

Mikroprojekt 2004

Metodik

Informationshantering vid systemutveckling styrd av CM

Författare	Håkan Edler Torbjörn Jungeby Tore Qvist
Dokument Id	CM rapport v1.0.doc
Version	1.0
Datum	15 mars 2005
Tillgänglighet	SESAM
Status	Slutlig
Granskad av	
Datum	

Dokumentets historik

Version	Författare	Datum	Ändringsinformation
0.0	HE	18 aug 2004	Disposition, stolpar
0.1	HE	30 aug 2004	Smärre ändringar efter mötet hos Bofors
0.2	HE	15 sep 2004	Ändrad disposition
0.3	HE, TJ	30 sep 2004	Bidrag från TJ
0.31	HE	19 okt 2004	Kap 5.2 och 7
0.4	TJ	31 jan 2005	Kap 9
1.0	HE	15 mar 2004	Kap 11, slutredigering

Innehållsförteckning

1. Sammanfattning	4
2. Bibliografi	4
3. Syfte och mål	5
4. Bakgrund	5
5. Tillgänglig teknik	6
5.1 Aktuella standarder	6
5.2 Utvecklingsmetoder inom försvarsindustrin	8
5.3 Projektstyrningsmodeller inom försvarsindustrin	9
5.4 Aktuella CM-verktyg	10
6. Begrepp i systemutvecklingsprocessen	11
6.1 Terminologi	11
6.2 Objekttyper	11
7. Processen konfigurationsstyrning	12
7.1 Syfte	12
7.2 Resultat	12
7.3 Aktiviteter	12
8. En generell modell av delprocesser	13
8.1 Process kontra information	13
9. En fallstudie	14
10. Fortsatt arbete	15

1. Sammanfattning

Mikroprojektet har ur ett antal synvinklar studerat CM för utvecklingsdokument. Ett av projektets första konstateranden var, att aktuella standarder för systemutveckling har svag koppling till CM. Anledningen är att standarder koncentrerar sig på processer medan CM hanterar information. I aktuella standarder finns beskrivningar av processer för konfigurationsstyrning och de skall givetvis användas i det fortsatta arbetet.

Ett angreppssätt projektet prövade var, att göra en informationsanalys av ett utvecklingsprojekt. Det är motiverat, då systemutveckling är ett stort informationshanteringssystem. Arbetet gav till en början en god grund att stå på med begrepp och terminologi preliminärt klargjorda. Det kan vara värdefullt för fortsatta resonemang om den roll CM kan ha i en stor utvecklingsprocess. I detta läge skulle dock den fortsatta analysen ge alltför mycket arbete.

Ett väsentligt bidrag från mikroprojektet är den fallstudie, som gjorts hos Saab Bofors Dynamics. Den visar hur alla dokument, som idag hanteras i ett stort utvecklingsarbete, skulle kunna administreras i ett CM-system. Det skulle ge den ordning och reda och den enda ingång till all information i ett utvecklingsprojekt, som vi efterlyst.

2. Bibliografi

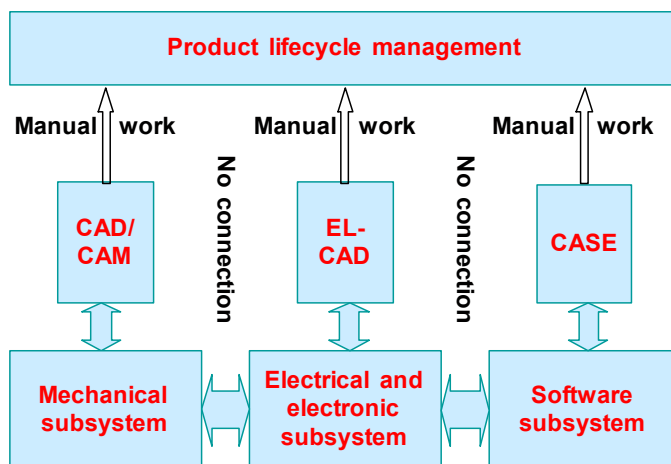
- [Props] Props, Ericssons modell för projektstyrning
- [PSM] PSM, processstyrningsmetodik, Saabs anpassning av Props
- [12207] ISO/IEC 12207, "Information technology – software life cycle processes", ISO/IEC, Geneve, 1998
- [www12207] www.12207.com, senast besökt 10 oktober 2004.
- [IEEE/EIA.0] IEEE/EIA 12207.0-1996, "Industry Implementation of International Standard ISO/IEC 12207:1995, Standard for Information Technology, Software life cycle processes", IEEE, 1996
- [IEEE/EIA.1] IEEE/EIA 12207.1-1997, "Industry Implementation of International Standard ISO/IEC 12207:1995, Standard for Information Technology, Life cycle data", IEEE, 1997
- [15288] ISO/IEC 15288, "Systems engineering - system life cycle processes", ISO/IEC, Geneve, 2002
- [15504] ISO/IEC 15504, "Information Technology – Software Process Assessment", ISO/IEC, Geneve, 1998.
- [AM] <http://agilemanifesto.org/>, senast besökt 19 oktober 2004.
- [Boz04] Bozheva, T., Gallo, E., Vergara, M. och Regil, A., "Agile patterns", ESI rapport från ett EU-projekt, under utarbetande
- [Str03] Ström, M., Andersson, D., Hörnstein, J., Lindqvist, M., Sihlbom, A., "Sammanhållen informationsmodell för inbyggda system, IVF rapport 03002, 3 dec 2003.
- [Wan00] Wang, Yingxu and King, Graham, "Software engineering processes: principles and applications", CRC Press LLC, 2000, ISBN 0-8493-2366-5
- [Ögr98] Ögren, I. "O4S™ Objects For Systems, handbook part 2, activities and documents", Romet AB, mars 1998.
- [Hum00] Humphrey, W. S., "Introduction to the team software process", Addison-Weslwey, Reading, 2000, ISBN 0-201-39815-X.

3. Syfte och mål

Syftet med arbetsgruppens aktuella arbete är, att möjliggöra ett samordnat informationsutbyte mellan personer, processer och verktyg under utveckling av ett inbyggt system oberoende av utvecklingsfas och teknikområde.

Målet med det aktuella mikroprojektet är, att undersöka hur CM kan användas som styrinstrument i utvecklingsarbete för inbyggda system.

4. Bakgrund



Figur 1. Bristen på kommunikation är ett av de största problemen vid utveckling av inbyggda system.

Ett stort problem vid utvecklingen av inbyggda system är att arbetet inom ett teknikområde till stor del arbetar separerat från andra teknikområden. De verktyg man arbetar med kan i allmänhet inte byta data med varandra. Inte för att standarder för datarepresentation och datalagring saknas, tvärtom, flera ansatser har gjort under de senaste decennierna, men de har inte fått genomslag på marknaden. Inom mekaniskt konstruktionsarbete har man dock kommit ett stycke på väg, flera CAD-verktyg använder standardiserade filformat. Detsamma gäller troligen elektronik- och el-konstruktion, men vid utveckling av programvara är problemet accentuerat. De kommersiella verktyg som används kan inte kommunicera med verktyg från andra leverantörer. Undantag finns, men dem får man leta efter inom Open Source-världen.

Motsvarande problem gäller lagring av produktdata i en produktdatabas antingen det är ett PDM-system, ett CM-verktyg eller ett versionshanteringssystem. Data som skapas under utvecklingen av ett inbyggt system måste manuellt föras in i produktdatabasen.

Visionen är att data om en produkt kan lagras på sådant sätt, att utvecklare, förvaltare och användare skall kunna få fram de data man för tillfället behöver på ett enkelt sätt.

Mikroprojektet började med en idé om att använda CM som styrmedel under en produkts hela livscykel. Idén formulerades bäst på engelska: "Development information enforced by CM" och gruppen hade inledningsvis svårt att hitta en svensk titel, som så pregnant uttryckte vad vi ville åstadkomma. Senare visade det sig att titeln inte var så entydig, som vi i först tyckte. Tolkningarna är två: Dels kan CM ge en god styrning av informationshanteringen i ett projekt, dels kan CM påtvinga ett projekt onödigt stor och rigid informationshantering. I det fortsatta arbetet har gruppen valt den förra tolkningen. Utveckling av ett inbyggt system är ett stort informationssystem i sig och CM av informationen kan bidra till att ge ordning och reda.

Några funderingar från en av gruppens medlemmar:

Titeln antyder för mig att CM i någon mening tvingar (driver) fram informationsmängder i en utvecklingsprocess. Snarare är det väl önskan om "ordning och reda" som drivit fram ett verktyg som underlättar lagring av stora informationsmängder, jmf bibliotek. Många av dagens utvecklingsprojekt tenderar att innehålla för en mänsklig hjärna oöverskådliga mängder information som vi människor då behöver hjälp med att lagra, strukturera, koppla till varandra, kunna spåra, göra möjliga att publicera för många m m. Kravet på många produkters kvalitet (sekretess, personsäkerhet, funktion etc.) säkras idag m h a av olika utvecklingsmetoder, standarder etc. Dessa standarder spänner upp en volym av olika grundelement av karaktären "bra att ha information" och tillvägagångssätt, beslutssekvenser, informationsspridning etc. för att med större sannolikhet kvalitetssäkra slutprodukten. Även kravet på att kunna byta personal eller föra över projekt mellan olika arbetsgrupper eller företag förutsätter också ett stort mått av systematisk "ordning och reda" som ett CM bidrar starkt till.

Beslutsregler

Vad avser beslutsregler kopplat till processer och roller är idag ett CM till stor nytta, eftersom roller kopplat till t ex användare och användargrupper är lätt att implementera. Med dessa roller kan en organisation till stor del få en del oönskade arbetsätt som t ex "Quick and dirty".

"Vart vill vi"

I ett ideellt fall finns för mig önskan om ett gemensamt CM-verktyg som för varje projekt innehåller en och endast en ingång och via denna ingång ger tillgång till all information och i vissa fall speciell utpekning för specifika roller. De projekthemsidor som ibland tas fram idag fyller så vitt jag ser till viss del denna roll. Som komplement till denna diskussion kan nämnas att verktyget iNotion från Ilogics är ett verktyg som kan leda mänskligheten ytterligare ett steg på struktureringens (ordning och reda) underbara väg, genom att tillhandahålla en projekt-, produktgång som i sin tur kan sättas upp att peka på alla relevanta verktyg och informationsmängder.

"Svaret"

Alltså vad gäller frågan huruvida CM tvingar fram utvecklingsinformation eller vice versa, vill vi påstå att CM är ett verktyg som tagits fram för att kvalitetssäkra framtagningen av system med stor komplexitet och stora informationsmängder samt krav på kvalitet och funktion och så bör rubriken tolkas.

5. Tillgänglig teknik

5.1 Aktuella standarder

Citerat från [www12207] och referat ur [12207]:

ISO/IEC 12207 publicerades 1995 och är den första standard, som beskriver en samordnad uppsättning processer, aktiviteter och uppgifter för hantering av programvara under hela dess livscykel. Programvaran kan vara ett självständigt system eller en del av ett större system eller kan vara mer allmän mjukvara som tjänster runt ett programvarusystem. Standarden ger en allmän arkitektur för processer för anskaffning, leverans, utveckling och drift av programvara. Den specificerar också de nödvändiga processerna, aktiviteterna och uppgifterna för understöd, styrning och förbättring.

Standarden är ett allmänt ramverk för utveckling och hantering programvara under hela dess livscykel. Den definierar ett antal processer och ger en väldefinierad terminologi. Standarden är generell och man förutsätter, att den anpassas till organisation och projekt när den skall tillämpas.

Begränsningar i standarden:

- Den beskriver arkitekturen hos programvaras livscykelprocesser och specificerar inte hur aktiviteter och uppgifter skall realiserats och utföras.

- Den föreskriver inte namn, format eller innehåll i de dokument som produceras i samband med utveckling eller användning av programvara.
- Den föreskriver inte någon livscykelmodell eller utvecklingsmetod.

Standarden har kompletterats efter krav som dykt upp under utvecklingen av ICO/IEC 15504 (se nedan) och är nu grunden för en referensmodell för utvecklingsprocesser. En referensmodell definierar alla processer, som hanterar programvara under hela dess livscykel med definition av deras syfte och förväntade resultat. Den definierar också sambanden mellan alla processer, sekvensiellt och hierarkiskt.

ISO/IEC 12207 gäller enbart utveckling och hantering av programvara. Den beskriver 22 processer, 95 aktiviteter, 325 uppgifter (tasks) och 254 processresultat (outcomes). Standarden beskriver mer än 400 objekttyper som policies, procedurer, planer, poster, dokument, uppföljningar och granskningar.

IEEE/IEA 12207.0 och -1 publicerades 1996 resp. 1997 och är en anpassning av ISO/IEC 12207 till elektronikindustrin. 12207.0 innehåller ISO/EIA standarden och har lagt till sex appendix. 12207.1 lägger tonvikt vid dokumenten och identifierar och namnger ett stort antal dokument. Den ger riktlinjer för dokumenten, men detaljerat innehåll och format specificeras inte och några Data Item Descriptions går inte att finna i standarden.

Referat ur [15288]:

ISO/IEC 15288 publicerades 2002 och är ett allmänt ramverk för att beskriva livscykeln i system, som konstrueras och byggs av oss, samt en mängd väldefinierade processer med tillhörande terminologi. Hela livscykeln omfattas från koncept till utfasning. Standarden tar med alla aktiviteter som behövs för att anskaffa eller leverera system som består av en eller flera av systemelementen: Maskinvara, programvara och operatörer. Den hanterar fyra huvudkategorier av processer: Upphandling, företag, projektstyrning och teknik. Utvärdering och processförbättring är en integrerad del av standarden i samverkan med ISO/IEC 15504.

Standarden är generell och man förväntas välja en lämplig delmängd och anpassa den till en organisations speciella behov antingen det gäller:

- En hel organisation, som vill skapa en komplett utvecklingsmiljö med väldefinierade processer. Dessa kan sedan kompletteras med en infrastruktur av metoder, procedurer, tekniker, verktyg och utbildad personal. Organisationen kan sedan använda denna utvecklingsmiljö för att genomföra och styra sina projekt genom produkternas hela livscykel.
- Ett projekt inom en organisation, då man väljer lämpliga delar ur den etablerade utvecklingsmiljön.
- En upphandling, då standarden hjälper köpare och säljare att komma överens om processer och aktiviteter.

Standarden beskriver en referensmodell med syfte och resultat från varje process. Processerna kan användas sekvensiellt och hierarkiskt. Indata till processerna lämnas emellertid därhän, då de anses svåra att definiera och standardisera.

Begränsningarna i standarden är:

- Detaljer i processerna som metoder, procedurer och tekniker tas inte upp.
- Detaljer i dokument, som namn, format, innehåll eller lagringsmedium, definieras inte.

ISO/IEC 15288 gäller utveckling och hantering av system med maskinvara, programvara och operatörer. Den beskriver 25 processer, 208 aktiviteter och 123 processresultat.

Referat ur [15504]:

ISO/IEC 15504 är ett ramverk för utvärdering av processer för utveckling och hantering av programvara. Den är avsedd att användas såväl för utvärdering internt inom en organisation, som extern utvärdering av leverantörer. En väsentlig egenskap hos standarden är, att den ger en processprofil och inte ett enstaka värde. Den ger värden för varje ingående process.

Standarden definierar en referensmodell för utveckling av programvara. Syftet är att skapa en gemensam bas för utvärderingar och att en utvärderare skall kunna modellera ett analysobjekt så att det överensstämmer med referensmodellen. Då kan man jämföra processer från olika organisationer.

Referensmodellen är tvådimensionell. Den ena dimensionen är processerna och syftet med dem, vilket beskrivs i mätbara termer. Definitionen av processerna baseras på ISO/IEC 12207. Den andra dimensionen är hur väl en organisation når syftet med varje process.

ISO/IEC 15504 gäller utvärdering av processer för utveckling och hantering av programvara. Den modellerar 5 processkategorier, 35 processer och 201 aktiviteter (base practices).

Wang [Wan00] analyserar några väl spridda standarder för kvalitetssäkring eller utvärdering och förbättring av utvecklingsprocesser för programvara: CMM, BOOTSTRAP, ISO/IEC 15504 och ISO 9001. Baserat på sin egen processalgebra definierar han så en egen referensmodell SEPRM, - Software engineering process system analysis - och relaterar den till de övriga. Resultatet är en modell med 3 delsystem av processer, 12 processkategorier, 51 processer och 444 grundläggande processaktiviteter, base process activities – BPA.

Gemensamt för alla modeller ovan är, att de koncentreras på processer eller aktiviteter. Informationen eller dokumenten är sekundära och namnges endast. Detaljerna som innehåll och format lämnas till den som instansierar processerna. Detta är en radikal skillnad jämför med tidigare standarder för utvecklingsprocesser, där dokumenten var det viktiga och specificerades i detalj.

Eftersom CM hanterar konfigurationsenheter och i en utvecklingsprocess är dokumenten konfigurationsenhet, ger aktuella standarder inte mycket hjälp i vår undersökning. IEEE/IEA är undantaget.

5.2 Utvecklingsmetoder inom försvarsindustrin

5.2.1 Projektstyrningsmodeller:

- PERT
Pert är en akronym för Program Evaluation and Review Technique, en projektstyrningsmodell med ursprung USAs försvarsorganisation Department of Defence:
<http://www.susning.nu/PERT>
- CPM
CPM är en akronym för en projektstyrningsmodell utvecklad vid företaget Du Pont:
<http://www.susning.nu/CPM>
- WBS
WBS är en akronym för Work Breakdown Structure en kompletterande projektstyrningsmodell till PERT med ursprung i USAs försvarsorganisation Department of Defence:
<http://www.susning.nu/WBS>
- PROPS, Projektet för Projektstyrning
PROPS är en projektstyrningsmodell som används inom Ericsson. Modellen används för alla projekt inom företaget och även hos flera av dess partners. Namnet är en akronym för Projektet för projektstyrning, som startade 1987 (en annan tolkning är PProject Operation and Planning System) och som 1989 levererade den första versionen av PROPS, version R1 som även innehöll arbetsmodellen AXE 10 Design System. Hela historiken finns beskriven i en bok, "Historien om PROPS", SV/LZT 108 104 R1.
En av utvecklarna bakom Rational Unified Process (RUP) från Rational Software, (Ivar Jacobsson, även en av Tre amigos), har ett förflutet inom Ericsson, och detta är förmodligen anledningen till att delar av RUP liknar PROPS. Läs mer om PROPS hos Semcon:
http://www.semcon.se/spm/eng/about_props.html
<http://www.susning.nu/PROPS>

- RUP
RUP står för Rational Unified Process, en produktifiering från Rational Software, Inc. av programutvecklingsmetodikerna "unified process" (UP), som utvecklades i början av 1990-talet utifrån tankarna om objektorienterad design. Starkt förknippad med RUP är modelleringsspråket UML och dess diagram.
RUP är egentligen inte en process utan ett ramverk för en process. Den konkreta processen som ett projekt följer uppstår när RUP konfigureras. Denna konfiguration är en mycket svår balansgång som kräver stor insikt i projektet och inte så lite fingertoppskänsla. En felkonfigurerad RUP kan bli antingen diktatoriskt rigid eller ostrukturerat kaotisk. Många företag tillämpar RUP i en eller annan form. Ericsson har en egen likartad metodik (vilken även används för andra projekt) som kallas PROPS.
<http://www.susning.nu/RUP>
- PPS
PPS är en projektstyrningsmodell som marknadsförs av TietoEnator:
<http://www.susning.nu/PPS>
- MIPS
MIPS, Metodik inom Posten i Sverige - är en projektstyrningsmodell som används av Posten:
<http://www.susning.nu/MIPS>
- Dynamic Systems Development Method (DSDM)
DSDM är en lättviktsmetod och ett erfarenhetsbaserat ramverk för projektstyrning och systemutveckling med tyngdpunkt på medverkan från användare, prioritering tillsammans med kunden, iterativ utveckling och testning som driver utvecklingen framåt.
<http://www.precipio.se/courses/aCourseSWE.asp?courseID=139&courseCategoryId=1>
- LOTS

5.2.2 Utvecklingsmodeller

- V-modellen.
Utvecklingsmetodik, maskinvara och programvara, GKE ElektornikAB:
<http://www.gke.org/rapporter/GKE%20UTVECKLINGSMETODIK.pdf>
- The Delta Method,
Delta - a method for constructing computer systems on the basis of users' needs, WM-data AB, ERS AB:
<http://www.deltamethod.net/>

5.2.3 Metoder och Processer

- XP
XP, eXtreme Programming, är en lättviktsmetod och bygger på jordnära och mänskliga tankesätt. Det finns många synpunkter på om detta är bra eller dåligt:
<http://www.precipio.se/courses/aCourseSWE.asp?courseID=133&courseCategoryId=1>
- ITIL, Infrastructure Library
Process specialdesignad för företag som tillhandahåller IT-tjänster:
<http://www.precipio.se/courses/aCourseSWE.asp?courseID=220&courseCategoryId=1>
- UML
UML, Unified Model Language, en notation för modulera system:
<http://www.precipio.se/courses/aCourseSWE.asp?courseID=11&courseCategoryId=1>

5.3 Projektstyrningsmodeller inom försvarsindustrin

Mikroprojektet har fått information från bara en SESAM-medlem.

5.3.1 SaabTech

Inom SaabTech används PROPS som generell projektledningsmodell (ursprung Ericsson). Respektive division inom SaabTech har en egen adaptation/applikation för hur man ska tillämpa

den generella modellen inom sin verksamhet. Denna PROPS-applikation/adaption är vägledande för genomförande av projekt beträffande styrning, ansvar, roller, processer m.m. Inom T-divisionen är dessutom PROPS-applikationen integrerad med Projekteringsmodellen, Produktframtagningssmodellen och Produktförvaltningsmodellen. Applikationen har huvudfokus på kundorderprojekt men tillämpas även för interna projekt och FoU-projekt. Samtliga modeller ingår i verksamhetssystemet och är åtkomliga via intranätet.

Not. PROPS modellen omfattar styrnings-, lednings- och operativa processer samt definierar tollgate och milstolpar.

5.4 Aktuella CM-verktyg

Som inledning till en mer omfattande översyn av tillgängliga verktyg har mikroprojektet studerat Inotion.

5.4.1 Inotion

För:

- Projektplanering och uppföljningsstödet verkar bra. Jag har dock ingen koll på hur sådana verktyg brukar vara.
- Binder-begreppet låter som lämpligt för baselines; att samla samhörande dokument.
- Enhetligt projektverktyg för alla.
- Underlättar rapportering av tid, kostnad, progress.
- Förenklar granskningar.
- Underlättar arbete över olika orter.

Mot:

- Det känns som om iNotion har tänkt sig att ha kopior av dokument och tillhandahålla en produktstruktur, och stöd för granskningar och livscyklar. Det låter precis som PDM. Tänk på att t.ex. mekanik redan har infraPDM (för att göra något innan PDM fanns?).
- Kommer programvarudokument att finnas i kopior i CCM, Argus (Metaphase), PDM(IFS), iNotion och slutligen i Arkivet?
- Hur mycket jobb är det att använda iNotion på API och processsidan? Jag menar jämfört att skriva något med script som genererar html (eller manuellt skapa en egen hemsida, eller ett grupprum, på intranätet?)
- Har vi råd att integrera iNotion med våra verktyg och arbetsflöden?
- Har vi råd att utbilda alla?
- Kan vi få stöd att införa detta från alla och få dem att använda iNotion (företagsledning, marknad, utveckling, produktion, underhåll)?
- Blir vi låsta till Windows?
- Ska vi ha det både i röd och gul miljö?

Synpunkter:

- I iNotion spårar man från WBStasks till iNotion tasks och vidare till dels dokument i iNotion och filer i CCM. För koppling till CCM vore det bättre att spåra till CCM tasks, än källkodsfiler.
- Pekare till CCM verkar bra, men jag undrar om man vill peka på låsta projekt i stället för filer.
- Det borde vara möjligt att öppna ”project view” i CCM, men då måste man förstås använda sig av CCM.

Frågor:

- Hur kommer det att fungera med röd och gul miljö?
- Hur väljer man när det skall vara länkar till filer i CM system, och när lägger man in kopior i iNotion. Hur vet man att kopiorna inte lever vidare i CCM, utan att checkas ut i iNotion?

- Det här med referensen till CM systemet förstod jag inte; man har väl inte URLar i CCM. Någon kanske kan göra ett cgi-skript till CCM som hämtar ut info via http.

6. Begrepp i systemutvecklingsprocessen

Kapitlet är en början till informationsanalys av ett CM-system. Det är ett diskussionsunderlag för fortsatt arbete.

6.1 Terminologi

- Process
- Processmönster
- Arbetsflöde
- Beslut, -projekt, -konstruktion
- Aktivitet – analys, konstruktion, programmering, inspektion, provning, planering, mätning, uppföljning, styrning, kommunikation, validering, verifiering, evaluering
- Startkriterier (start criteria)
- Färdigkriterier (end criteria)
- Input
- Output
- Resurs
- Team
- Roll – utvecklare (analytiker, arkitekt, programmerare, integratör, provare), utvecklingschef, produktchef, kvalitetschef, projektledare, företagsledare, beställare, användare
- Metod
- Teknik
- Regel
- Checklista
- Informationsmodell
- Arkitektur
- Konstruktion (Artefact) – dokument, modell, program, provdata
- Baseline
- Version

6.2 Objekttyper

- Förfrågan
- Krav, delkrav, sammansatta krav
- Restriktion
- Kravtyper – kund-, specifikations-, programvaru-, utvecklings-, konstruktions-
- Planer, resurs-, tid-
- Kalkyl
- Kontrakt
- Dokument – Funktionsspecifikation, SRS, SDD, protokoll
- Anteckningar under utveckling som t ex konstruktionsbeslut.
- Användningsfall, uppdrag
- Prov, provfall, provningsdata, förväntade resultat
- Provtyper, enhets-, integrations-, system-
- Granskningsrapport
- Risklista
- Problemlista

- Ändringsbegäran
- Felrapport
- Konstruktionsmönster
- Tjänst
- Gränssnitt, funktions-, konfigurations-, provnings-
- Objekt, -typ, -instans
- Funktion
- Beteende
- Parameter
- Konfigurationsenhet
- Modul
- Komponent
- Klass
- Programdel
- Maskinvara
- Plattform
- Operatör

7. Processen konfigurationsstyrning

En beskrivning av processen nedan är hämtad ur [15288]. Motsvarande finns också i [12207]

7.1 Syfte

Syftet med konfigurationsstyrningsprocessen är, att etablera och förvalta alla resultat från ett utvecklingsarbete och göra dem tillgängliga för alla intressenter.

7.2 Resultat

En instansiering av processen för konfigurationsstyrning ger:

- En definierad strategi för konfigurationsstyrning
- Objekt som kräver konfigurationsstyrning, identifierade och definierade
- Etablerade nivåer (baselines) för konfigurationer
- Styrning av konfigurationsobjekt
- Överensstämmelse mellan produkt och specifikationer
- Styrd leverans av konfigurationsobjekt
- Information om aktuellt tillstånd för varje konfigurationsobjekt under hela livscykeln

7.3 Aktiviteter

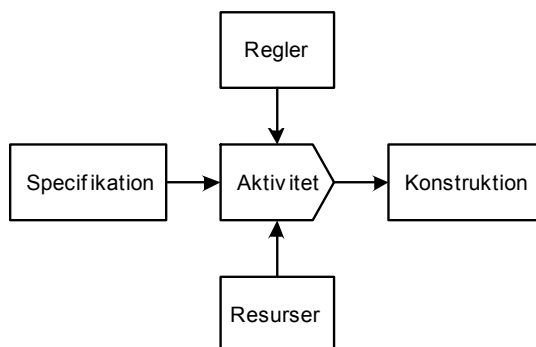
Följande aktiviteter skall med hänsyn till organisationens policier implementeras i konfigurationsstyrningsprocessen:

- Definiera en strategi för konfigurationsstyrning
- Identifiera de objekt, som skall vara konfigurationsobjekt
- Förvalta konfigurationsinformationen med lämplig nivå på integritet och intrångssäkerhet
- Etablera en instans för ändringshantering för att ge samordnad granskning, evaluering och ordning av dokumenterade och berättigade ändringsförslag
- Kontrollera en produkts överensstämmelse med specifikationer och andra överenskommelser.

8. En generell modell av delprocesser

8.1 Process kontra information

Vi ser systemutveckling som ett stort informationsbehandlingssystem och alltså informationen som viktigast. I allmänhet är informationen i ett system tidsinvariant, medan sättet att hantera informationen varierar över tiden.



Specifikation är information som aktiviteten utnyttjar. Det kan vara all slags information från välstrukturerade och väldefinierade dokument till ostrukturerad bakgrundsinformation och domänkunskap. Konstruktion är information aktiviteten producerar. Den bör vara ett väldefinierat och välstrukturerat dokument, som fastlagts i beskrivningen av aktiviteten.

Regler bestämmer hur aktiviteten skall genomföras och skall också vara fastlagda i beskrivningen av aktiviteten. De kan baseras på en allmän standard, anpassas till organisationen och anpassas till projektet, processen och aktiviteten. Hur mycket och när de kan ändras bör fastläggas i förväg, men i moderna lättviktsprocesser bör reglerna kunna ändras i en pågående aktivitet. Hur detta påverkar kvalitet och pålitlighet är en fråga att fundera över.

Resurser är tid, pengar och kompetens. Notera, att personer inte nödvändigtvis tillför resurser.

Informationsklasser enligt EEE/IEA 12207:

Beskrivning	Beskriver en existerande eller planerad funktion, konstruktion, prestation eller process.
Plan	Definierar när, hur och av vem givna aktiviteter skall genomföras med optioner och alternativ när så behövs.
Procedur	Definierar i detalj när och hur givna uppgifter skall genomföras och vilka verktyg som skall användas.
Arkiv (record)	Beskriver det material en organisation sparar, som t ex kvalitetsmätningar, affärshandlingar, ekonomiska data, konstruktionsöverväganden och konstruktionsbeslut.
Rapport	Beskriver resultatet av en aktivitet, som t ex undersökning, utvärdering eller prov.
Begäran	Sparar information som behövs för att ge ett svar
Specifikation	Specificerar krävda funktioner, prestanda eller processer, som kravspecifikation.

9. En fallstudie

Med fallstudien exemplifieras hur ett typiskt projekts dokumenthantering kan gestalta sig med avseende på CM.

I exemplet utgår vi från ett antal dokumenttyper som identifierats från två olika processbeskrivningar:

- Praktisk ProjektStyrning, PPS och
- VerksamhetsSystem (VS), hos Saab Bofors Dynamics

PPS är en projektstyrningsmodell som marknadsförs av TietoEnator AB.

VS hos Saab Bofors Dynamics är ett ”reglemente” som ställer krav på hur olika verksamheter skall bedrivas, för att uppnå mål vad avser tid, kostnad och kvalitet, t ex för utveckling av programvaruenheter.

Från PPS har vi utgått från följande tre grundprocesser vid val av dokumenttyper:

- Åtagande: Varför, hur, vad, när, överenskommelser.
- Ledning: Analysera, värdera, beslut, verkställa, delegera, motivera.
- Återkoppling: Nuläge, jämför åtagande, varför differenser, åtgärder.

Från Saab Bofors Dynamics VS har vi utgått från processen ”Utveckla programvaruenhet” vid val av dokumenttyper. Processen innehåller bl a följande faser:

- Analysera programvarukrav, SSR, Utforma arkitektur, PDR, Detaljutforma, CDR, Implementera och enhetstesta.
- Planera integration och test, Utarbeta testfall.
- Integrera och testa, TRR1, CSCI-test, TRR2, BP2 och BP3.

I tabellen nedan redovisas hur de olika dokumenttyperna i ett typiskt projekt lagras i tre olika lagringsverktyg, vilka har olika huvudsyften:

- Utvecklings CM (Utv.CM): Används i första hand av konstruktörer för att konfigurationsstyra tekniska filer t ex programvarufiler.
- Projekt CM (Proj.CM): Används i första hand av projektstyrande personal för lagring av framförallt textfiler.
- Arkiv: Syftar i detta fall på det lagringsverktyg som formellt är företagets långtidsarkivering ur ett juridiskt perspektiv.

Av tabellen framgår även vilken processbeskrivning som refererar dokumenttypen samt en gruppering av dokumenttyperna som definierar dokumenttypernas huvudsyfte.

Dokumenttyp	Gruppering	PPS	VS	Utv. CM (3)	Proj. CM (2)	Arki v (1)
SRS	Kravstyrande		x	x	x	x
Kravdatabas	Kravstyrande		x	x		
UseCasebeskrivning	Kravstyrande		x	x	x	x
Operational Concept	Kravstyrande		x	x	x	x
Teknisk Rapport	Kravstyrande		x	x	x	x
Interface specifikation	Kravstyrande		x	x	x	x
Kontrakt	Kravstyrande	x			x	x
SDD	Resultat		x	x	x	x
UML modell	Resultat		x	x		
Källkod	Resultat		x	x		x
Konstruktionsbeslut	Resultat		x	x		
Informal technical notes	Resultat		x	x		
Prototyp kod (experiment)	Resultat		x	x		

Anpassad preprocessor kod	Resultat		x	x		x
Byggskript	Resultat		x	x		x
Objektkod	Resultat		x			
Exekverbar kod	Resultat		x	x	x	x
SDF	Resultat		x	x		
Enhetstest bygge	Resultat		x	x		
Testdefinitionsfiler	Resultat		x	x		
Testresultatfiler	Resultat		x	x		
STP	Resultat		x	x	x	x
STD	Resultat		x	x	x	x
STR	Resultat		x	x	x	x
VDD	Resultat		x	x	x	x
Programmerarens dagbok	Arbetsdokument					
Integratörens dagbok	Arbetsdokument					
Projektledarens dagbok	Arbetsdokument	x		x	x	
Fax	Arbetsdokument				x	
Mail	Arbetsdokument					
Projektmötesprotokoll	Arbetsdokument	x		x	x	
Kallelse	Arbetsdokument					
CCB protokoll	Arbetsdokument		x	x	x	
Risklista	Arbetsdokument	x			x	
Loose Ends lista	Arbetsdokument			x		
Projekthemsida	Arbetsdokument			x		
Tidplan	Projektstyrning	x		x	x	
Kalkyl	Projektstyrning	x			x	
Projektplan	Projektstyrning	x		x	x	
SDP	Projektstyrning		x	x	x	
Metodbeskrivning	Projektstyrning		x	x		
Dokumentmall	Projektstyrning		x	x		
Statusrapport ur CM	Kvalitetsdokument		x			
Baseline beskrivning	Kvalitetsdokument		x	x	x	
Metricsammanställning	Kvalitetsdokument	x		x	x	
Granskningsprotokoll	Kvalitetsdokument		x	x	x	
Protokoll från Tekniska genomgång	Kvalitetsdokument		x	x	x	
Protokoll från Beslutsmöte	Kvalitetsdokument	x		x	x	
Arbetsbesked	Kvalitetsdokument	x		x		
PR	Kvalitetsdokument	x	x	x		
PR-task	Kvalitetsdokument	x	x	x		
ÄB, ÄO – extern ändringshantering	Kvalitetsdokument	x			x	

10. Fortsatt arbete

Frågor att studera:

- Vad är ett konfigurationsobjekt i systemutvecklingsprocessen?
- Kan även regler för en aktivitet konfigurationsstyras?
- Kan CM ge oss bättre styrning på vårt utvecklingsarbete?
- Vilka delar av en utvecklingsprocess låter sig inte med fördel styras med hjälp av CM?

Bakgrundsinformation:

- Eureka Software Factory – vart tog den vägen och varför?

- Aspse, Kapse och Mapse - vart tog de vägen och varför?
- PCTE – genomarbetad standard, varför används den inte?
- XML for UML – kolla på nätet.
- ArgoUML – kolla på nätet.

Aktuella utbytesformat:

- STEP
- CALS
- XML
- Express, som är en utvidgning av STEP
- RDF – resource description format for the semantic web

Hur hanteras CM i allmänt använda utvecklingsmetoder som RUP och Agile Methods?