, som då även kan kallas "objektgraf", förutsatt att man accepterar att diagrammet används bortom sitt ursprungliga ändamål, att modellera enbart programvara.

Komponentdiagrammet i UML syftar till att beskriva beroenden mellan objekt. Det innebär att när man har etablerat ett systems

uppgifter, som objekt på högsta nivå, så kan man använda diagrammet för att lägga in objekt av kategorier operatörsroll, maskinvara och programvara för att bygga upp de förmågor som krävs för att genomföra uppgiften eller uppgifterna. Detta kräver naturligtvis fortfarande att man accepterar att komponentdiagrammet används "bortom programvarumodellering".

Resultatet blir en beroendestruktur, med en uppsättning objekt, som är ordnade för att visa hur de bidrar till att genomföra systemets uppgifter. Man kan sedan fortsätta och lägga till information, som har att göra med kraven på systemet, som attribut till objekten:

- numrerade krav, som är fördelade till objekten för att visa var de bör provas i objektstrukturen
- provfall, fördelade till objekten och kopplade till de krav som är fördelade till samma objekt
- formella beteendespecifikationer för varje objekt att användas som ledning för implementation av objektet som t. ex. en operatörsroll, en programvarumodul eller en maskinvarumodul
- valideringsscenarior, relaterade till uppgiftsobjekten för att kunna säkerställa att det färdiga systemet verkligen kan genomföra de avsedda uppgifterna.

Resultatet blir en systemmodell, som innehåller olika typer av kravrelaterad information, med en potential att skapa en bättre förståelse och bättre verifiering och validering, jämfört med mera konventionella ansatser som kan vara begränsade till en uppsättning numrerade krav. Den ovan skisserade principen för systemarbete stöder att man arbetar parallellt med krav, konstruktion och verifiering, något som erfarenhetsmässigt är nödvändigt i varje icke-trivial systemutveckling.

Ett exempel

För att visa ett praktiskt exempel, börjar jag med att diskutera UML komponentdiagram. Diagrammet består av boxar och pilar där boxarna representerar objekt (klasser) och pilarna representerar beroenden mellan objekten. Problemet med detta kan vara svårigheter att se de olika nivåerna, när det

blir många objekt, och krav på en grafisk editor som inte alltid finns tillgänglig. Ett enkelt sätt att åtgärda problemet är att rita diagrammet som en indenterad lista med den enkla regeln att ett objekt beror direkt av underliggande objekt, som står ett steg indenterat. För att undvika oändliga slingor när ett objekt kommer igen flera gånger i strukturen får man komplettera med en regel som säger att när ett objekt dyker upp andra gången i listan så ersätts dess underliggande objekt med ett citationstecken ("-").

För att ta ett praktiskt modernt exempel har jag valt "integrerat simuleringsstöd till en stabsenhet". Stabsenheten förutsätts då (bland annat) ha uppgifter som:

- Planera taktiskt uppdrag (kräver simule-

ring för att utvärdera möjliga alternativ)

- Utbilda personal (stabsenhetens operatörsroller måste fyllas med utbildad och erfaren personal)
- Utveckla taktik (ett måste i en föränderlig värld).

Simuleringsstödet behöver kunna genomföra uppgifter som:

- Bereda simulering
- Köra simulering
- Utvärdera simulering.

Nedan syns hur man kan rita upp en objektstruktur för simuleringsstödet tillsammans med stabsenheten, enligt principer, som beskrivits ovan.

```

Stabsenhet
| MI_Planera_taktiskt_uppdrag
| | OP_Stabsofficer_planering
| | SW_Datorstöd_planering
| | | MI_Bereda_simulering
| | | | OP_Beredare
| | | | SW_Datorstöd_beredning
| | | | | SW_Databas_simuleringstillämpn
| | | | | HW_Datorsystem_simulering
| | | | | Kommunikationsnät.Kommunikationsterminal
| | MI_Köra_simulering
| | | OP_Simuleringsledare
| | | OP_Operatör_i_simulering
| | | SW_Simuleringstillämpning
| | | | HW_Datorsystem_simulering
| | | SW_Grundprogram_simulering
| | | | SW_Databas_simuleringstillämpn
| | | | HW_Datorsystem_simulering
| | | | Kommunikationsnät.Kommunikationsterminal
| | MI_Utvärdera_simulering
| | | OP_Utvärderare
| | | SW_Utvärderderingstillämpning
| | | | HW_Datorsystem_simulering
| | | SW_Grundprogram_simulering -"-
| | HW_Datorsystem_stab
| | | Kommunikationsnät.Kommunikationsterminal
MI_Utbilda_personal
| | OP_Utbildningsofficer
| | SW_Datorstöd_utbildning
| | | MI_Bereda_simulering -"-
| | | MI_Köra_simulering -"-
| | | MI_Utvärdera_simulering -"-
| | | HW_Datorsystem_stab
MI_Utveckla_taktik
| | OP_Utvecklingsledare
| | SW_Datorstöd_taktikutveckling
| | | MI_Bereda_simulering -"-
| | | MI_Köra_simulering -"-
| | | MI_Utvärdera_simulering -"-
| | | HW_Datorsystem_stab

```

Observera i trädet ovan hur uppgifterna är inlagda både på stabsnivå och på simulator nivå på raderna med prefixet "MI" (Mission). Observera vidare hur trädet visar hur varje uppgift stöds av operatörer (prefix "OP"), programvara (prefix SW) och maskinvara (prefix HW). Något som kan ifrågasättas är att flera av uppgifterna stöds parallellt av en operator och ett programvaruobjekt. Här kan man tycka att programvaruobjektet hellre borde vara underordnat operatören. Min erfarenhet är dock att gränssnitt mellan operatörer och programvara (kallas även användningsfall) framför allt handlar om utbyte av meddelanden: Operatören sänder meddelanden genom tangenttryckningar, musklick etc. och programvaran sänder meddelanden som skärmbilder, ljudsignaler etc. Detta motiverar att de två objekttyperna normalt läggs parallellt i modellen.

Från uppgiftsobjekt till krav på programvara

Trädet ovan kan ses som en första konstruktionsansats, som visar hur man kan sätta in systemets uppgifter i sitt sammanhang med operatörsroller, programvara och maskinvara. För att komma vidare är ett bra sätt att börja med att skapa en uppsättning scenarior i anslutning till varje uppgiftsobjekt. Granskning och genomgång av scenarion kan sedan ge ett underlag till en uppsättning krav, i anslutning till varje uppgift. Dessa krav måste uppfyllas av objekten som stöder uppgiften och kan följaktligen fördelas till dessa.

Längst till höger i trädet hittar man maskinvaruobjekt och externa system som kommunikationssystemet. Kraven på dessa ges sällan direkt från de krav man har hittat på uppgiftsnivå, men arbetet med trädet ger ändå ett väsentligt bidrag till kravbildningen på den här "lägsta nivån":

- arbete med de olika programvarumodulerna ger en uppfattning om vilka krav som gäller för maskinvaran och för de externa systemen.
- information om tillgängliga externa system och maskinvara ger underlag att formulera krav som gäller för programvarumodulerna.

Trädet ger också anledning till en hel del diskussioner om vad de olika objekten egentligen innebär och hur de bäst skall ordnas efter sina beroenden. Den här typen av diskussioner brukar ge underlag, dels till

uppdateringar av trädet, dels till ytterligare skrivna krav, som kan fördelas på objekten. Dessa diskussioner brukar också resultera i att man hittar problem, som behöver utredas, och som berör ett eller flera objekt.

Resultatet av arbetet med trädet, med utgångspunkt från systemets uppdrag blir alltså att man har genomgången och uppdaterat träd, som systemets intressenter är överens om och en uppsättning nedskrivna krav för varje objekt. Detta ger ett utmärkt underlag för vidare arbete med definition av beteende inom objekten och formulering av provfall för de olika objekten.

Några invändningar

Att vidga tillämpningen av UML komponentdiagram utöver modellering av programvara och dessutom ändra syntaxen från "boxar och pilar" till en indenterad lista kan ifrågasättas på flera sätt. Nu finns dock erfarenhet från flera projekt, som tyder på att det här arbetssättet är väl motiverat. Några invändningar kan vara:

- Vi arbetar med programvara, inte med operatörer och deras uppgifter.

Visst har vi som leverantörer ofta som vårt huvudansvar att leverera programvara. Dock kommer vi inte ifrån att programvaran måste samverka med operatörer och att den skall bidra till att genomföra uppgifter. Om vi då, före programutvecklingen, kan ha en god uppfattning om uppgifter och operatörsroller så ökar det chansen avsevärt att vi utvecklar rätt programvara.

- I UML är det inte komponentdiagrammet man skall använda för samverkan med operatörer utan användningsfallsdiagrammet.

Man kan ifrågasätta om användningsfallen egentligen passar in i det objektinriktade synsättet eftersom de egentligen handlar om gränssnittet mellan två olika kategorier av objekt. Genom att förbigå användningsfallen och arbeta genomgående med objekt, som sammanlänkas i ett beroendeträd med sina inbyggda gränssnitt kan man få en struktur med en genomgående och enhetlig syntax, baserad på komponentdiagrammet.

- Komponentdiagram skall ritas med pilar och boxar, inte med indenterade träd.

Problemet här är att verkliga system kan innehålla åtskilliga nivåer och att det blir svårt att ordning på nivåerna när man

ritar med "boxar och pilar". Vidare att korsvisa beroenden kan göra ett "renlärigt" komponentdiagram (eller klassdiagram) minst sagt rörigt. Det viktiga är att åskådliggöra systemstrukturen på ett begripligt sätt, inte att följa en "renlärig UML-syntax"!

Slutligen

Som har visats ovan ger "uppgiftsobjekten" en möjlighet att utgå från ett systems övergripande uppgifter och från dessa bygga en

struktur, som ger en "kravomgivning" till programvaran.

För att göra det hela så enkelt som möjligt är ett sätt att utgå from UML komponentdiagram och förenkla dessa till "indenterade träd".

Invändningar och diskussionsinlägg är välkomna för ett aktivare Rendezvous!

Ingmar Ögren
iog@romet.se

Referat av "Which Way SQA" i jan/febr 2001 numret av IEEE Software. Artikeln avser främst förhållanden i kommersiella företag och organisationer. Men kanske just därför av intresse för SESAM-kretsen i dessa dagar.

Vilken väg går kvalitetssäkringen?

I januari/februarinumret av IEEE Software ställer Emanuel E Baker den frågan. Baker är President of Software Engineering Consultants i Los Angeles.

Han undrar om kvalitetssäkringen (Software Quality Assurance, SQA) blivit passé i och med att företagen tillämpar RAD (Rapid Applications Development), moderna verktyg och spiralmodeller för utveckling.

Innan han försöker ge ett svar, tar Baker upp den gamla frågan om SQA är en disciplin eller organisation. Hos organisationer i den kommersiella världen består SQA av testning, i första hand systemtest. Ofta är för litet tid tillgänglig för att testa ordentligt, beroende på dålig planering eller ledning. Kravdokumentationen kanske inte är tillgänglig, så att testprogrammets förmåga att upptäcka fel är suspekt. Att låta testning utgöra SQA är som *"locking the barn door after the horse has escaped"*; det är knappast att kvalitetssäkra, inte heller är det disciplinen SQA.

Folk i företag som antagit CMM, ser ofta SQA som "processpolis". I den rollen avgör SQA huruvida utvecklare och projektledning uppfyller processpolicyn, standarder och förfaranden. Man kollar arbetsresultat mot mallar som skall definiera deras innehåll och form. I organisationer som har en separat

SQA funktion, kanske de som gör detta, har liten programvarubakgrund och inte så tydligt kan se processavvikelse, som mer kunrig SQA-personal skulle göra.

Baker jämför med hur det gick till i den produktsäkerhetsavdelning han basade för förr i tiden. Då definierade man organisationens process, kollade efterlevnaden, utvärderade arbetsresultaten m a p innehållets kvalitet och överensstämmelse med kraven, ledde granskningar och utförde en del tester. Och så gick det till också i många andra företag, mycket annorlunda i förhållande till det ovan beskrivna.

Idag ser man ett antal förfaranden som varierar mycket; knappast passar de in i beskrivningen av en disciplin. SQA är inte alltid en organisation. Inte heller när det är en organisation är den obereonde, enligt Baker en nödvändighet för att säkra objektiva utvärderingar.

Vilken inverkan har den teknologiska utvecklingen på hur man utövar SQA?

Baker citerar en artikel av Walker Royce (son till vattenfallsmetodens upphovsman och själv framstående datorsystemarkitekt, tidigare vid TRW, numera hos Rational). Royce be-

skriver där hur modern programvaruutveckling först producerar arkitekturen, sedan användbara inkrement av funktionalitet och småningom hela systemet. Brister i krav och konstruktion upptäcks och åtgärdas tidigt i livscykeln och man undviker "big-bang" integrationen vid projektets slut. Kvalitetskontrollen förbättras genom att systemegenskaper som följer av arkitekturen (prestanda, fel-tolerans etc) kan identifieras tidigare och åtgärdas utan att kostnader och leveranstider behöver äventyras.

Enligt Royce tenderar den gamla principen med separat kvalitetssäkring att resultera i projekt som isolerar "kvalitetspoliserna". Det är bättre att arbeta in kvalitetsvärderingar (QA) i varje aktivitet genom "checks and balances" som finns i organisations-team som är fokuserade på arkitektur, komponenter och användbarhet. Ett annat fel i det gamla sättet att driva QA är enligt Royce att slaviskt insistera på spårbarhet från krav till konstruktion, då detta ofta blir kontraproduktivt genom att det tvingar konstruktionen att bli strukturerad på samma sätt som kraven. Bra komponentbaserade arkitekturer har "kaotisk" spårbarhet till kraven. Strikt spårbarhet krav-lösning var kanske produktivt när man producerade 100% skraddarsydd programvara; dessa dagar är förbi.

Den roll för QA som träder fram ur Royce syn är att den bör ske genom utvärderingar i utvecklingsprocessen, inte genom separat SQA och "polisinsatser".

Hur går det till e-commerce världen

Baker fortsätter med att jämföra Royce åsikter med vad som tillämpas bland e-commerceutvecklarna (och även överhuvudtaget av "shrink-wrap" leverantörerna), där det ju talas om att göra saker "at Internet speed". Där gäller det ofta att få ut produkterna så fort som möjligt för att komma före konkurrenterna. Processer är av ringa intresse för många e-commerce-företag. Det är få sådana företag som har visat intresse för att tillämpa kvalitetsmodeller som CMM. Som resultat får kunderna ofta bristfällig programvara.

Under lång tid har rättsligt ansvar inte varit en stor fråga; kunden tolererade helt enkelt bristfällig programvara. Men med Internet och e-commerce har ansvarsfrågan blivit mer akut. Om ett program inte ger tillräckligt skydd t ex för kreditkorts-transaktioner, reser det naturligtvis allvar-

liga frågor om rättsligt ansvar.

Baker citerar olika studier som säger att 30-40 % av programvaruprojekten misslyckas, att upp till hälften av misslyckandena beror på otillräcklig kravdefinition och att upp till 40 % av felen under ett projekts livstid, kan hänföras till kravproblem. Trots detta utvecklar e-commerce-företagen tillämpningar "at Internet speed" och otivelaktigt har de inte mycket tid för ordentlig kravdefinition och testning. Baker anser att dessa företag löper stora ansvarsrisker och att de har mycket att lära om de kvalitetsprinciper som diskuterats ovan.

CMM och ISO 9000

Baker diskuterar sedan kvalitetsmodeller som CMM och ISO 9000 och konstaterar att de i sina nuvarande utformningar föreskriver QA övervakning med omfattande granskningar, revisioner och protokoll.

CMM, som är den i USA mest använda kvalitetsmetoden av de två nämnda, fordrar att man sätter upp en SQA-funktion för att nå Level 2 i huvudsak av typen "processpolis". Detta kan innebära att slaviskt följa processen vare sig det är meningsfullt eller ej. Det kan också leda till personliga motsättningar; "polimentalitet" kan underminera lagandan. CMM ger heller ingen utvecklingsväg för SQA upp genom CMM-nivåerna. Då är CMMI bättre i det avseendet för den definierar en evolutionär roll för SQA (där kallad Product and Process QA). Å andra sidan verkar CMMIs ramverk avse att denna evolution endast gäller mätningar och förbättringar av SQA-processen, inte att den kunde utföra många av de andra mätningar och kvantitativa processutvärderingar som på Level 4 och 5 specificeras för alla Key Process Areas, vilket skulle kunna göra att man undviker för många avgränsade roller i organisationen.

SQA i framtiden

Baker försöker sedan att finna en meningsfull roll för SQA i framtiden. Det är helt klart önskvärt att innefatta granskningar och evalueringar i processen, men det är troligen svårt att göra detta på ett konsistent sätt på CMMs lägre nivåer, därför att det finns för litet av processkultur i de flesta organisationer som befinner sig på de nivåerna.

Baker föreslår i stället följande. För att nå Level 2, som är den nivå där projektledningsdisciplin blir obligatorisk, krävs en större förändring av organisationens in-

ställning till att acceptera processdisciplin. Eftersom organisationer tenderar att ignorera process så mycket som möjligt och fokuserar på att få ut produkten på marknaden snabbt, så kan SQA-rollen där behöva lägga tyngdpunkten på "polisinsatser och fredsskapande" för att inympa en processorienterad kultur i organisationen. Detta är förmodligen mindre nödvändigt när man går från Level 2 till Level 3, där en definierad process skall finnas.

I Level 2 organisationer, som ibland återfaller i primitivare beteenden, kan det vara nödvändigt för SQA att ägna mycket uppmärksamhet åt processuppfyllande. När man rör sig mot Level 3, har organisationen förhoppningsvis sett att processdisciplinen har gjort livet lättare och SQA skulle kunna att ägna sig mindre åt "polisinsatser" o dyl. När organisationen nått Level 3 har man upprättat en processdatabas och SQA kan ägna sig mer åt att samla in och analysera processmetrik.

När organisationen nått upp till Level 4 och 5, bör behovet för SQA att agera "processpolis" minska än mer. På Level 4 skall processdisciplin vara en "way of life" och verifiering av uppfyllelse än mindre nödvändig. SQA kan fokusera på processmätning, att vidta korrekta åtgärder när så behövs och att övervaka att man uppnår de kvantitativa målen.

På Level 5 kan SQA roll vara som metrikinsamlare och att göra en första analys av trender som kan fordra processförändringar samt föreslå sådana. Sedan skulle de som

är experter på specifika processer och teknologier, ha ansvaret för att bedöma att ändringarna är ändamålsenliga.

En sådan roll för SQA behöver inte vara helt fristående. Utvärderare kunde på deltid ha SQA-funktioner, utvecklare som gör metrikanalyser kunde på deltid verka i SQA-roller. Dock bör de som definerar och organiserar programmet och övervakar dess resultat vara oberoende för att kunna se till att det fungerar bra. Dena grupp anser Baker, behöver inte vara stor, men den beskrivna rollen skulle kunna vara kostnadseffektiv, icke störande och värdeadderande.

Dock menar Baker, skulle den beskrivna rollen för SQA ändå inte ge en övergripande QA-bild. Att genomföra "peer reviews", inspektioner, revisioner o s v, tenderar att ge en mikroskopisk uppfattning om kvaliteten inom projektet eller produkten. Det behövs också en makroskopisk vy av kvalitet, där man tar ett steg tillbaka och ser hela bilden för att avgöra om alla delarna passar ihop, t ex om kundens krav verkligen tillfredsställs, eller om det tillämpade SQA-förfarandet skapar andra slags kvalitetsproblem. Det behövs fortfarande mer globala granskningar på projektnivå vid viktigare milstolpar i projektet.

Bakers slutkläm är att det är önskvärt att "embedda" kvalitetsvärderingar i utvecklingsprocessen, men att motiveringen för detta inte enbart är att få ner "overhead", samt att balanserade vyer av makro- och mikroskopisk kvalitet beaktas när man strukturerar SQA-program.

Utredningarna om försvarsteknikens framtid nu klara

Vid SESAM-seminariet i oktober orienterade Lennart Lundh från Försvarsdepartementet om arbetsläget i de två stora teknik- och industriorienterade offentliga utredningar som då pågick, nämligen Materieförsörjningsutredningen resp FoU för totalförsvaret. Bägge utredningarna levererade sina slutbetänkanden den 28 februari. De är nu ute på remiss till den 15 maj.

Vi skall inte här försöka oss på någon recension, utan hänvisar intresserade dels till Lennart Lundhs föredrag från seminariet vilket finns på SESAM hemsida, dels till Regeringens hemsida [<http://forsvar.regeringen.se/propositionermm/sou/index.htm>], varifrån betänkandena (SOU 2001:21 resp 22) går att ladda ner.

Systems Engineering standarden 15288

närmar sig målnöret

En gemensam ISO & IEC arbetsgrupp (ISO/IEC JTC 1/SC 7/WG 7) arbetar för fullt med att färdigställa vad som skall bli den kommande Systems Engineering standarden ISO/IEC 15288, eller Systems Life Cycle Processes som den kanske kommer att kallas, med den breddade inriktning den har fått under arbetets gång. Committee Draft 3 (Final) är (har varit) ute på remiss till den 15 april tillsammans med Working Draft 2 av den tillhörande handboken Guide for ISO/IEC 15288.

Både ITS, det svenska IT-standardiseringsorganet, som är det officiella svenska organet för deltagandet i standardiseringsarbetet, och INCOSEs svenska chapter har ordnat genomgångar av de aktuella förslagen för industri- och myndighetsrepresentanter och samlat in deras synpunkter på dem under de senaste månaderna. Svensk representant i WG 7 är Bud Lawson, som också är gruppens arkitekt för standardförslaget. Bud är den som sammanställer de svenska synpunkterna f v b till WG 7.

När Committee Draft 3 är färdigbehandlad i WG7 går den vidare till de formella remissrundorna och omröstning inom ISO och IEC. Någon gång under nästa år kan vi vänta oss att den är fastställd.

Förslaget som det numera ser ut, skiljer sig en hel del från den amerikanska MIL-STD 499B – Systems Engineering, som kan sägas vara utgångspunkten för det aktuella standardiseringsarbetet. 499B blev klar just som DoD med sin Acquisition Reform 1994 bestämde att man så långt möjligt skulle gå över till civila standarder och fastställdes därför aldrig. I stället engagerades DoD-representanter i civila standardiseringsorgan för att hjälpa till att få fram standarder inom dessa som även fyllde angelägna DoD-behov.

Det första område som man satsade på från DoD sida var programvaruutveckling. Detta ledde till den för ett par år sedan fastställda ISO/IEC 12207 – Software Life Cycle Processes, vilken (delvis) kan sägas vara en efterföljare till MIL-STD 498, välkänd i programvarukretsar. Denna följdes sedan av arbetet med 15288. Vid ett SESAM-seminarium 1995

eller 96 berättade DoD dåvarande sammanhållande för deltagandet i ISO/IEC-verksamheten på dessa två områden, Raghu Singh från US Navy, om planerna för den verksamhet som nu närmar sig sitt slut.

15288 har i sin uppläggning tagit stora intryck av 12207, men också utformats så att den täcker in mycket mer än det traditionella Systems Engineering området, därför också den namnändring eller komplettering som benämningen Systems Life Cycle Processes indikerar.

15288 är avsedd att kunna användas i flera sammanhang:

- Av en organisation, för att upprätta en miljö av önskvärda processer som stöds av en infrastruktur med metoder, procedurer, tekniker, verktyg och utbildad personal. Denna miljö kan användas av organisationen för att styra och genomföra projekt under deras livscykel.
- Av ett projekt inom en organisation, för att hjälpa till att välja, strukturera, och tillämpa delar i den etablerade miljön för att leverera produkter och tjänster.
- Av en anskaffare och en leverantör, för att stödja att åstadkomma ett avtal beträffande processer och aktiviteter. Via avtalet väljs ut, överenskomms och utförs dessa aktiviteter

I samtliga moder kan utvärdering (assessment) ske av uppfyllandet av den internationella standarden och parternas prestationer i förhållande till vad som avtalats.

Mycket intressant om och kring 15288 finns på <http://www.15288.com/>

FMV avser basera sina interna processer på 15288

Av särskilt intresse för SESAM-kretsen torde vara att FMV efter försök med andra förebilder, avser basera utformningen av ramverket för sina interna processer på 15288. Om inte av andra skäl, så kommer detta helt säkert att leda till att även industrin i sina kontakter med FMV kommer att behöva bekanta sig med 15288. Det ökande internationella sam-

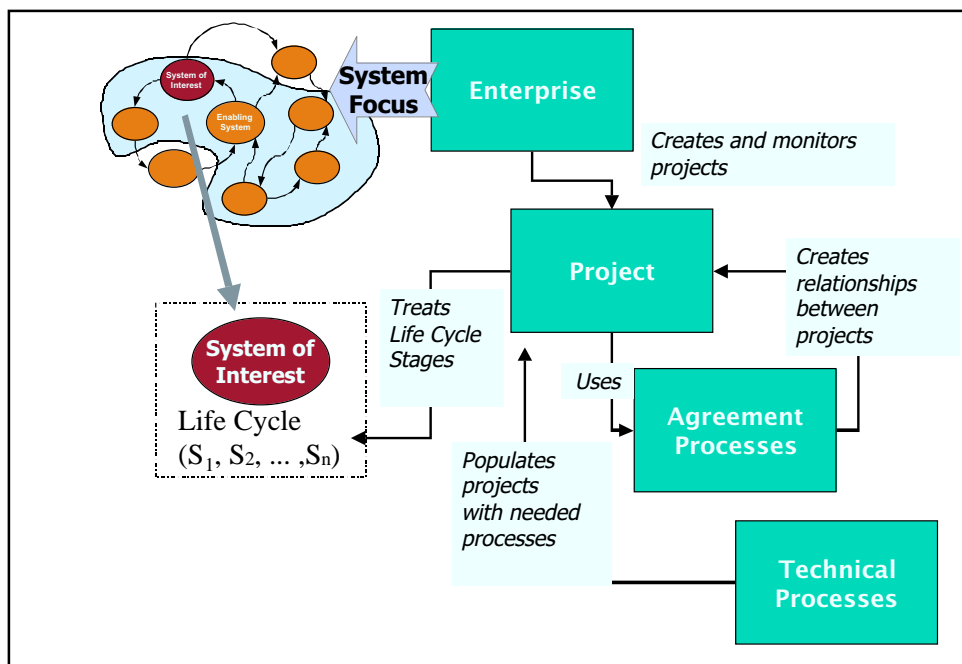
arbetet på försvarsmaterielområdet är ett annat givet tillämpningsområde för en internationell standard som ISO/IEC 15288.

Bud Lawson höll häromveckan en genomgång för FMV-personal om 15288 och dess möjliga tillämpning inom FMV.

Vi återger nedan genom tillmötesgående av Bud ett par av hans OH-bilder vid detta tillfälle, som visar tillämpligheten av standarden i FMV-sammanhang.

WHAT IS ISO/IEC 15288?

Process Building Blocks For Managing The Life Cycle of Systems



How can 15288 help FMV? And others?

- Provides a "framework" for the "life cycle" of the "FMV process system".
- Can be "tailored" to accommodate special organizational and project requirements.
- Provides a common language for agreements between FM - FMV and FMV- Suppliers.
- A vital complement to ISO 9001:2000.
- The catalyst for promoting international trade in system products and services.

There are no advantages of a "Process Approach"

UNLESS

- Concepts of the process approach are understood by all parties that are affected.
- Processes are "executed" by competent people in their respective roles.
- Organizational structure supports process requirements and promotes true "teamwork" leading to collective added value

OBS

För den som vill ha en koncentrerad och initierad genomgång av standardförslaget ordnar Bud Lawson med Tom Strandberg på Syntell AB, ett heldagsseminarium i Stockholm den 29 maj. Se även notis på annan plats i detta nummer.

Seminarium ISO/IEC 15288 System Life Cycle Processes

Standard för hantering av komplexa system - Innehåll och tillämpning

Seminarieret anordnas i Stockholm tisdag 29 maj, 2001, kl 09.00-17.

Målgruppen inkluderar samtliga som arbetar med komplexa system och verksamheter under deras livscyklar. Tex projektledare, processutvecklare, produktansvarig, systemutvecklare, tjänsteansvarig, upphandlare, kvalitetsansvariga, mfl.

Målet är att deltagarna ska erhålla:

- en förståelse för standardens innehåll
- en inblick i tillämpningar av standarden
- en beskrivning av standardens koppling till ISO9001:2000 och angränsande standarder

Programinnehåll

Inledning

- Behovet av och målsättning för standarden
- Standardiseringsarbetet: historik, status och framtid

Standardens innehåll

- Standardens arkitektur (uppbyggnad och relationer)
- Systembegreppet (helhetssyn)
- Processerna (byggblock för organisation, projekt, avtal och teknik)
- Livscykel (från idé till avveckling)

Standardens tillämpning

- Processutveckling (ramverk för organisationens processer)
- Projektledning (ramverk för projekt)
- Kontraktshantering (upphandling och leverans)

Praktikfall

- Introduktion av ISO/IEC 15288 i en stor organisation
- Utnyttjande av ISO/IEC15288 för certifiering enligt 9001:2000
- Ramverk vid utbildning av projektledare och systemingenjörer

Diskussion

Sammanfattning

Presentatörer:

Harold (Bud) Lawson, Lawson Konsult AB, ACM och IEEE Fellow, Head of Swedish Delegation to ISO/IEC JTC1 SC7 WG7, som utvecklar ISO/IEC15288 (System Life Cycle Processes), och vald arkitekt av standarden.

Tom Strandberg, Technical Director, Syntell AB, Regional Director INCOSE (International Council on Systems Engineering) och medlem INCOSE Standards Technical Committee.

Pris: 3.450 kronor inkl. litteratur och lunch & kaffe, men exkl. moms

Frågor

Om innehållet, kontakta Tom Strandberg, Syntell 08-660 02 80 eller e-post strandberg@syntell.se

Angående de praktiska arrangemangen kontakta Iréne Öhrn Franzén, samma telefonnummer.

Anmälan senast den 22 maj.

SNART och Högskolan i Halmstad arrangerar den svenska konferensen

Real-Time in Sweden 2001 RTiS2001

den 21-22/8 2001 i Halmstad

Konferensen Real-Time in Sweden 2001 har som målsättning att sammanföra industri, och akademi och presentera en översikt över forsknings- och utvecklingsaktiviteter i vilka realtidsegenskaper spelar en viktig roll.

Konferensen som arrangeras vartannat år av SNART, är den sjätte i ordningen sedan 1991. SNART (Svenska Nationella Realtidsföreningen) är en ideell förening med medlemmar från högskolor och industri med ett gemensamt intresse för realtidssystem. Mer information om SNART och uppdaterad information om konferensen finns på <http://www.docs.uu.se/snart/>.

Årets konferens kommer att organiseras på ett nytt sätt med målet att ge en helhetsbild av ämnet snarare än att enbart presentera specifika forskningsprojekt. Programmet är uppbyggt kring sex generella ämnen med mycket utrymme för personliga och/eller mer specialiserade diskussioner, möten och seminarier. Målet är att tillgodose systemutvecklare, forskare och andra med intresse av realtidssystem, med information om viktiga problemställningar och lösningar; fortgående forskning och utveckling i Sverige; samt trender och en översikt över vem som gör vad.

Realtidsaspekter inom följande ämnen kommer att adresseras:

Operativsystem och programvaruinfrastruktur (middleware); applikationer - fordonstillämpningar, programmering, kommunikation, systemdesign och säkerhetskritiska system. Varje ämne kommer att presenteras ur både en industriell och en akademisk synvinkel av erfarna och välrenommerade forskare och utvecklare. Korta introduktioner (tutorials) till viktiga tekniker, verktyg och metoder kommer att arrangeras, liksom utställningar av produkter, experiment, projekt m.m.

Konferensbidrag

Arrangörerna inbjuder bidrag till Real-Time in Sweden 2001. Den som vill bidra med en presentation på konferensen, skall skicka in ett bidragsförslag till RTiS2001-contribution@ide.hh.se senast den 20/4 2001. Alla typer av konferensbidrag välkomnas , men speciellt följande:

- o Examensarbeten avslutade under tiden 2000-06-01—2001-04-20. Vänligen skicka in hela rapporter. Ett pris för bästa exjobb 2001 kommer att utses av programkommitten. Mer information finns på <http://www.docs.uu.se/snart/>
- o Presentationer av forsknings- och industrigrupper eller avdelningar i seminarie- eller posterform.
- o Demonstrationer, produkter och experiment för utställning.
- o Företags-, universitets- och institutspresentationer i utställnings- eller posterform.
- o Individuella projekt på seminarie- eller posterform.
- o Mini-introduktioner (tutorials) på seminarieform.

Hemsidor

Tyvärr har det av olika skäl varit litet dåligt med uppdateringen av vår egen hemsida senaste tiden, men bättring är på gång. När det är något som inte vi kan stå till tjänst med där, finns det några andra hemsidor som är mycket matnyttiga (se t ex länkarna från SESAMs hemsida). Speciellt vill vi fästa er uppmärksamhet på följande tre:

Ada-Belgien har en mycket innehållsrik hemsida som ger svar på det mesta man kan behöva veta om vad som händer i Ada-världen och dess omgivning
<http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/>

Ada Language UK har en helt ny utformning av sin hemsida som bör ses
<http://www.adauk.org.uk/>

DoD/AJPOs gamla hemsida Ada Information Clearinghouse drivs och utvecklas vidare av **Ada Resource Association**, som bl a har den träffande devisen **Choose Ada "The Most Trusted Name in Software"**™.
<http://www.adaic.com/ara/>

Du besöker väl vår websajt?
<http://sesam.tranet.fmv.se>

Kalendern

26 april	Rådsmöte
27 april	AiS Vårmöte 2001 i Stockholm. Om Corba och Ada. Några stora företag kommer att presentera sina lösningar. Exempelprojekt från industri och akademi granskas. Program och anmälningssblankett finns på AiS hemsida http://www.ada-i-sverige.se/spring01.pdf
5 juni	VU
11-13 juni	ICECCS 2001, 7th IEEE International Conference on Engineering of Complex Computer Systems, Högskolan i Skövde, Skövde http://www.elet.polimi.it/iceccs2001
30 juni-4 juli	ISCA '01: 28th International Symposium on Computer Architecture, Göteborg http://www.ee.princeton.edu/~isca2001
23 aug	VU
11 sept	Ag Metodik sem "Drömverktyget"
4 okt	VU
24-25 okt	Höstsem o Rådsmöte
6 dec	VU

SESAM-Sekretariatet:	AerotechTelub AB c/o Kåsjös Kontor Ytterspåret 14 187 54 TÄBY	Telefon: 08-510 51866 Telefax: 08-510 51932 GSM: 070-716 9702 E-post: alkas@tranet.fmv.se
-----------------------------	--	---