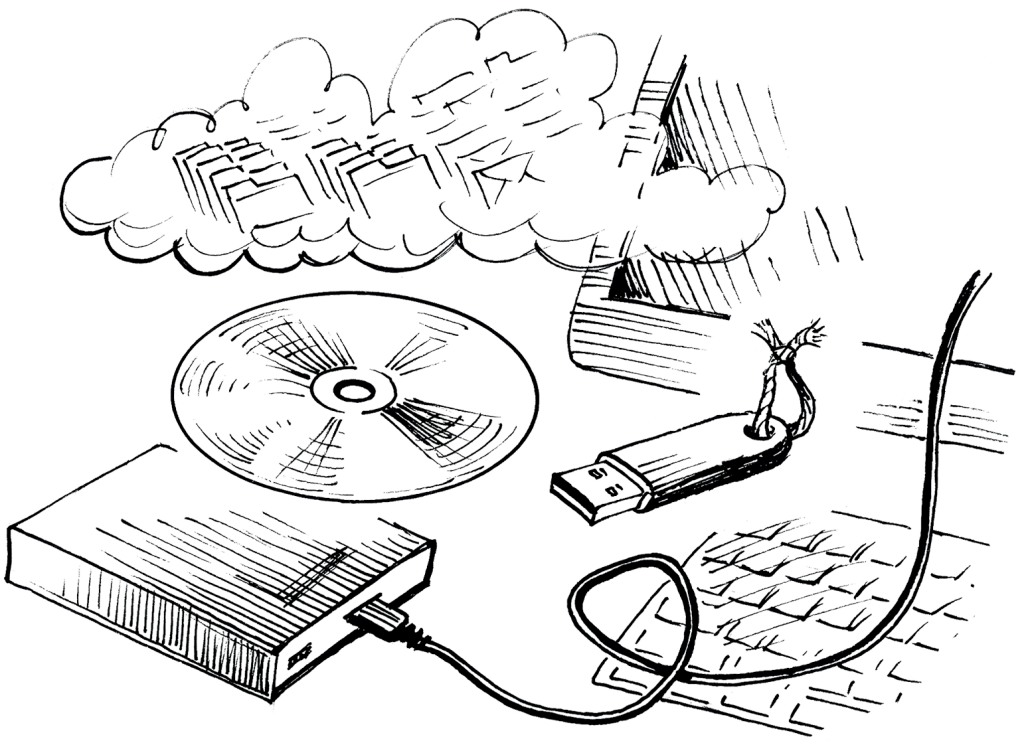
Handbok för programvara i säkerhetskritiska tillämpningar



2018

Försvarets materielverks handbok
Programvara i säkerhetskritiska tillämpningar

H ProgSäk 2018

FMV

Öppen/Unclassified **BESLUT**

Datum	Diarienummer	Ärendetyp
2017-12-11	15FMV2367-28:1	3.5
	Dokumentnummer	Sida
	ange	1(1)
Giltig t.o.m.	Upphäver	
t.v	H ProgSäk 2001	

Beslutande
Kristin Strömberg
FMV Teknisk Direktör

Föredragande
Svante Wåhlin, FMV

Fastställande av Handbok Programvara i säkerhetskritiska tillämpningar 2018 (H ProgSäk 2018)

Handbok Programvara i säkerhetskritiska tillämpningar, H ProgSäk, 2018 års utgåva, M7762-001041 H PROGSÄK 2018, fastställs att gälla från och med 2018-01-01.

Den tidigare utgåvan H ProgSäk 2001 M7762-000531, Plan 14910:3476/01, daterad 2001-12-13 och den engelska utgåvan, H ProgSäk E M7762-000621, VO StraMtrl 14910:17567/2004, daterad March 15, 2005 samt M7702-000461 CD H ProgSäk upphävs att gälla 2018-01-01.

I beredningen av beslutet har Jonas Persson, SPL Armé (Teknisk Chef Mark), Jan Ericsson, SPL Marin (Teknisk Chef Sjö), Axel Nilsson, SPL Flyg (Teknisk Chef Flyg & Rymd), Lars Burström, SPL LED (Teknisk Chef Led), Bo Persson, SPL LOG (Teknisk Chef Log), Mats Hallberg, T&E TL, Jan Söderberg, FSV Ledningsstöd, Lars Lange, SPL Stab S&D samt Svante Wåhlin, SPL Stab S&D deltagit, den senare som föredragande.

FÖRSVARETS MATERIELVERK

Kristin Strömberg
Teknisk Direktör

Central lagerhållning: Försvarets bok- och blankettförråd
Tryck: Arkitektkopia Växjö, mars 2018

FÖRORD

Omfattning

FMV har i denna handbok sammanställt programvarustandarder, handböcker och erfarenheter av säkra konstruktioner för tekniska system som innehåller programvara i säkerhetskritiska tillämpningar. Detta för att om möjligt undvika ohälsa, olycksfall, skada på miljö och ekonomisk skada vid användning av militära tekniska system samt för att öka förtroendet för tekniska system vid utbildning, övning och insats inom Försvarsmakten.

Denna handbok baseras på Försvarsmaktens syn på systemsäkerhetsverksamhet och grundas på metodiken i Försvarsmaktens Handbok för Systemsäkerhet (H SystSäk). För att uppnå kravställd tolerabel risknivå finns såväl generella konstruktionsinriktade krav som verksamhetskrav i H SystSäk, och de mer specifika kraven på utformning och framtagningsprocess för programvara i säkerhetskritiska tillämpningar finns i denna handbok.

Denna handbok riktar sig främst till aktörer som kravställer, anskaffar, utvecklar, modifierar eller hyr in tekniska system som innehåller programvara i säkerhetskritiska tillämpningar och kan sålunda tillämpas för andra aktörer inom svenska staten.

Tillämpning av handboken

Handbok Programvara i säkerhetskritiska tillämpningar (H ProgSäk) har ingen juridisk status utan dess användning regleras utifrån avtal eller föreskrift. Dess tillämpning vid anskaffning, utveckling, modifiering och inhyrning av tekniska system regleras av myndighetens egen designorganisation. Handboken upplyser om lämpliga krav att ställa vid en upphandling där programvara kommer att ingå i säkerhetskritiska tillämpningar. Detta för att få säkrare konstruktionslösningar samt att ge bakgrundsinformation, hänvisningar och rekommendationer utifrån krav från militär verksamhet. Kraven får formuleras om för att bättre harmonisera med aktuellt tekniskt system.

Standarder innehåller vanligtvis olika exempel på dokumenterad kunskap. Att följa en standard är frivilligt och en hänvisning till en standard ska ses som en rekommendation för att uppfylla föreskrifter eller EG-direktiv. I vissa föreskrifter förekommer dock en direkt hänvisning till en specifik standard som då ska följas. Om FMV i beställning har ställt krav på att utvecklande industri ska följa en viss standard, blir även den därigenom tvingande.

Erfarenhet visar att det kan finnas motstridiga krav mellan standarder från olika teknikområden och detta får hanteras från fall till fall.

Läsanvisning

Den läsare som är obekant med systemsäkerhetsverksamhet bör läsa Försvarsmaktens Handbok Systemsäkerhet (H SystSäk). För vapen och ammunition hänvisas till FMV Handbok Vapen- och ammunitionssäkerhet (H VAS), vilken även innehåller ett särskilt avsnitt om programvara i tändsystem.

H ProgSäk är ett komplement till H SystSäk och kan i huvudsak läsas och tillämpas fristående men hänvisningar till texten i H SystSäk förekommer i vissa avsnitt. För vissa funktioner eller delsystem kan både H ProgSäk och H VAS behöva tillämpas parallellt.

Bilaga 3 och *4* är viktiga för den övergripande förståelsen av arbetssättet. Där ges exempel på arbetsgången från Försvarsmaktens kravdokument till industrins utvecklingsarbete. *Kapitel 3* ger en förenklad beskrivning av arbetsgången mellan Försvarsmakten och FMV. Samordningsavtalet (SamO), som är upprättat mellan Försvarsmakten och FMV, bör läsas för kompletterande information om arbetssättet.

Kravnumrering

Handboken innehåller i *kapitel 8* krav som ska ställas på leverantören vid upphandling av tekniska system och produkter. Avsnitt inleds med fakta och förklarande text till kraven. I förekommande fall finns även kommentarer till kraven.

Kraven i handboken är numrerade efter följande princip: 2.801.03-A där:

2	prefix för krav i H ProgSäk
801	kapitel 8, avsnitt 1 (= 8.1)
03	löpnummer
A	administrativt krav (infogas oftast i Verksamhetsåtagandespecifikation, VÅS)
T	tekniskt krav (infogas oftast i Teknisk specifikation, TS)

Förbättringsförslag

Förslag om förbättringar av H ProgSäk skickas till:

FMV

Systemsäkerhet

115 88 Stockholm

e-post: systemsakerhet.fmv@fmv.se

Innehåll

1 Handbokens inriktning

1.1 Bakgrund

Handbok Programvara i säkerhetskritiska tillämpningar 2018 (H ProgSäk 2018) är en vidareutveckling av tidigare utgåvor (2001 svensk och 2005 engelsk). Föregående utgåva var en gemensam handbok mellan Försvarsmakten och FMV. Denna utgåva är en FMV-publikation.

1.2 Syfte

Syftet med denna nya utgåva av H ProgSäk är att kunna ställa rätt krav på utveckling av programvara för att erhålla säkra tekniska system. Handboken innehåller grundläggande krav motsvarande lägsta accepterade kritikalitetsnivå (GKPS) och vägledning för arbetet. För samtliga högre kritikalitetsnivåer ska etablerad programvarustandard följas utöver handbokens grundläggande krav. I denna handbok ges exempel på vanligt förekommande etablerade programvarustandarder.

Programvara för säkerhetskritiska tillämpningar finns i produkter och system från exempelvis ammunition, med en mycket begränsad storlek på programvaran, till mycket stora och komplexa system med många ihopkopplade datorer och många programvaror. Exempelvis kan ett ledningssystem innefatta hundratals datorer eller finnas som ett integrerat datorsystem på ett fartyg. Handboken ska kunna användas oberoende av det tekniska systemets storlek eller komplexitet.

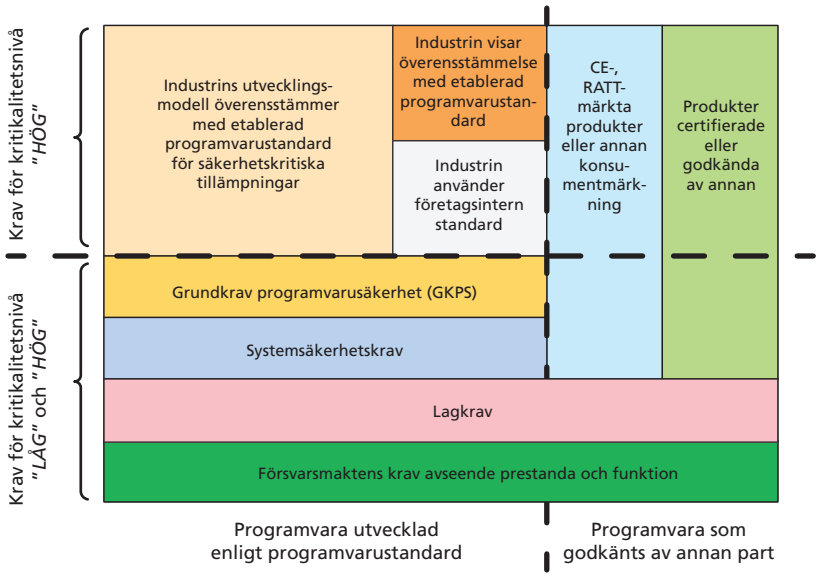


Bild 1:1 Illustration av krav för tekniska system med säkerhetskritisk programvara

Denna handbok beskriver en metodik för att på ett tidigt stadium i utvecklingen definiera en säkerhetsarkitektur så att de mest kritiska felen, som kan orsakas av det säkerhetskritiska datorsystemet, bland annat kan reduceras med hjälp av tillförda säkerhetsfunktioner.

Kritikaliteten hos det säkerhetskritiska datorsystemet är beroende dels på konsekvens vid en tänkt olycka, dels på kravet för sannolikhet för vådahändelse, dels hur säkerhetsarkitekturen för systemet är utformad. Målet med att utforma en säkerhetsarkitektur är att kunna reducera kritikalitetsnivån på det säkerhetskritiska datorsystemet så långt som detta är praktiskt möjligt i det tekniska systemet. Om kritikalitetsnivån på datorsystemet kan sänkas så kan normalt även programvarans kritikalitet sänkas. En metodik för att initialt bestämma kritikaliteten hos det säkerhetskritiska datorsystemet och därigenom ingående programvara beskrivs i denna handbok.

I denna handbok används ett antal grundläggande begrepp med innebörd enligt nedanstående.

Säkerhetsfunktion

”Tillförd funktion vars syfte är att reducera sannolikheten för att vådahändelse ska inträffa vid fel i den säkerhetskritiska funktionen.”

Källa: H ProgSäk

Säkerhetskritisk funktion

”Funktion som styr eller övervakar energier och som vid fel kan leda till vådahändelse och i förlängningen olyckor.”

Källa: H ProgSäk

Säkerhetskritiskt datorsystem

”Datorsystem som direkt eller indirekt styr eller övervakar energier och som vid fel kan orsaka vådahändelse och i förlängningen olyckor.”

Källa: H ProgSäk

Säkerhetskritisk programvara

”Programvara som styr eller övervakar energier och som vid fel kan orsaka vådahändelse och i förlängningen olyckor.”

Källa: H SystSäk

Programvara

”Innehåller datorprogram, procedurer, regler, tillhörande dokumentation och data.”

Källa: AQAP 2210

Data

”Avser information, ofta lagrad som filer eller databaser, som programvaran använder då den ger funktion eller genererar annan information.”

Källa: H ProgSäk

Kritikalitet

”Ett relativt mått på den inverkan ett tekniskt system har på systemsäkerheten.”

Källa: H ProgSäk

Vid utformning av tekniska system som innehåller programvara i säkerhetskritiska tillämpningar är det viktigt med en genomtänkt säkerhetsarkitektur, ett strukturerat arbetssätt och balanserade tekniska krav. Handbokens grundkrav syftar till att uppnå detta även om de särskilda militära krav som ställs på tekniska system i militär miljö vid utbildning, övning och insats kan medföra att de tekniska förutsättningarna för att åstadkomma en optimal systemeffekt kan vara svår att uppnå. Då stridseffekt och säkerhet står i konflikt med varandra vägs dessa samman och proportionerliga insatser görs för att skapa en optimal avvägning mellan säkerhet och stridseffekt. Sådan avvägning sker i samråd mellan Försvarsmakten och FMV.

1.3 Innehåll

Handboken visar på en metod för både nationell och internationell upphandling av en säker (engelskans ”safe”) programvara, även för icke säkerhetskritiska tillämpningar, som en del i ett tekniskt system, system av system eller för en separat produkt.

Handboken beskriver åtgärder som krävs av de olika inblandade aktörerna, för att få fram ett säkert teknisk system eller produkt mot de ställda kraven. Det är viktigt att alla aktörer såsom Försvarsmakten, FMV och utvecklande industri bidrar till att det tekniska systemet eller produkten ska bli tillräckligt säker. Handboken ska därför användas för tekniska system och produkter inom alla teknikområden.

H ProgSäk är ingen standard för, och ersätter heller ingen standard för, programvaruframtagning. Handboken beskriver inte hur programmering går till eller hur programvara ska tas fram.

Handboken är rådgivande och innehållet är rekommendationer över metoder för att kunna uppnå ställda krav på tekniska system och produkter och kan göras tvingande genom att införas i kontrakt till utvecklande industri.

Handbokens grundkrav (GKPS) gäller för framtagning av all programvara och för alla tekniska system där programvara ingår. För programvara i icke säkerhetskritiska tillämpningar är skälet att säkerställa möjlighet till framtida systemuppdateringar. Vid framtagning av programvara för initial kritikalitetsklassificering ***HÖG*** enligt denna handbok ska, förutom grundkraven (GKPS), en etablerad programvarustandard tillämpas. Handboken presenterar ett urval av rekommenderade standarder inom området för säkerhetskritisk programvara.

En företagsintern standard kan inte ersätta en etablerad standard. Om en företagsintern standard tillämpas ska korsreferenser finnas till en etablerad standard. Användning av en företagsintern standard ska i förekommande fall överenskommas med FMV i samband med kontraktsgenomgången.

Då det finns många olika programvarustandarder inom området och en del begrepp har olika definitioner i dessa, så innehåller H ProgSäk bilagor med begreppsförklaringar. Vidare är det viktigt att ha kännedom om att det finns skilda definitioner inom olika programvarustandarder. Detta bör omhändertas i en Systemsäkerhetsledningsplan (SSMP).

Handboken innehåller krav på att utvecklande industri, innan utvecklingsarbetet får påbörjas, ska redogöra för vilken eller vilka programvarustandarder som man avser följa. Kravverifiering och eventuell anpassning av standarden ska överenskommas med FMV.

Handboken tar upp krav på innehåll i den dokumentation som ska levereras till FMV och Försvarsmakten.

Handboken innehåller en allmän beskrivning av möjlig hantering kring problematiken med komplex elektronik, till exempel programmerbara elektroniska kretsar.

Det är viktigt att kritikalitetsklassificering av programvara görs på ett korrekt och spårbart sätt. Handboken belyser fördelarna men även problematiken kring detta. Metoder för nedbrytning av överliggande systemsäkerhetskrav beskrivs också i handboken. Handboken innehåller de rekommenderade arbetssteg som man bör ha med i planeringen vid framtagning av programvara och dess dokumentation.

Handboken tangerar flera närliggande områden och dessa beskrivs summariskt i *kapitel 12*. Exempelvis omfattar handboken inte området informationssäkerhet (security/information security), men man bör komma ihåg att man inte kan få ett säkert tekniskt system, om man inte har beaktat såväl safety som security. Det kan finnas motstridiga krav mellan systemsäkerhet och informationssäkerhet och dessa krav måste hanteras parallellt för att undvika dåliga lösningar eller att kraven blir kostnadsdrivande.

Handboken omfattar inte heller vilka krav man ska ställa på utvecklingen av programvara utifrån hur säker man behöver vara på att den levererar önskad funktion att motverka fientlig (eller egen) bekämpning (motmedelssystem, igenkänningssystem) eller hur viktig funktionen är för att genomföra ett visst uppdrag (mission critical). Det finns erfarenhet från utländska utvecklingsprojekt för militära system där metodik för kritikalitetsklassificering tillämpats som innebär att kritikalitetsklassificeringen av programvara (och därmed krav på utvecklingen av programvaran) enligt aktuell programvarustandard kunnat tillämpas också för dessa aspekter.

Bild 1:2 Omfattning och närslutna områden till programvara i säkerhetskritiska tillämpningar

1.4 Tillämpning

Lagstiftningen kan medge undantag för militära tekniska system och produkter respektive för militär verksamhet. Tekniska system och produkter som är avsedda för Försvarsmakten och som normalt endast används vid krigs- eller beredskapstillstånd samt under fältmässiga övningar kan behöva kompletterande konstruktionsinriktade krav för att uppnå krav på tolerabel risknivå. Civil lagstiftning kan vara styrande vid övningar där militär verksamhet bedrivs parallellt med civil verksamhet, exempelvis i civilt luftrum eller vid insatser som inte kan förväntas ske i krig eller under beredskap. Undantag från lagstiftningen innebär givetvis inte en avvikelse från arbetsmiljölagens huvudregel att alla arbetstagare ska skyddas mot ohälsa och olycksfall. Skyddet mot ohälsa och olycksfall ska tillgodoses för Försvarsmaktens personal på samma sätt som för andra arbetstagare i samhället. Ytterligare information finns i H SystSäk.

Denna handbok behandlar programvara i säkerhetskritiska tillämpningar vilken på de översta systemnivåerna kan påverka systemsäkerheten och därför klassas som säkerhetskritisk. Handboken tar upp de krav som ska ställas på programvara med en initial kritikalitetsklassificering benämnd ***LÅG*** motsvarande person-, ekonomisk och miljöskadeklass IV i H SystSäk. Det rekommenderas att även programvara som inte har systemsäkerhetspåverkan, men som man avser att livscykelhantera, upphandlas med samma krav och procedurer. I denna handbok benämns dessa krav som ”GrundKrav ProgramvaruSäkerhet” (GKPS) och framgår av *kapitel 8*.

Syftet med dessa grundkrav (GKPS) är att programvaran ska erhålla tillräcklig kvalitet och att man ska erhålla dokumentation för granskning och konfigurationsledning för att möjliggöra framtida systemuppdateringar.

För upphandling av programvara med kritikalitetsklassificering benämnd ***HÖG*** förordar denna handbok att utvecklande industri använder en etablerad programvarustandard som ställer krav på processerna kring utveckling av säkerhetskritisk programvara.

Produkter som kan innehålla programvara för utförande av viss verksamhet, och som redan finns på marknaden (COTS-produkter), där systemuppdateringar av ingående programvara inte är planerad, kan upphandlas som en produkt. Programvara i produkter som är CE-märkta för fristående användning och som inte får integreras i militära system eller som inte heller kommer att modifieras, såsom medicintekniska produkter och mätinstrument, hanteras enligt *kapitel 10*.

Hantering av tidigare utvecklad programvara, så kallad *Previously Developed Software* (PDS), tas upp i *kapitel 11*.

Denna handbok kompletterar enbart krav på utformning av säkerhetskritisk programvara, hantering av data samt integrering av PDS-programvara.

För särskild verksamhet såsom medicinsk teknisk utrustning tillkommer verksamhetsspecifika krav på utformning av programvaran genom olika lagar, förordningar och författningssamlingar. Utformning av programvara och underhållsutrustning som stöd för olika typer av underhållsarbeten hanteras i denna handbok.

1.5 Aktörer

Aktörerna i denna handbok är Försvarsmakten, FMV och utvecklande industri. Rollerna kan i normalfallet beskrivas enligt nedan:

Försvarsmakten	Brukare
FMV	Designansvarig
Utvecklande industri	Utvecklare/integratör

I de fall Försvarsmakten åtar sig att vara Designansvarig rekommenderas att kraven i *kapitel 7* följs. Om Försvarsmakten eller FMV påtar sig rollen som utvecklare/integratör behöver kraven i *kapitel 8* följas. Denna handbok kan även användas om Försvarsmakten eller FMV själva anskaffar programvaror för eget bruk.

1.6 Tillämpning internationellt

Vid framtagning av denna handbok har hänsyn tagits till de EU-förordningar, EU-direktiv och harmoniserade standarder som används internationellt, varför handboken bedöms vara tillämplig i sin helhet även vid internationell upphandling. Då utvecklingsuppdrag läggs hos en utländsk leverantör ska systemsäkerhetsverksamhet genomföras enligt samma förfarande som hos svenska leverantörer.

Vid köp av färdigutvecklade system utomlands ska alltid tillses att information och dokumentation erhålls så att utvärdering av systemsäkerheten kan genomföras.

1.7 Övriga kunder och andra myndigheter

Denna handbok riktar sig främst till aktörer som anskaffar, utvecklar eller uppdaterar programvara i tekniska system. Detta gäller även om man hyr in tekniska system.

FMV kan använda denna handbok vid upphandlingar eller i samarbeten med andra myndigheter såsom Fortifikationsverket (FORTV), Försvarets forskningsinstitut (FOI) och Försvarets radioanstalt (FRA).

2 Lagar, standarder och handböcker

Syftet med detta kapitel är att ge en introduktion och bakgrund utifrån lagar, de vanligaste tillämpade programvarustandarderna samt några handböcker inom området.

2.1 Lagkrav för användning av programvara i produkter

Lagar och förordningar är ofta skrivna på ett övergripande sätt och ger inga detaljer för användning av programmerbara system. Det är ovanligt att hitta stöd för hur man får utveckla och använda programvara i säkerhetskritiska tillämpningar. Tidigare kunde det förekomma att programmerbara system uttryckligen förbjöds i säkerhetsfunktioner. Innan den programmerbara tekniken ansågs mogen föreskrevs ofta reläbaserad logik för säkerhetsrelaterade funktioner. I militära tekniska system kan det dock fortfarande finnas skäl att tillämpa konservativ kravställning och teknik i säkerhetskritiska tillämpningar.

Formuleringar i lagar och förordningar är ofta skrivna teknikneutralt, vilket innebär att kraven uttrycks på sådant sätt att det inte spelar någon roll vilken teknik systemen byggs med. Säkra funktioner kan realiseras med olika tekniker såsom pneumatik, hydraulik, pyroteknik, elektriska kretsar, elektronik eller programvara. Det viktiga är att utvecklande industri har genomfört en systemsäkerhetsanalys och sedan använt tekniker och metoder för att undvika fel vid konstruktion och för att hantera fel under drift som kan leda till olyckor. En genomtänkt säkerhetsarkitektur ska därför prioriteras i konstruktionsarbetet.

2.2 Europeiska regelverk

Inom den europeiska unionen eftersträvas att harmonisera lagstiftningen inom flera områden. Därför utfärdas europeiska direktiv vilka förväntas införlivas i de olika medlemsstaternas lagstiftning och föreskrifter. Ett direktiv är bindande med avseende på det resultat som ska uppnås, men överlåter åt de nationella myndigheterna att bestämma tillvägagångssättet för genomförandet.

Flera direktiv, såsom lågspänningsdirektivet, EMC-direktivet, radiodirektivet, maskindirektivet och direktivet för medicinteknisk utrustning hanterar produktsäkerhet. När en tillverkare intygar att produkten uppfyller de grundläggande hälso- och säkerhetskraven enligt alla tillämpliga direktiv kan produkten CE-märkas som tecken på detta. Särskilt farliga produkter ska tredjepartsgranskas av ett särskilt kontrollorgan (notified body) som i sin tur utfärdar ett kontrollintyg, vilket ligger till grund för CE-märkningen.

Deklaration om överensstämmelse (Declaration of Conformity, DoC) ska gälla alla tillämpliga direktiv för produkten. En maskin i verkstadsindustrin omfattas ofta av både maskindirektivet, lågspänningsdirektivet och EMC-direktivet. Jordbruksmaskiner, maskiner i verkstadsindustrin, paketeringsmaskiner, tryckerimaskiner och automatiska dörrar är exempel på maskintyper som omfattas av maskindirektivet. Bland de maskintyper som maskindirektivet inte omfattar finns vapen, motorfordon och fartyg.

En fördel med gemensamma regler för produktsäkerhet är att en produkt kan marknadsföras i flera medlemsländer i ett och samma utförande, utan en upprepad godkännandeprocedur i varje enskilt medlemsland. De grundläggande hälso- och säkerhetsföreskrifterna är desamma i EU:s alla medlemsstater. Kraven i direktiven är obligatoriska och måste uppfyllas för att produkten ska få sättas på marknaden.

För vissa teknikområden utfärdar EU förordningar, som i sin tur pekar direkt på de regelverk som ska följas. För fordonsområdet pekar EU-förordningar direkt på ECE-reglementen.

2.3 Standardisering

De europeiska direktiven anger grundläggande hälso- och säkerhetskrav utan att i detalj gå in på vad detta betyder. Detaljerade frågor kring teknisk utformning hänskjuts till standarder. Detta arbetssätt innebär att direktiven blir stabila och inte behöver ändras i samma takt som dagens tekniknivå ändras. Ett exempel på denna princip är maskindirektivet som till exempel bland annat hävdar att *”Ett styrsystem ska vara konstruerat och tillverkat så att riskfyllda situationer inte ska kunna uppstå. …”*. För att söka vägledning vad direktivets formulering betyder i praktiken får man läsa de europeiska standarder som handlar om maskiners styrsystem.

Standarder uppdateras regelbundet. Det är vanligt att en standard kommer i ny utgåva omkring vart femte år.

För att uppfylla kraven i direktiven kan man välja att följa så kallade harmoniserade europeiska standarder, utfärdade av CEN, CENELEC eller ETSI. Att standarderna är harmoniserade innebär att de är granskade och uppfyller kraven i motsvarande direktiv. CEN är den allmänna europeiska standardiseringsorganisationen kring maskiner med mera, CENELEC omfattar huvudsakligen standardisering inom det elektrotekniska området. ETSI utvecklar standarder inom telekommunikation. I Sverige sköts standardiseringen genom SIS och SEK.

Det finns även andra standardiseringsorganisationer som arbetar med att ta fram standarder, men som nästan alltid är knutna till en viss bransch. Exempel på sådana organisationer är *Society of Automotive Engineers* (SAE) och *Radio Technical Commission for Aeronautics* (RTCA). Militära standarder ges ut av *NATO Standardisation Agency* (NSA). Bland militära standarder finns även Def Stan, MIL-STD och *Standard NATO Agreement* (STANAG) inklusive *Allied Ordnance Publication* (AOP).

En standard är frivillig att följa. Lagar och förordningar är däremot tvingande och de ställer krav på produktsäkerhet beträffande till exempel maskinsäkerhet, elmiljö (EMC), explosionsskydd (ATEX) och elsäkerhet (LVD). En tillverkare av en utrustning kan välja ett annat sätt än det som beskrivs i standarder för att uppnå en produkt med god säkerhet. Det är ofta mycket arbetsammare

att bygga upp en egen säkerhetsbevisning, men det kan vara nödvändigt i de fall som den egna produkten ligger före den teknikniv som förutsätts i motsvarande standard.

Kunden som köper en utrustning har givetvis alltid möjlighet att ställa krav på att vissa standarder ska uppfyllas, även om landets lagar inte kräver det. Ett exempel på detta är stora företag vilka ofta har företagsinterna regler för hur utrustning i deras anläggningar ska vara utformad, till exempel beträffande elektrisk installation.

2.4 Standarder och handböcker för programvara i säkerhetskritiska tillämpningar

I *avsnitt 2.5 – 2.19* redovisas exempel på de vanligast förekommande programvarustandarderna, samt ett par handböcker inom området, som vunnit bredast kännedom kring livscykelhantering, kritikalitetsklassificering samt tekniker och metoder att tillämpa på säkerhetskritiska programvaror i civila och militära tekniska system. Standarden IEC 61508 anses vara förlaga till flera av de sektorspecifika standarderna.

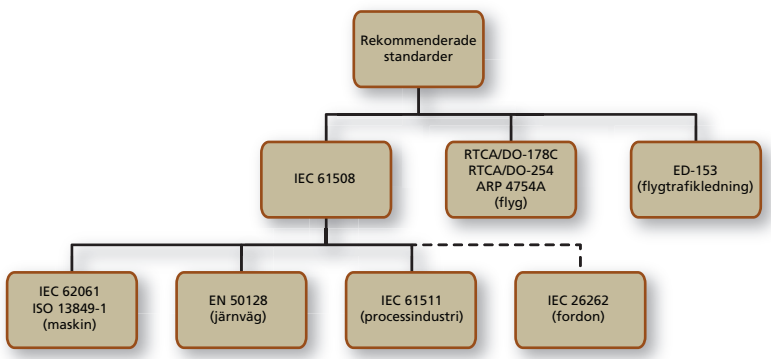


Bild 2:1 Relationer mellan rekommenderade standarder

Syftet med avsnitten nedan är att översiktligt beskriva olika programvarurelaterade standarder och handböcker såsom innehåll, omfattning och tillämpningsområde. Vidare besvaras även ett antal konkreta metodikrelaterade frågor. För full förståelse och för att kunna använda sig av standarden, bör användaren ha den specifika standarden tillgänglig.

> Användare av viss standard ska alltid införskaffa den från utgivaren. Dels för att ha tillgång till den senaste utgåvan, dels för eventuell upphovsrätt.

I handbokens jämförelse mellan standarderna redovisas ett antal aspekter avseende likheter och skillnader mellan dessa. Standarderna är olika till sina upplägg och innehåll, vilket medför att jämförelser måste inkludera många olika aspekter och representationer. I *bilaga 1* finns denna sammanfattning och jämförelse i tabellform över följande egenskaper:

- administrativa aspekter
- kritikalitetsklassificering
- teknisk omfattning
- tekniker och metoder
- metodik.

I tabellerna i *bilaga 1* relateras det till det huvudsakliga alternativet, exempelvis om en standard huvudsakligen är till för programvara, men om systemaspekter nämns till en liten del så blir bedömningen ändå programvara. I *tabell B1:2 Kritikalitetsklassificering* finns en jämförelse avseende kritikalitetsnivåer mellan olika standarder.

2.5 ISO/IEC 61508 (Elektriska/elektroniska/programmerbara elektroniska system)

IEC 61508 *Functional safety of electrical/electronic/programmable electronic safety-related systems* är en internationellt etablerad programvarustandard.

Standarden är utvecklad inom *International Electrotechnical Commission* (IEC) men har även antagits som europeisk standard och kallas då EN 61508. Dessutom finns den som svensk standard med namn SS-EN 61508. Samma information finns således i IEC 61508, EN 61508 och SS-EN 61508.

2.5.1 Innehåll och omfattning

Standarden berör hela livscykeln men är inte harmoniserad med maskindirektivet eftersom standarden är generisk. Det innebär att den används som bas för sektorspecifika standarder, såsom IEC 62061 (maskiner), IEC 61513 (kärnkraft) och IEC 61511 (processindustri).

Standarden fokuserar på säkerhetsfunktioner. Om kravet på felfrekvens är mindre strängt än 10^{-5}/timme är IEC 61508 inte avsedd att tillämpas. Standarden anger att om en säkerhetskritisk funktion har ett krav på lägre felfrekvens än 1×10^{-5}/timme kan hela funktionen ses som en säkerhetsfunktion och därmed blir standarden tillämplig på hela utvecklingsprocessen, vilket kan vara värt att beakta om den ska tillämpas vid generell programvaruutveckling.

Standarden använder den så kallade V-modellen, vilket gör den tillämpbar på processer som är uppbyggda enligt denna modell.

Standardens första fyra delar är normativa medan de övriga tre är informativa. Se *tabell 2:1*.

Tabell 2:1 ISO/IEC 61508 olika delar i standarden

Del	Rubrik	Normativ/informativ
1	General requirements	Normativ
2	Requirements for electrical/electronic/ programmable electronic safety-related systems	Normativ
3	Software requirements	Normativ
4	Definitions and abbreviations	Normativ
5	Examples of methods for the determination of safety integrity levels	Informativ
6	Guidelines on the application of IEC 61508-2 and IEC 61508-3	Informativ
7	Overview of techniques and measures	Informativ

Notera att krav på programvara finns i del 3, medan hårdvara och system båda ingår i del 2. Att informativa delar inkluderas bör tolkas som att de visserligen inte är obligatoriska, men att de ändå är starkt rekommenderade. Liksom i alla standarder finns tolkningsutrymme. Några exempel på icke preciserbara formuleringar är *should*, *consider*, *ensure*, *be detailed* och *appropriate*. Relevanta bedömningar och tolkningar av dessa termer behöver därför göras.

Standarden omfattar faser i hela livscykeln för en eller flera säkerhetsfunktioner i ett tekniskt system. Standarden kan tillämpas på hela, eller del av, en säkerhetsfunktion.

Om ett tekniskt system integrerar flera säkerhetsfunktioner med övrig icke-säkerhetskritisk styrning behöver hela systemet hanteras som säkerhetskritiskt. Det är därför önskvärt att särskilja säkerhetsfunktionerna från ”den vanliga styrningen” för att inte få en alltför dyrbar konstruktions- och verifieringsprocess, det vill säga att försöka att få någon form av oberoende mellan dessa. Om standarden ska kunna tillämpas på en del av säkerhetsfunktionen krävs att oberoende kan påvisas mot övriga delar.

Kraven på integritet (riskreduktion) hos säkerhetsfunktionerna bedöms enligt *Safety Integrity Level 1–4* (SIL 1-4), där SIL 4 ställer de högsta kraven. En funktion tar in insignaler, analyserar dem och sätter utsignaler. I funktionen kan både hårdvara och programvara ingå, men andra sätt att minska olycksrisker, till exempel genom mekanisk förstärkning, ingår inte.

Lämplig SIL-nivå bestäms genom att genomföra riskanalys och ju större olycksrisk desto högre värde på SIL behövs. I standarden indelas konsekvenser enligt skada på människor, utrustning, omgivning, informationssäkerhet och ekonomisk skada. Standarden kopplar sannolikheten för farliga fel till varje SIL-nivå. Ju högre SIL-nivå desto lägre sannolikhet för farliga fel tolereras. Kraven på lägre sannolikhet för farliga fel ökar alltså med ökande SIL-nivå.

I standardens del 5 Annex E visas hur SIL beräknas. Observera dock att del 5 är informativ, det vill säga att man inte måste följa den. För att få fram SIL-nivån använder man sig av en riskmatris som är baserad på konsekvensen av en oönskad händelse (vådahändelse), samt sannolikheten för att händelsen inträffar. I värdet för inträffandesannolikheten ingår frekvensen med vilken risken för händelsen bedöms vara förekommande samt sannolikheten för att säkerhetsfunktionerna inte lyckas undvika händelsen när risken väl förekommer.

Ett exempel på säkerhetsfunktion är ett överhastighetsskydd för en bearbetningsmaskin (betecknas som EUC – *Equipment Under Control*). Säkerhetsfunktionen behövs för att minska olycksrisken för att en person skadar sig. Notera att standarden inte säger ingenting om framtagning av EUC, utan den är helt fokuserad på säkerhetsfunktionen.

IEC 61508 skiljer mellan sannolikheter för *Low demand mode*, *High demand mode* och *Continous operation*. För *Low demand mode* anges sannolikheten för felfunktion då säkerhetsfunktionen behöver användas. För *High demand mode* eller *Continuous operation* anges felsannolikheten per timme. Säkerhetsfunktioner som används mer sällan än en gång per år räknas som *Low demand mode*. För *High demand mode* samt *Continuous mode* anges istället säkerhetsfunktionens medelvärdesbildade felfrekvens (i enheten ”per timme”).

Hela styrsystemet för *Equipment Under Control* (EUC) kan i vissa fall, då kravet på felfrekvens är $<10^{-5}$/timme, betraktas som en säkerhetsfunktion i *Continuous operation*.

Genom att betrakta tåligheten mot hårdvarufel (*Hardware Fault Tolerance*, HFT) och felsäkerhetskvot (*Safe Failure Fraction*, SFF) kan standarden beskriva den maximala SIL-nivå som kan uppnås med en viss konstruktionsarkitektur. Kraven på tålighet mot hårdvarufel ökar med högre SIL-nivå.

För hårdvara ska ett antal värden beräknas för att kontrollera att SIL-nivån uppfylls, medan olika metoder specificeras för programvara, det vill säga inga beräkningar görs för programvaran. För hårdvaran finns i standardens del 2 Annex A och B *Techniques and Measures* och motsvarande finns i del 3 Annex A och B

för programvara. Alla SIL-beroende krav i programvara finns samlade i dess *Techniques and Measures*. Annex B i del 3 är informativ, men bör också användas för programvaran. I *Techniques and Measures* för programvara anges metoder och tekniker med avseende på SIL-nivå.

Standardens Del 3 innehåller ett antal sub-faser och krav på dessa (*Software Safety Lifecycle Requirements*, SSLR). För fasen *Software Design and Development* finns dessutom sub-sub-faser och krav på dessa är definierade.

Eftersom standarden innehåller många krav, behöver en *Functional Safety Assessment* (FSA) för bedömning alltid göras. Relevant dokumentation ska göras tillgänglig om valet av SIL-nivå anger detta.

2.5.2 Tillämpningsområde

Standarden IEC61508 är generisk och oberoende och har ingen speciell civil eller militär aspekt. Standarden gäller specifikt för säkerhetsfunktioner men många delar av standarden, dock inte alla, kan användas för hela det tekniska systemet. Inga speciella krav finns avseende skador på egendom eller miljö. Ingen koppling görs heller till verksamhetsområden såsom mark, sjö eller luft. Ett område som undantas i standarden är medicinsk utrustning som istället täcks av standarden IEC 60601 *Elektrisk utrustning för medicinskt bruk*.

Standarden avser inte att täcka informationssäkerhetsaspekter. Den bedöms dock vara en de facto-standard för komponentleverantörer av certifierade och standardiserade komponenter för industrin, exempelvis sensorer, ställdon och logikelement med tillhörande programvara samt rena programvarukomponenter som kommunikationsstackar och drivrutiner.

2.6 ISO 26262 (Vägfordon)

Standarden ISO 26262 *Road Vehicles - Functional Safety* (svensk titel: *Vägfordon – Funktionssäkerhet i el- och elektroniksystem* är en internationell standard avsedd för fordonsindustrin och första utgåvan gavs ut 2011. Standarden har tagits fram av ISO Technical Committee ISO/TC 22, Road vehicles, Subcommittee SC 3, Electrical and electronic equipment. Den behandlar hela livscykeln från konceptframtagning, till systemkonstruktion, hårdvaruutveckling, programvaruutveckling, utvärdering samt användning och underhåll.

2.6.1 Innehåll och omfattning

Standarden består av tio delar och programvara behandlas huvudsakligen i del 6. Se *tabell 2:2*.

Tabell 2:2 ISO 26262 olika delar i standarden

Del	Rubrik
1	Vocabulary
2	Management of functional safety
3	Concept phase
4	Product development at the system level
5	Product development at the hardware level
6	Product development at the software level
7	Production and operation
8	Supporting processes
9	Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses
10	Guideline on ISO 26262

Standarden ISO 26262 bygger i huvudsak på standarden IEC 61508, men är en sektorspecifik version för funktionssäkerhet inom fordonsindustrin. Detta beror bland annat på att fordonsindustrin arbetar med produktion i stora serier. Begreppet *Start of Production* (SOP) är centralt. Validering av säkerheten utförs före produktionsstart. Arbetet delas in i konceptfas, produktutveckling och arbete efter produktionsstart. Standardens Del 7 ställer

krav som gäller vid produktionen, vilket saknas i IEC 61508. Det hävdas också att ett stort antal av funktionerna i ett fordon är säkerhetsrelaterade och hela tiden inverkar på fordonets säkerhet, men de funktionerna har inte karaktären av utpräglade säkerhetsfunktior som exempelvis överlastskydd.

Standarden menar att systemsäkerhet uppnås genom åtgärder med olika tekniker såsom mekaniska, hydrauliska, pneumatiska, elektriska och programmerbara elektroniska system. Även om standarden rör funktionssäkerhet i elektriska och elektroniska system ger standarden ett ramverk som kan användas också för andra tekniker.

Standarden beskriver processen ur ett livscykelperspektiv. Den inledande konceptfasen tar bland annat upp definition, riskanalys och koncept för funktionssäkerhet. Utvecklingsarbetet beskrivs på systemnivå samt för hårdvara och programvara. Livscykeln omfattar också aktiviteter efter produktionsstart.

ISO 26262 har ett sätt att klassificera olycksrisker som är anpassat för fordonsindustrin. Beroende på allvarlighet, sannolikhet och kontrollerbarhet väljs en ASIL klass (A, B, C eller D), där D motsvarar den högsta graden av riskreducering. Allvarligheten (severity) klassas från S1 till S3, där S3 är den högsta allvarlighetsgraden. Sannolikheten (probability) klassas från E1 till E4, där E4 är den högsta sannolikheten. Kontrollerbarheten (controllability) är ett mått på hur väl en förare kan hantera en uppkommen situation och klassas från C1 till C3, där C3 är en situation som inte kan hanteras av föraren. De största olycksriskerna bedöms i en matris i standarden där situationen beskrivs med S3, E4 och C3. Om någon av variablerna får en risk klassad som S0, E0 eller C0 så ansätts inga värden för de övriga variablerna. För dessa risker tilldelas heller ingen ASIL, utan standarden bedömer att normal kvalitetsstyrning (*Quality Management*, QM) är tillräckligt.

För programvaruutveckling erbjuder standarden en referensmodell som kallas V-modell. Arbetet med programvaruutveckling ska anpassas, men ska vara baserat på denna referensmodell.

2.6.2 Tillämpningsområde

Fordonsindustrin har behov av gemensamma riktlinjer för hur man ska hantera säkerhetskritiska inbyggda system och standarden har därför kommit till bred användning. Den är viktig, inte minst eftersom antalet ”smarta” funktioner i fordon ökar kraftigt. Nya stödfunktioner som avses öka säkerheten presenteras och branschen planerar för funktionalitet avseende självkörande person- och lastbilar.

Standarden är avsedd att tillämpas på säkerhetsrelaterade system som innehåller ett eller flera elektriska/elektroniska delsystem installerade i en serietillverkad personbil med en totalvikt upp till 3 500 kg. Standarden avser inte elektriska/elektroniska system i specialfordon som till exempel handikappfordon. Eftersom standarden drar en gräns vid 3 500 kg omfattas inte lastbilar och bussar. Branschen refererar ändå ofta till ISO 26262 eftersom en motsvarande standard saknas för tyngre fordon. I kommande utgåva 2 av ISO 26262 (2018) är begränsningen på 3 500 kg borttagen och en del som hanterar motorcyklar kommer också att finnas.

Standarden hanterar risker beroende på felfunktion hos elektriska och elektroniska system, men hanterar inte risker såsom elsäkerhet, skydd mot eld, rök, strålning, gifter eller korrosion, såvida detta inte direkt orsakas av felfunktion hos ett elektriskt eller elektroniskt system. Standarden behandlar inte heller prestanda hos elektriska eller elektroniska system, även om det finns standarder för funktionsprestanda bland annat för bromssystem, krockkuddar, farthållare och automatbromsar.

Standarden är enbart inriktad på personsäkerhet och utelämnar skador på egendom och miljö. Standarden avser heller inte att täcka informationssäkerhetsaspekter.

2.7 EN ISO 13849-1 (Maskinstyrningar)

Standard EN ISO 13849-1 *Maskinsäkerhet – Säkerhetsrelaterade delar av styrsystem – Del 1: Allmänna konstruktionsprinciper* ger säkerhetskrav och vägledning om konstruktions-principer och integration av säkerhetsfunktioner i styrsystem för maskiner. Standarden har sitt ursprung i EN 954-1 och ISO 13849-1.

Standarden har utarbetats av ISO/TC 199, *Safety of Machinery*, som är en teknisk kommitté inom den internationella standardiseringsorganisationen ISO och har övertagits som

EN ISO 13849-1:2008 av den europeiska standardiseringen CEN/TC 114, *Safety of Machinery*. En svensk språkversion SS-EN ISO 13849-1:2015 har givits ut och har övertagits som EN ISO 13849-1. Den första versionen av EN ISO 13849-1 gavs ut 2008 och en uppdaterad version gavs ut 2015.

2.7.1 Innehåll och omfattning

De delar av en maskins styrsystem som är avsedda för skyddsfunktionerna kallas säkerhetsrelaterade delar i styrsystem (Safety Related Parts/Control Systems, SRP/CS). Dessa kan bestå av hårdvara och programvara. Förutom skyddsfunktioner kan SRP/CS även hantera funktioner för driften av maskinen såsom tvåhandsmanöveranordning eller stopp.

Arbetsmetodiken som förutsätts för maskiner är att konstruera bort olycksrisker, skydda mot kvarstående olycksrisker och, i de fall där inget annat är möjligt varna för kvarstående olycksrisker. Som en del i den övergripande riskreduceringsstrategin för en maskin försöker konstruktören ofta vidta åtgärder för att minska olycksriskerna genom att använda tekniska skydd med en eller flera skyddsfunktioner.

Behovet av riskreducering bedöms genom att kombinera skadans allvarlighetsgrad med exponeringsfrekvensen. Riskbedömningen utgår från en situation innan den avsedda skyddsfunktionen tillämpas och gäller alltså oavsett om programvara ingår i SRP/CS eller om logiken är byggd med annan teknologi.

Beroende på riskanalysen utformas skyddsfunktionerna med olika prestandanivå (*Performance Level*, PL). Standarden specificerar egenskaper för de säkerhetsrelaterade delarna i styrsystemet inklusive erforderlig prestandanivå för att utföra skyddsfunktionerna. Prestandanivåerna är indelade i fem nivåer efter sannolikheten per timme för farlig felfunktion. Detta probabilistiska angreppssätt skiljer sig från det kvalitativa angreppssätt, till exempel tålighet mot enkelfel, som förut varit vanligt för att beskriva prestanda för maskinstyrningar. Sättet att uttrycka sannolikhet för felfunktion kan jämföras med standarden IEC 61508, som också arbetar med felsannolikheter.

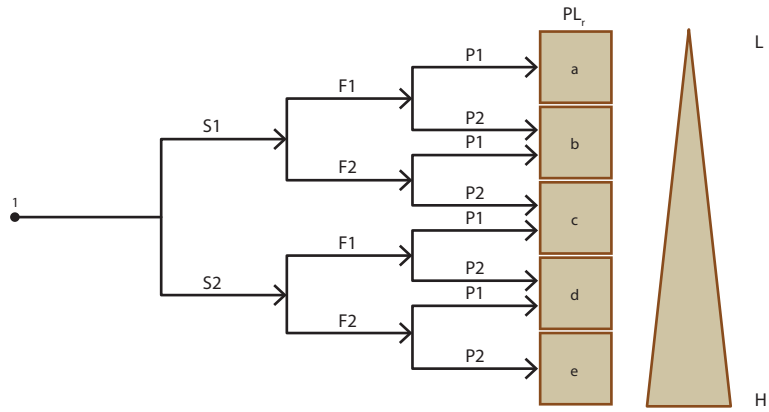
Alla maskinstyrningar förutsätts vara i *high demand* eller i *continuous mode*. Därför kan sannolikhetsvärdena jämföras med SIL 1-3 enligt IEC 61508.

Tabell 2:3 Prestandanivåer (Performance Levels, PL) återgiven från EN ISO 13849-1

Prestandanivåer	Genomsnittlig sannolikhet för farlig felfunktion per timme
a	$\geq 10^{-5} - < 10^{-4}$
b	$\geq 3\times10^{-6} - < 10^{-5}$
c	$\geq 10^{-6} - < 3\times10^{-6}$
d	$\geq 10^{-7} - < 10^{-6}$
e	$\geq 10^{-8} - < 10^{-7}$

Anmärkning: Förutom den genomsnittliga sannolikheten för farlig felfunktion per timme måste också andra åtgärder vidtas för att uppfylla prestandanivån (PL).

Behovet av riskreducering kan vara svårt att bedöma. Genom att kombinera skadans allvarlighetsgrad med exponeringsfrekvensen kan en bedömning görs. Se exempel i *bild 2:2* nedan. Riskbedömningen utgår från en situation innan den avsedda skyddsfunktionen tillämpas och gäller oavsett om programvara ingår i SRP/CS eller om logiken är byggd med annan teknologi.



Förklaring

1 startpunkt för utvärdering av en skyddsfunktions bidrag till riskreducering
L lågt bidrag till riskreducering
H högt bidrag till riskeducering
PL$_r$ erforderlig prestandanivå

Riskparametrar

S skadans allvarlighet
S1 lätt (vanligtvis övergående skada
S2 svår (vanligtvis obotlig skada eller dödsfall)
F frekvens och/eller exponeringstid för riskkällan
F1 sällan till mindre ofta och/eller kort exponeringstid
F2 ofta till kontinuerlig och/eller lång exponeringstid
P möjlighet att undvika riskkällan eller begränsa skadan
P1 möjligt under vissa omständigheter
P2 knappast möjligt

Bild 2:2 Riskdiagram för att fastställa erforderlig prestandanivå enligt EN ISO 13849-1

Prestandanivån fastställs genom uppskattning av följande aspekter:

- medeltid till farlig felfunktion
- *Mean Time to Dangerous Failure* ($MTTF_d$-värdet för enskilda komponenter)
- feldetekteringsförmåga, *Diagnostic Coverage* (DC)
- gemensam felorsak, *Common Cause Failures* (CCF)
- strukturen
- skyddsfunktionens beteende vid feltillstånd
- säkerhetsrelaterad programvara
- systematiska fel
- förmåga att utföra en skyddsfunktion under förväntade miljövillkor.

För att underlätta bedömningen av den uppnådda prestandanivån tillämpar standarden en metodik som bygger på kategorisering enligt specifika konstruktionskriterier och specificerade beteenden vid feltillstånd. Dessa kategorier tilldelas en av fem nivåer, kallade kategori B, 1, 2, 3 och 4. Se *tabell 2:4*.

Tabell 2:4 Sammanfattning av kraven för olika kategorier återgiven från EN ISO 13849-1

Kategori	Sammanfattning av kraven	Systemets beteende	Principer för att uppnå säkerhet
B	SRP/CS och/eller deras skyddsutrustning, liksom deras komponenter, ska konstrueras, tillverkas, väljas, monteras och kombineras enligt relevanta standarder, så att de kan motstå förväntad påverkan. Grundläggande säkerhetsprinciper ska tillämpas.	Feltillstånd som inträffar kan leda till förlust av skyddsfunktionen.	Huvudsakligen genom val av komponenter
1	Kraven i B ska uppfyllas. Väl beprövade komponenter och väl beprövade säkerhetsprinciper ska användas	Feltillstånd som inträffar kan leda till förlust av skyddsfunktionen, men sannolikheten att de inträffar är lägre än i kategori B.	Huvudsakligen genom val av komponenter
2	Kraven i B ska uppfyllas och väl beprövade säkerhetsprinciper ska användas. Skyddsfunktion ska kontrolleras med lämpliga intervaller av maskinens styrsystem.	Feltillstånd som inträffar kan leda till förlust av skyddsfunktion mellan kontrolltillfällen. Förlust av skyddsfunktion detekteras genom kontroll.	Huvudsakligen genom systemets struktur
3	Kraven i B ska uppfyllas och väl beprövade säkerhetsprinciper ska användas. Säkerhetsrelaterade delar ska konstrueras så att: • ett enstaka feltillstånd i någon av dessa delar inte leder till förlust av skyddsfunktionen och • närhelst det är praktiskt möjligt detekteras det enstaka feltillståndet.	När ett enstaka feltillstånd inträffar kvarstår alltid skyddsfunktionen. Några men inte alla feltillstånd detekteras. Ackumulering av ej detekterade feltillstånd kan leda till förlust av skyddsfunktionen	Huvudsakligen genom systemets struktur

Kategori	Sammanfattning av kraven	Systemets beteende	Principer för att uppnå säkerhet
4	Kraven i B ska uppfyllas och väl beprövade säkerhetsprinciper ska användas. Säkerhetsrelaterade delar ska konstrueras så att: • ett enstaka feltillstånd i någon av dessa delar inte leder till förlust av skyddsfunktionen och • det enstaka feltillståndet detekteras när, eller före, skyddsfunktionen påkallas första gången, men om denna detektering inte är möjlig ska en ackumulering av ej detekterade feltillstånd inte leda till förlust av skyddsfunktionen.	När ett enstaka feltillstånd inträffar kvarstår alltid skyddsfunktionen. Detektering av ackumulerade feltillstånd minskar sannolikheten för förlust av skyddsfunktionen (hög DC). Feltillstånden detekteras i tid för att förhindra förlust av skyddsfunktionen.	Huvudsakligen genom systemets struktur

Aspekterna på prestandanivå kan grupperas i kvantifierbara aspekter (till exempel medeltid mellan farliga fel, $MTTF_d$-värdet för enskilda komponenter, *Diagnostic Coverage* (DC), *Common Cause Failure* (CCF), struktur), respektive icke kvantifierbara, kvalitativa aspekter som påverkar beteendet hos SRP/CS (till exempel skyddsfunktionens beteende vid feltillstånd, säkerhetsrelaterad programvara, systematiskt fel och miljövillkor). Standarden visar vilka prestandanivåer som är möjliga att uppnå med olika val av kategori. För att uppnå den högsta prestandanivån ska styrsystemet vara konstruerat enligt kategori 4 och dess $MTTF_d$-värde ska vara högt.

Standarden ger vägledning för utveckling av programvara genom att ge övergripande krav på styrsystem som använder programmerbara elektroniska system. Den hänvisar ofta vidare till standarden IEC 61508 för detaljerade tekniker och metoder. Säkerhetskrav på programvara ställs bland annat genom livscykelaktiviteter för att programvaran ska vara läsbar, begriplig samt möjlig att testa och uppdatera. Aktiviteterna syftar i första hand till att undvika feltillstånd som uppstår under programmets livscykel.

Ofta finns programvaran inbyggd i styrsystemet på ett sådant sätt att den inte är avsedd att påverkas av användaren. Detta kallar standarden för säkerhetsrelaterad inbyggd programvara (*Safety Related Embedded Software*, SRESW). Det finns också programvara som utvecklas av konstruktören för den speciella maskinstyrningen så kallad säkerhetsrelaterad applikationsprogramvara (*Safety Related Application Software*, SRASW). Eftersom SRESW och SRASW hanteras på olika sätt skiljer sig också kraven för dessa åt.

Standarden behandlar både programspråk med begränsat språkomfång (*Limited Variability Language*, LVL) och programspråk som inte har begränsat språkomfång (*Full Variability Language*, FVL). LVL används ofta för PLC-system inom automation där programmeringen är strikt styrd genom exempelvis programmering i funktionsblock. FVL innebär att programmeraren kan skriva sina konstruktioner fritt, exempelvis i vanliga högnivåspråk som C och Ada.

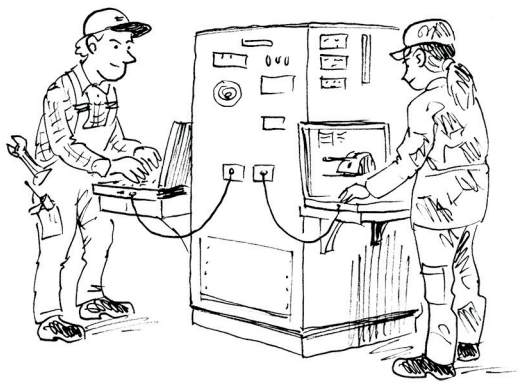


Bild 2:3 Industrin utvecklar och programmerar datorsystemet medan användaren anger parametrar inom godkända intervall

EN ISO13849-1 behandlar även programvarubaserad parametrisering av säkerhetsrelaterade parametrar. Detta betraktas som en säkerhetsrelaterad aspekt av konstruktionen och ska beskrivas i specifikationen av programvarusäkerhetskraven. Parametriseringen ska göras med ett anpassat programvaruverktyg från SRP/CS-tillverkaren och säkerheten för alla data som används vid parametriseringen ska upprätthållas.

I en informativ del av standarden ges exempel på typiska aktiviteter för att realisera SRESW

- tillämpning av V-modell för programvarans säkerhetslivscykel (definierat *Safety Life Cycle*, SLC)
- verifiering av programvaruspecifikation
- programmeringsregler på programstrukturnivå
- programmeringsregler vid användning av variabler
- programmeringsregler på funktionsblocksnivå.

Standarden EN ISO 13849-2 beskriver validering. Denna del tar även upp validering av säkerhetsrelaterad programvara, men beskriver endast valideringsaktiviteterna översiktligt.

2.7.2 Tillämpningsområde

Grundläggande hälso- och säkerhetskrav för maskiner inom EU ges genom det europeiska maskindirektivet. För att få detaljerad information om säkerhetsaspekter hänvisas till standarder. Standarden EN ISO 13849-1 är avsedd att ge vägledning till dem som arbetar med konstruktion och bedömning av styrsystem för maskiner. Standarden ger ingen specifik vägledning för överensstämmelse med andra EU-direktiv.

Komponenter som gränslägesbrytare och programmerbara styrsystem (PLC), vilka kan intygas av tillverkaren, kan användas i säkerhetsfunktioner med en viss prestandanivå. Standarden specificerar inte de skyddsfunktioner eller prestandanivåer som ska användas i ett enskilt fall. För att ta reda på vad som krävs av en skyddsfunktion på en viss maskin måste en riskanalys genomföras.

Standarden är enbart inriktad på personsäkerhet och utelämnar skador på egendom och miljö. Standarden avser heller inte att täcka informationssäkerhetsaspekter.

Både EN ISO 13849-1 och EN 62061 behandlar maskinstyrningar och det anses mindre lämpligt att det finns två olika standarder inom EU för samma tillämpningsområde. En samverkan mellan standarderna diskuteras.

2.8 EN 62061 (Maskinstyrningar)

Standard EN 62061 *Maskinsäkerhet – Funktionssäkerhet hos elektriska, elektroniska och programmerbara elektroniska säkerhetskritiska styrsystem* ger säkerhetskrav och vägledning om konstruktionsprinciper och integration av säkerhetsfunktioner för maskiner. Standarden bygger på IEC 61508 och är en sektorspecifik tillämpning för maskinindustrin, se *avsnitt 2.7*.

Standarden har utarbetats av IEC TC 44, *Safety of Machinery – Electrotechnical Aspects*, som är en teknisk kommitté inom den internationella elektrotekniska standardiserings-organisationen IEC. Standarden har övertagits som EN IEC 62061 av den europeiska standardiseringsorganisationen CENELEC och som svensk standard av *Svenska Elektriska Kommittén* (SEK).

2.8.1 Innehåll och omfattning

Standarden beskriver funktionssäkerhet hos elektriska, elektroniska och programmerbara elektroniska säkerhetskritiska styrsystem (*Safety-Related Electrical Control Systems*, SRECS). Riskbedömningen resulterar i en strategi för riskreduktion där de säkerhetsrelaterade funktionerna (*Safety-Related Control Functions*, SRCF) identifieras. Funktionerna dokumenteras med en funktionell kravspecifikation och en kravspecifikation för säkerhetsintegritet.

Metodik och krav ges för att tilldela erforderlig *Safety Intergrity Level* (SIL) till varje säkerhetsrelaterad styrfunktion som sköts av SRECS. Standarden ger stöd vid specifikation, konstruktion och validering. Även integration av skyddsfunktioner utvecklade enligt EN ISO 13849-1 stöds av standarden EN 62061.

Beroende på resultatet av riskanalysen utformas *Safety-Related Control Functions* (SRCF) med olika SIL-nivåer, SIL 1-3, där SIL-3 ställer de högsta kraven. I standard IEC 61508 finnas också SIL 4 definierad, men denna höga grad av nivå bedöms inte vara aktuell för maskiner. Alla maskinstyrningar förutsätts vara i *high demand* eller i *continuous mode*. Sannolikhetsvärdena för SIL 1-3 är desamma som i standard IEC 61508.

I en informativ del av standarden visas ett tillvägagångssätt för att bedöma SIL-nivån. Allvarlighetsgraden (*severity*) bedöms på en skala från 1 till 4 där 4 motsvarar de mest allvarliga konsekvenserna. På liknande sätt läggs exponering, sannolikhet och möjlighet att undvika faran samman till en klass (*class*) där ett högt värde motsvarar hög sannolikhet. Vid hög allvarlighetsgrad och hög sannolikhet väljs SIL 3, medan för lägre olycksrisker kan normala kvalitetsstödjande åtgärder (*Other Measures*, OM) vara tillräckliga. *Safety-Related Control Functions* (SRCF) med försumbara olycksrisker tilldelas ingen SIL-nivå.

Genom att betrakta tåligheten mot hårdvarufel (*Hardware Fault Tolerance*, HFT) och felsäkerhetskvot (*Safe Failure Fraction*, SFF) kan standarden beskriva den maximala SIL-nivån som kan uppnås med en viss konstruktionsarkitektur. Kraven på tålighet mot hårdvarufel ökar med högre SIL-nivå medan standarden tillåter att en-kanaliga konstruktioner används upp till SIL 3, förutsatt att möjligheten till felupptäckt är tillräckligt hög.

En kravspecifikation för säkerhet i programvara (*Software Safety Requirements Specification*, SSRS) ska tas fram för varje delsystem.

För konstruktion och utveckling av inbyggd programvara hänvisas till IEC 61508-3. Däremot tar standarden upp parametrisering och utveckling av tillämpningsprogramvara eftersom dessa kan förväntas vara aktiviteter som utförs av många maskinbyggare. Ofta bygger maskinstyrningen på ett modulärt styrsystem (*Programmable Logic Controller*, PLC) där konstruktören matar in parametrar och programvara för att styra maskinen.

För parametrisering föreskrivs att integriteten i data ska bibehållas bland annat genom att undersöka att dessa ligger inom giltigt område och att de inte förvanskas. Krav ställs också på verktyget som används för parametrisering och på tillvägagångssättet för att påverka de säkerhetskritiska parametrarna.

Vid utveckling av applikationsprogramvara föreskrivs att IEC 61508-3 ska följas vid användning av FVL (*Full Variability Language*). Exempel på FVL är programmering i högnivåspråk som C eller Ada. Om maskinstyrningen programmeras i ett LVL (*Low Variability Language*) ställer EN 62061 ett antal krav på utveck-

lingsprocess, konfigurationsledning, programvaruarkitektur, verktyg, utvecklingsmetodik och testning. Exempel på LVL är programmering i funktionsblock för PLC.

I de fall tillämpningsprogramvaran styr både säkerhetsrelaterade och icke säkerhetsrelaterade funktioner ska hela tillämpningsprogramvaran betraktas som säkerhetsrelaterad förutsatt att inte tillräckligt oberoende mellan programdelarna kan visas.

2.8.2 Tillämpningsområde

Grundläggande hälso- och säkerhetskrav för maskiner inom EG ges genom det europeiska maskindirektivet. För att få detaljerad information om säkerhetsaspekter hänvisas till standarder. Standarden EN 62061 är avsedd att ge vägledning till dem som arbetar med konstruktion och bedömning av styrsystem för maskiner. Den ger ingen specifik vägledning för överensstämmelse med andra EU-direktiv.

Komponenter som gränslägesbrytare och programmerbara styrsystem (PLC), vilka kan intygas av tillverkaren, kan användas i säkerhetsfunktioner med en viss SIL-nivå. Standarden specificerar inte de skyddsfunktioner eller SIL-nivåer som ska användas i ett enskilt fall. För att ta reda på vad som krävs av en viss skyddsfunktion på en viss maskin måste en riskanalys genomföras.

Standarden är inriktad enbart på personsäkerhet och utelämnar skador på egendom och miljö. Standarden avser heller inte att täcka informationssäkerhetsaspekter.

En diskussion pågår hur en samordning ska ske mellan EN 62061 och EN ISO 13849-1. Se *avsnitt 2.7.2*.

2.9 RTCA DO-178C/EUROCAE ED-12C (Flyg)

RTCA DO-178C *Software Considerations in Airborne Systems and Equipment Certification* är en internationellt etablerad standard som fokuserar på programvara inom flygtillämpningar. Den gällande versionen av standarden är C och den gavs ut 2011. Standarden omfattar programvarans hela livscykel och berör till viss del hårdvara men bara i dess relation till programvara. Det finns till exempel ingen processbeskrivning för hårdvara.

Standarden är framtagen i samarbete mellan RTCA (*Radio Technical Commission for Aeronautics*) Special Committee 205 (SC-205) och EUROCAE Working Group 71 (*European Organization for Civil Aviation Equipment WG-71*).

2.9.1 Innehåll och omfattning

En aspekt som genomsyrar standarden är civil certifiering. Standarden är konsekvent utarbetad och kan betraktas som komplett med avseende på den omfattning standarden har. Standarden innehåller även en hel del vägledning, tillvägagångssätt, exempel, definitioner och förklaringar. Standarden innehåller även två ANNEX och två APPENDIX.

En speciell aspekt är att det inte finns några *shall* (skall) eller *must* (måste) i standarden utan bara should. Anledningen är att det inte finns något lagkrav på att följa standarden. Dokumentet bygger på konsensus inom flygbranschen, men erkänner samtidigt att det kan finnas alternativa metoder. Detta är anledningen att orden *shall* och *must* undviks i texten. Däremot används ordet *may* (får, kan) genomgående. Hur standarden ska tillämpas bestäms i PSAC. PSAC överenskoms mellan industri och FMV eller mellan industri och certifierande myndighet om industrin ska leverera en certifierad produkt enligt tillämpligt regelverk för luftfart.

Standarden är fokuserad på vad som ska göras men inte hur. Det innebär att granskningen, exempelvis vid certifiering, måste ta noggrann ställning till om rätt avgränsning har gjorts och om innehållet stämmer överens med de använda delarna av standarden.

I och med att i princip alla delar av kravtexten (rubriker, syften, listor, definitioner, exempel mm) har identitet underlättas referenser, exempelvis kan man i ANNEX A relatera till referenser och dokumentation till Software Level.

Standarden är uppbyggd i olika sektioner. Notera att *Integral Processes* utförs parallellt med *Software Planning Process* och *Software Development Processes* under hela livscykeln. Standarden beskriver programvaruprocessens relation till systemprocesser och det finns ett stort informationsutbyte mellan dessa olika processer.

De dokument som kan ingå anges i *tabell 2:5*. I standardens Section 11: *Software Life Cycle Data* finns beskrivning av innehållet.

Tabell 2:5 Exempel på dokumentation angivet i RTCA DO-178C

Dokument	Skapad/uppdaterad i process
Design Description	Software Design Process
Executable Object Code	Integration Process Software Development Process
Parameter Data Item File	Integration Process Software Development Process
Plan For Software Aspects Of Certification	Software Planning Process Certification Liason Process
Problem Reports	Software Configuration Management Process
Software Accomplishment Summary	Certification Liaison Process
Software Code Standards	Software Planning Process
Software Configuration Index (SCI)	Software Configuration Management Process Certification Liaison Process
Software Configuration Management Plan	Software Planning Process
Software Configuration Management Records	Software Configuration Management Process
Software Design Standards	Software Planning Process
Software Development Plan	Software Planning Process
Software Life Cycle Environment Configuration Index (SECI)	Software Configuration Management Process

Dokument	Skapad/uppdaterad i process
Software Quality Assurance Plan	Software Planning Process
Software Quality Assurance Records	Software Quality Assurance Process
Software Requirements Data	Software Requirement Process
Software Requirements Standards	Software Planning Process
Software Verification Cases And Procedures	Software Verification Process
Software Verification Plan	Software Planning Process
Software Verification Results	Software Verification Process
Source Code	Software Coding Process Software Development Process
Trace Data	Software Development Processes Software Verification Process

Det finns fem definierade nivåer av Software Level (SL) med avseende på allvarlighet för programvaran. Se standarden RTCA DO-178C för fullständiga definitioner:

- Level A: ”…resulting in a catastrophic failure…”
- Level B: ”…resulting in a hazardous failure…”
- Level C: ”…resulting in a major failure…”
- Level D: ”…resulting in a minor failure…”
- Level E: ”…no effect on aircraft operational capability or pilot workload…”.

Standarden definierar kritikalitetsnivåer utifrån konsekvens. Det medtagna exemplet för en systemsäkerhetsprocess anger flygplanskrasch med många döda (A), fåtal personer (passagerare) skadas eller avlider (B), avsevärd minskning av säkerhetsmarginaler eller funktionalitet (C), viss minskning av säkerhetsmarginaler eller funktionalitet (D), ingen effekt med avseende på allvarlighet (E).

Ekonomisk skada och skada på miljön ingår inte. Det finns inget stöd för hur val av nivå ska gå till utan hänvisning görs istället till systemprocesser.

För programvaror som ingår i ett tekniskt system bestäms kritikalitetsnivån utifrån en riskanalys genomförd exempelvis enligt standarden SAE-ARP 4754A, se *avsnitt 2.11*.

I ANNEX A anges vilka aktiviteter och vilket resultat som krävs i relation till Software Level (SL).

Kravnedbrytning görs enligt följande:

1. Säkerhetsrelaterade krav, inkluderande *Software Level* (SL) för programvara, tas fram på systemnivå (resultat från *System Safety Assessment Process*).
2. Krav bryts ner till högnivåkrav för programvara beroende på *Software Level* (SL).
3. Krav bryts ner till successivt lägre nivåer (om tillämpbart) för programvara.
4. Lågnivå krav (lägsta nivån) definieras för programvara.
5. Härledda krav ska identifieras på både hög och låg nivå.
6. Kravspårning beroende på *Software Level* (SL).

På PDS (benämnd COTS i standarden) ställs samma krav som på egenutvecklad programvara, se DO-278A.

2.9.2 Tillämpning

Standarden är framtagen för civil luftfart men kan även tillämpas för militär luftfart. Standarden gäller programvaruprodukter inom flygburna system, vilket visar sig i definitionen av *Software Level* (SL).

Kopplingen till systemprocesser är liten i standarden och visar bara de som är direkt relaterade till programvaruprocesser. Hårdvaruprocesser berörs inte alls. Därför behövs kompletterande hantering av systemprocesser och hårdvaruprocesser, men det finns ingen vägledning för att välja dessa i standarden.

Trots de specifika hänvisningarna till flygburna system kan standarden i princip användas för andra typer av tekniska system, men behöver då omtolkas och hanteras speciellt. Man får dock tydligt specificera varför man väljer en sådan standard, vilka delar som ingår och tolkning av innehållet. För att ha en komplett hantering måste även systemprocesser och hårdvaruprocesser tas med.

Standarden täcker inte informationssäkerhetsaspekter.

Som stöd för tillämpning för DO-178C och DO-278 kan DO-258C användas.

Det finns fyra ytterligare dokument som tar upp speciella aspekter inom programvaruutveckling. Se referenser nedan:

- RTCA DO-330 *Software Tool Qualification Considerations* utvecklades för att ge vägledning hur verktyg som används vid utvecklingen kan kvalificeras.
- RTCA DO-331 *Model-Based Development and Verification Supplement to DO-178C and DO-278* beskriver modellbaserad utveckling och verifiering.
- RTCA DO-332 *Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A* tar upp objektorienterad programvara och under vilka förutsättningar den kan användas.
- RTCA DO-333 *Formal Methods Supplement to DO-178C and DO-278A* beskriver formella metoder och hur de kan komplettera testning.

För certifiering av flygsystem finns även RTCA DO-297 *Integrated Modular Avionics* (IMA) Design Guidance and Certification Considerations.

Råd för tillämpning av DO-178C och DO-278A ges i DO-248C.

En tillämpningsguide/anpassning av DO-178C för flygtrafikledningssystem finns i RTCA DO-278A *Guidelines for Communication, Navigation, Surveillance, and Air Traffic and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance*.

2.10 RTCA DO-254 (Programmerbar logik, flyg)

Standarden RTCA DO-254 *Design Assurance Guidance for Airborne Electronic Hardware* specificerar riktlinjer för konstruktion av elektronikhårdvara för flygindustrin.

Standarden är framtagen i samarbete mellan RTCA (*Radio Technical Commission for Aeronautics*) Special Committee 180 (SC-180) och EUROCAE Working Group 46 (*European Organization for Civil Aviation Equipment WG-46*).

2.10.1 Innehåll och omfattning

Syftet med standarden är att försäkra sig om att avsedda funktioner utförs på ett säkert sätt. Även om standarden avser hårdvara bör den beaktas när man diskuterar programvara, eftersom den täcker programmerbara kretsar. Skillnaden mellan logik i form av programvara för en dator och logik i form av innehållet i en programmerbar krets kan tyckas liten.

Standarden definierar fem olika nivåer, *Design Assurance Level* (DAL), för att försäkra sig om en korrekt utveckling av systemet respektive hårdvaran, DAL A-E. Dessa nivåer baseras på konsekvenserna av fel som kan orsaka olyckor. För att utveckla elektronikhårdvara motsvarande DAL A krävs mycket mer omfattande verifiering och validering än för DAL E.

Konstruktionsarbetet börjar på systemnivå genom att fördela olika funktioner till hårdvara och tilldela hårdvaran DAL-nivå för att försäkra sig om korrekt utveckling. En funktion i systemet kan fördelas till en hårdvaruenhet, en programvarukomponent eller till en kombination av både hårdvara och programvara.

Standarden pekar på tre utvärderingsprocesser för systemsäkerhet:

- Functional Hazard Assessment (FHA)
- Preliminary System Safety Assessment (PSSA)
- System Safety Assessment (SSA).

En livscykel för utvecklingsarbetet beskrivs med fem huvudprocesser som kan följas för kompletta hårdvaruenheter, kretskort eller ASIC/PLD (*Application Specific Integrated Circuits/Programmable Logic Devices*). Huvudprocesserna är kravställning, konceptkonstruktion, detaljkonstruktion, implementering och överförande till produktion.

För att konstruera elektronikhårdvara är konstruktören beroende av verktyg. Detta gäller i extra hög grad för ASIC och PLD. Ett fel i konstruktionen kan mycket väl introduceras av felaktigt fungerande konstruktionsverktyg. På liknande sätt kan ett felaktigt fungerande testverktyg undgå att hitta fel som finns i kon-

struktionen. Därför föreskriver standarden att verktyg bör utvärderas innan användning och att resultatet av verktygskvalificeringen ska dokumenteras och sparas.

Om resultatet från verktygskvalificeringen ska genomgå en oberoende utvärdering behöver inte verktyget i sig självt utvärderas. För de lägre nivåerna bedöms även att utvärdering av verktyget inte krävs. Däremot behöver konstruktionsverktyg för DAL A, B och C samt testverktyg för DAL A och B, utvärderas. Enda undantaget är om man kan visa att det finns en relevant historik från tidigare användning av verktyget.

Formella metoder beskrivs som en teknik som kan ge ytterligare bevisning i konstruktionsprocessen (se *RTCA/DO-254, appendix B, avsnitt 3.3.3*). För att kunna använda formella metoder behöver kravspecifikationen skrivas formellt. Detaljeringsgraden i den formella beskrivningen av en komponent beror på målen med de valda formella analysmetoderna.

En annan aspekt är att det inte finns några *shall* och i princip inga *must* i standarden utan bara *should*. Anledningen är att det inte finns något lagkrav på att följa standarden. Dokumentet bygger på konsensus inom flygbranschen men möjliggör samtidigt alternativa metoder. Dessutom används *may* genomgående i standarden.

2.10.2 Tillämpning

Standarden RTCA/DO-254 behandlar elektronikhårdvara för användning inom flygindustrin. Standarden är tillämpbar på, men inte begränsad till:

- utbytbara moduler (*Line Replaceable Unit*, LRU)
- kretskort
- mikrokodade komponenter som ASICs och PLDs
- komponenter med teknologi-integrerade kretsar, till exempel multichip-moduler.

Standarden avser inte att täcka informationssäkerhetsaspekter.

2.11 ARP 4754A (Flyg)

Standarden SAE ARP4754A *Aerospace Recommended Practice - Guidelines for Development of Civil Aircraft and Systems* gäller för systemaspekter och refererar till DO-178C/ED-12C för utveckling av programvara och DO-254/ED-80 för utveckling av hårdvara.

Standarden är framtagen i samarbete mellan SIRT (*Systems Integration Requirements Task*) och EUROCAE Working Group 46 (*European Organization for Civil Aviation Equipment WG-46*).

2.11.1 Innehåll och omfattning

Standarden innehåller tre APPENDIX (*ett fjärde appendix har utgått*):

- APPENDIX A; Process objectives data
- APPENDIX B; Safety program
- APPENDIX C; FDAL/IDAL Assignment process example.

Processerna i standarden delas in i tre huvudgrupper:

- Aircraft and system development process
- Integral processes
- Modifications to aircraft or systems.

I och med att i princip alla delar av kravtexten såsom rubriker, syften, listor, definitioner, exempel och checklistor har identitet underlättas referenser. Exempelvis kan man i APPENDIX A relatera referenser och dokumentation till *Development assurance level*.

En aspekt som genomsyrar standarden är certifiering. Standarden är konsekvent och kan betraktas som komplett med avseende på den omfattning den har. Standarden innehåller även vägledning, tillvägagångssätt, exempel, checklistor, definitioner och förklaringar.

En annan aspekt är att det inte finns några *shall* och i princip inga *must* i standarden utan bara *should*. Anledningen är att det inte finns något lagkrav på att följa standarden. Dokumentet bygger på konsensus inom flygbranschen men möjliggör samtidigt alternativa metoder. Dessutom används *may* genomgående i standarden.

Standarden är starkt fokuserad på *vad* som ska göras, men också i många fall i form av exempel, *hur*. Det innebär att granskningen, exempelvis vid certifiering, måste ta noggrann ställning till om rätt omfattning har tagits med och om innehållet stämmer överens med de använda delarna av standarden.

Standarden är övergripande och därmed blir terminologin extra viktig inte minst på grund av att nya begrepp ingår och andra begrepp har en annan definition jämfört med andra standarder.

Classification matchar *Assurance level* enligt *Catastrophic – A, … No Safety Effect – E* (se Table 2). Notera att *fault*, *error* och *failure* skiljer sig från Laprie:s definitioner (vilka används frekvent i den akademiska världen). Nivå A – E används också för informationsutbyte enligt DO-178B/ED-12B och DO-254/ED-80.

Standarden lägger stor vikt vid krav och föreslår tidig validering av krav även om valideringen kan behöva göras om då design och implementation genomförts. Att lägga stor vikt vid kravformulering motiveras som mycket kostnadseffektivt. Stor vikt läggs också på att definiera funktioner, att analysera gemensamma felorsaker (*Common Cause Failure*, CCF) och att hantera härledda krav (*Derived Requirements*), det vill säga krav som tillkommer på lägre nivåer och som inte direkt kan spåras till krav på högre systemnivåer.

En tydlig redogörelse finns för utveckling av programvara och hårdvara där också informations-utbytet finns beskrivet mellan systemprocesser och programvaru-/hårdvaruprocesser.

Det som är speciellt med denna standard är tilldelning av FDAL till funktioner (*functions*) och IDAL till komponenter (*items*). Det innebär att samma hantering kan göras, exempelvis jämförs oberoende mellan funktioner och oberoende mellan komponenter. Standarden tar upp ett exempel som visar på metoden för tilldelning av FDAL och IDAL. Det är en applikation där inga beroen-

den finns och som är strukturerad i form av ett felträd, där bashändelse är felorsak och topphändelsen är ett katastrofalt fel. För att konsekvensen av en topphändelse ska mildras, så måste tilldelning av FDAL/IDAL komma från båda grenarna under topphändelsen och lista de felkombinationer som är relevanta. Det vill säga de kortaste vägarna till att topphändelse ska inträffa, så kallade *Minimal Cut Set* (MCS). Det gäller då att bestämma FDAL för funktionerna och IDAL för komponenterna.

För verifiering och validering av krav anges i standarden metoder och data beroende på *Development Assurance Level A-E*, (DAL A-E). För både verifiering och validering finns möjlighet till anpassning för certifiering (R - *Recommended for Certification*, A - *As Negotiated for Certification*, N - *Not Required for Certification*). Därmed finns även kvalitetskontroll på systemnivå. Det bör dock noteras att både verifiering och validering görs på systemnivå.

2.11.2 Tillämpning

Standarden har ingen specifik civil eller militär aspekt. Den gäller systemdelarna inom flygburna system och har en tydlig koppling till dessa i texten och som visar sig i definitionen av *Software Level* (SL), certifiering och många andra ställen där det anges *aircraft* och *airborne*. Det finns därmed ingen koppling till sjö eller mark i standarden.

Programvaruprocesser och hårdvaruprocesser berörs inte utan hanteras i associerade standarder. Dessa är så pass starkt kopplade till *SAE ARP4754A* att de inte är lämpliga att ersättas av andra standarder.

Standarden täcker inte informationssäkerhetsaspekter.

Trots de specifika hänvisningarna till flygburna system så kan standarden i princip användas för andra typer av applikationer. Det krävs dock att en tydlig motivering för valet av denna standard samt specificering av vilka delar som ingår och tolkning av innehållet. Ett exempel är *Classification* som då kan behöva göras om helt.

Standarden kan inte användas separat utan måste användas tillsammans med associerade standarder för utveckling av programvara eller hårdvara. Eftersom mycket information finns och krävs bör användare ha ordentlig utbildning innan standarden tillämpas. Ett kapitel som kräver djupare genomgång på grund av dess komplexitet och betydelse är *avsnitt 5.2 Development Assurance Level Assignment*.

2.12 EN 50128:2011 (Järnväg)

Det finns tre internationella standarder för järnvägsanläggningar som tillsammans bildar en enhet:

- Systemaspekter anges i CENELEC, *Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety* (RAMS), EN 50126.
- Programvaruaspekter anges i CENELEC, *Railway Applications - Communication, Signaling and Processing Systems - Software for Railway Control and Protection Systems*, EN 50128.
- Hårdvaruaspekter anges i CENELEC, *Railway Applications – Communication, Signaling and Processing Systems – Safety Related Electronic Systems for Signaling* EN 50129.

2.12.1 Innehåll och omfattning

Här sammanfattas EN 50128 och den aktuella versionen är från juni 2011 och det är inte helt säkert när en ny officiell version förväntas komma. De tre standarderna ska ses som en specialisering av den generiska standarden IEC 61508 och skapad för sektorn järnvägsanläggningar. Applikationen är järnvägsanläggningar men det finns inget som hindrar att man tillämpar standarden på andra områden efter viss anpassning. EN 50128 adresserar alltså säkerhetsfunktioner i programvara och består av följande delar:

Chapter 1 – Scope

Chapter 2 – Normative References

Chapter 3 – Terms, Definitions and Abbreviations

Chapter 4 – Objectives, Conformance and Software Safety Integrity Levels

Chapter 5 – Software Management and Organization

Chapter 6 – Software Assurance

Chapter 7 – Generic Software Development

Chapter 8 – Development of Application Data or Algorithms: Systems Configured by Application Data or Algorithms

Chapter 9 – Software Deployment and Maintenance

Annex A – Criteria for the Selection of Techniques and Measures

Annex B – Key Software Roles and Responsibilities

Annex C – Documents Control Summary

Annex D – Bibliography of Techniques

Två exempel på livscykelmodeller för framtagning av programvara visas i denna standard varav en är V-modellen. För bägge anges relaterade dokument och dessa finns också angivna i Annex C.

Terminologi finns i alla tre delarna. Notera att security inte är informationssäkerhet utan definieras enligt: ”Security, as an element that characterises the resilience of a railway system to vandalism and unreasonable human behaviour, …”. Security definieras bara och hanteras inte i övrigt i EN 50126, EN 50128 eller EN 50129. Informationssäkerhet adresseras inte.

Riskgrafsmetoden är applikationsberoende men ett rimligt exempel visas nedan. Denna anges i EN 50126. Här kan då ”Risk Levels” ersättas med motsvarande krav på säkerhetsfunktion; till exempel SIL 1 – Negligible, SIL 2 – Tolerable, SIL 3 – Undesirable, SIL 4 – Intolerable.

Frekvens	Risknivåer			
Frekvent	Oönskad	Ej tolerabel	Ej tolerabel	Ej tolerabel
Sannolik	Tolerabel	Oönskad	Ej tolerabel	Ej tolerabel
Tillfällig	Tolerabel	Oönskad	Oönskad	Ej tolerabel
Möjlig	Negligerbar	Tolerabel	Oönskad	Oönskad
Osannolik	Negligerbar	Negligerbar	Tolerabel	Tolerabel
Otrolig	Negligerbar	Negligerbar	Negligerbar	Negligerbar

Bild 2:4 Förekomst i händelse av fara, återgiven från EN 50126

Framtagning av programvara specificeras i *Chapter 7, Generic Software Development*, där följande delar ingår:

1. Lifecycle and Documentation for Generic Software.
2. Software Requirements
3. Architecture and Design
4. Component Design
5. Component Implementation and Testing
6. Integration
7. Overall Software Testing/Final Validation

Ett antal aktiviteter löper parallellt (*Chapter 6*) med själva framtagningen av programvara: *Software Testing, Software Verification, Software Validation, Software Assessment, Software Quality Assurance och Modification and Change Control*.

För verktyg används samma klassificering som IEC 61508.

Chapter 8 adresserar parametrisering, kallad *application data*, och en speciell process används för detta. Verktygsstöd behövs normalt. Tanken är att Generic software tas fram först och är till viss/stor del applikationsoberoende och som sedan görs applika-

tionsspecifik med hjälp av *application data*. Några viktiga saker att beakta är då: utvecklingsprocess för *application data*, SIL för *application data* och otillåtna kombinationer av *application data*.

Borttagning av uttjänt programvara adresseras inte i EN 50128 utan görs för hela systemet enligt EN 50126.

Stort fokus är lagt på roller (se Annex B) och följande roller finns definierade: *Requirements Manager*, *Designer*, *Implementer*, *Tester*, *Verifier*, *Integrator*, *Validator*, *Assessor*, *Project Manager*, *Configuration Manager*. En person/organisation kan ha mer än en roll. Krav ställs på vad respektive rollinnehavare ska göra; framtagning av speciella dokument och genomförande av aktiviteter. Hur de olika rollerna interagerar anges också.

Det finns några få *should* i krav och dessutom några i NOTE. I övrigt används *shall*.

Notera att RAMS egentligen inte berörs i EN 50128 och EN 50129 utan ligger i princip helt i EN 50126.

Annex A: Criteria for the Selection of Techniques and Measures.
SIL-beroende görs på motsvarande sätt som för IEC 61508, det vill säga enbart med hjälp av tabeller, men dessa skiljer sig från IEC 61508 del 3.

Några kommentarer i relation till IEC 61508:

- Det är bra att M (Mandatory) införs, för det finns saker som inte ska kunna undantas eller förhandlas bort.
- B-tabellerna är borttagna och informationen har istället överförts till de nya A-tabellerna.
- SIL 1 och 2 respektive SIL 3 och 4 har sammanförts, vilket innebär ökade krav på SIL 1 respektive SIL 3.
- Backward Recovery och Forward Recovery sätts som NR eftersom dessa är svåra att få bra i praktiken.
- Olika programspråks lämplighet skiljer sig.

Annex B: Key software roles and responsibilities.
De olika rollerna och deras ansvar definieras i Annex B. Dessutom anges vilken kompetens respektive rollinnehavare ska ha.

Annex C: Documents Control Summary.
Dokumentlistan definieras i Annex C och anges också relationer till roller.

Annex D: Bibliography of techniques.
Här beskrivs 71 olika metoder på samma sätt som i IEC 61508 del 7. Man måste inte följa dessa men de utgör en bra start och är oftast tillräckliga.

2.12.2 Tillämpning

Standarden gäller programvara och har ingen speciell civil eller militär aspekt men är avsedd för järnvägsanläggningar. Standarden kommer från CENELEC och är oberoende av användning för mark, sjö och luft men den har stark koppling till IEC61508 och därmed tillämplig för säkerhetsfunktioner. *Hazard* används men ingen koppling till typ av skada (person, utrustning, ekonomi, miljö) görs. Safety används också och gäller då skada på person. Standarden avser inte att täcka informationssäkerhetsaspekter.

2.13 ED-153 (Flygtrafikledningstjänst

EUROCAE, ED-153 *Guidelines for ANS Software Safety Assurance*, är en internationell standard avsedd för programvara inom flygtrafikledningstjänst (*Air Navigation Service*, ANS). Den aktuella versionen är från augusti 2009. Standarden gäller generellt för applikationer, det vill säga inte speciellt för exempelvis säkerhetsfunktioner (till skillnad mot IEC 61508).

2.13.1 Innehåll och omfattning

Standarden inkluderar också hantering av infrastruktur och projekt och består av följande delar:

Chapter 1 – Introduction

Chapter 2 – Document Strategy

Chapter 3 – Software Safety Assurance System

Chapter 4 – Primary Lifecycle Processes

Chapter 5 – Supporting Lifecycle Processes

Chapter 6 – Organisational Lifecycle Processes

Chapter 7 – Additional ANS Software Lifecycle Objectives

Chapter 8 – Software Safety Folder

Annex A – Reference to Existing Software Standards

Annex B – Roles and Responsibilities Scenarios

Annex C – Traceability with ESARR6

Standarden tillämpas för flygtrafikledning, men det finns inget som hindrar att man tillämpar den med viss anpassning inom andra områden. I *Chapter 8 Software Safety Folder* anges dokument och bevis för SWAL (motsvarande *Safety Case* i andra sammanhang). Två olika upplägg beskrivs: ”Projektbaserad struktur” (*Project-based Structure*) och ”Överrensstämmelsebaserad struktur” (*Compliance-based Structure*).

I Annex A görs en jämförelse mellan olika standarder: ISO/IEC 12207, ED-109/DO-278, ED-12B/DO-178B, IEC 61508, CMMI. I Annex B visas olika exempel på roller och ansvar. Annex C ger spårbarhet mellan ED-153 och ESARR6 (*Software in ATM Functional Systems*). Terminologin framgår av *tabell 2:6*.

Tabell 2:6 Terminologi inom standarden ED-153

Term	Förklaring
ANS	Air Navigation Service
COTS	Commercial Off The Shelf. COTS betecknar här inköpt SW, tidigare utvecklad SW med hjälp av ED-153 etc. PDS (Previously Developed Software) används också som beteckning för COTS.
Independence (oberoende)	Behöver hanteras av: annan person, olika avdelningar inom företag, olika organisationer etc.
SWAL	Software Assurance Level, 1 – 4 finns där 1 är den med högsta kraven (mest kritiska) och 4 den med lägsta kraven.
ESARR	Eurocontrol Safety Regulatory Requirement

Ett relativt stort antal processer ingår i standarden och det finns en inledande beskrivning av respektive process. Varje process innehåller ett antal *objectives* (mål/målsättningar) som ska ses som krav. Varje krav anges som ”shall” med numrering/identitiet. Det finns andra exempel, *should* och *note*, men dessa betraktas inte som krav och har därmed ingen numrering. Till varje *objective* finns angivet hur tillämpbart det är med avseende på SWAL (1 – 4) och dessutom om det krävs arbete av oberoende part (enligt någon nivå). Utdata anges också för varje mål. Vissa processer är inte fokuserade/applicerbara på programvaruutveckling utan istället på systemaspekter bland annat upphandling, leverans, validering och drift och underhåll. Även *Organisational Lifecycle Processes* kan vara relativt programvaruoberoende. Standarden har en ganska utförlig beskrivning av och kravställning på COTS (många ”should” och ”may”). COTS får dock inte användas för SWAL1 (se *chapter 7.2.0 NOTE*). Råd ges även hur kvalificering av verktyg bör hanteras.

För risker används en *cause-effect*-princip där man till exempel använder FTA (Felträdanalys, Fault Tree Analysis) för ”cause” (som ger ”hazard”) och till exempel ETA (Händelseträdsanalys, Event Tree Analysis) för ”effect” (som ger effekterna från ”hazard”). *Tabell 2:7* nedan visar hur SWAL väljs utgående från sannolikhet och allvarlighetsgrad.

Tabell 2:7 Val av SWAL baserat på sannolikhet och allvarlighet

Likelihood of generating such an effect (Pe × Ph)	Effect Severity Class			
	1	2	3	4
Very Possible	SWAL 1	SWAL 2	SWAL 3	SWAL 4
Possible	SWAL 2	SWAL 3	SWAL 3	SWAL 4
Very Unlikely	SWAL 3	SWAL 3	SWAL 4	SWAL 4
Extremely Unlikely	SWAL 4	SWAL 4	SWAL 4	SWAL 4

Effect Severity Class 1 – 4 finns inte definierade utan definieras av applikationen. Detta innebär att det är svårt att jämföra risker med andra standarder. *Severity Class 1* är den mest allvarliga. För sannolikhet (*Likelihood*) finns följande definierade: *Very Possible*, *Possible*, *Very Unlikely*, *Extremely Unlikely*. Risk är inte specifikt definierad men ska ses som en kombination av *Severity Class* och *Likelihood*.

2.13.2 Tillämpning

Standarden tillämpas för programvaruutveckling och har ingen speciell civil eller militär aspekt, men är avsedd för flygtrafikledningstillämpningar (ANS). Bakgrunden är dock civil, standarden kommer från EUROCAE (*The European Organisation for Civil Aviation Equipment*). Ingen speciell aspekt finns med avseende på skador på människa, ekonomi, utrustning eller miljö och ingen heller kopplad till mark och sjö. Det finns fyra *objectives* som adresserar *security*. Även om inte explicit definierat så antas att *security* gäller informationssäkerhet.

2.14 IEC 61511 (Processindustri)

Standard IEC 61511 *Functional Safety – Safety Instrumented Systems for the Process Industry Sector* är en internationellt etablerad standard som består av tre delar och utkom första gången 2003. Den nu gällande versionen utkom 2016. Standarden berör hela livscykeln.

2.14.1 Innehåll och omfattning

Standardens första del är normativ medan de övriga två är informativa:

- Del 1: Framework, definitions, system, hardware and application programming requirements (normativ).
- Del 2: Guidelines for the application of IEC 61511-1 (informativ).
- Del 3: Guidance for the determination of the required safety integrity levels (informativ).

Begreppet *Safety Instrumented System* (SIS) används för att beteckna säkerhetsrelaterade styrsystem. Ett SIS kan implementera flera säkerhetsfunktioner *Safety Instrumented Function* (SIF). Avsikten är att IEC 61511 ska vara till nytta för den som bygger ett SIS genom att koppla samman flera komponenter. Inom processindustrin är det vanligt att man köper färdiga styrsystem (PLC), färdiga givare och färdiga ställdon för att själv ansvara för funktionalitet och programvara. I detta fall kan IEC 61508 användas av komponenttillverkarna och IEC 61511 användas av den som bygger anläggningen.

Standarden innehåller ett antal tekniska krav vilka samtliga återfinns i del 1. Vissa avsnitt i del 1 samt hela del 2 och 3 innehåller stöd vid tillämpning av standarden.

Livscykeln bygger på den livscykel som finns i standard IEC 61508.

Kraven på integritet (riskreduktion) hos säkerhetsfunktionerna bedöms enligt SIL 1-4 (*Safety Integrity Level 1-4*) där SIL 4 ställer de högsta kraven. Både hårdvara och programvara ingår. Standarden är inte avsedd för användning om kraven på funktionssäkerhet inte matchar någon nivå SIL 1-4.

Standarden skiljer mellan applikationsprogramvara och inbyggd programvara. Inbyggd programvara levereras av tillverkaren och är inte tillgänglig för ändring av användaren. Applikationsprogramvara är specifik för tillämpningen och man skiljer på tre olika sorter:

- Fixed program language, FPL: endast möjligt att ändra med hjälp av parametrar.
- Limited variability language, LVL: programspråk för industriella styrsystem med begränsningar i vilka funktioner som kan programmeras.
- Full variability language, FVL: ett generellt programspråk med möjlighet att skapa önskvärda funktioner och tillämpningar. För applikationer skrivna i FVL görs i stället referens till IEC 61508-3:2010.

IEC 61511-1 innehåller krav för applikationsprogrammering. Bland annat kan man hitta information om:

- Livscykeln för applikationsprogrammering (avsnitt 6.3).
- Utveckling av applikationsprogram (avsnitt 12).

IEC 61511-2 innehåller vägledning för tillämpning av standarden. Här ges bland annat stöd för applikationsprogrammering, bland annat kan man hitta information om:

- Livscykeln för applikationsprogrammering (avsnitt A.6).
- Utveckling av applikationsprogram (avsnitt A.12).
- Exempel på utveckling med funktionsblock (appendix B).
- Metoder och verktyg för applikationsprogrammering (appendix E).
- Exempel på reläschemaprogrammering (appendix F).
- Arbetssätt för applikationsprogrammering (appendix G).

2.14.2 Tillämpning

Standard IEC 61511 är en sektorspecifik tillämpning av IEC 61508 främst avsedd för processindustrin.

Inom processindustrin resonerar man ofta i termer av skydd utformat i flera lager eller barriärer. Processen som kan orsaka en riskfylld händelse ska skyddas på flera olika sätt. Styrsystemet är bara ett av dessa lager. Mekaniska skydd, säkerhetskritiska styrsystem, varning och evakuering är andra åtgärder som kan tillgripas för att erhålla en tolerabel risk.

2.15 MIL-STD 882E SYSTEM SAFETY

Försvarsmaktens och FMV:s systemsäkerhetsmetodik grundar sig på USA:s Department of Defence (DoD) militära standard *MIL-STD 882C SYSTEM SAFETY*. Systemsäkerhetmetodiken finns beskriven i Försvarsmaktens Handbok Systemsäkerhet (H SystSäk). H SystSäk 2011 avsnitt 4.4 och Appendix B i MIL-STD 882E ersätts i huvudsak av denna handbok (H ProgSäk).

MIL-STD 882E beskriver hur kritikalitetsklassificering av programvara ska ske som en del i det totala systemsäkerhetsarbetet. Utgående från riskklassificering på systemnivå beskrivs en metodik där programvaran kritikalitetsklassificeras (Software Criticality Indices, SwCI) utifrån kontrollkategori (Software Control Category, SCC) och allvarlighetsklassificering (Severity Category, SC). SwCI blir då ett systemsäkerhetskrav för utveckling av programvara.

Software Control Categories		
Level	Name	Description
1	Autonomous (AT)	• Software functionality that exercises autonomous control authority over potentially safety-significant hardware systems, subsystems, or components without the possibility of predetermined safe detection and intervention by a control entity to preclude the occurrence of a mishap or hazard. *(This definition includes complex system/software functionality with multiple subsystems, interacting parallel processors, multiple interfaces, and safety-critical functions that are time critical.)*
2	Semi-Autonomous (SAT)	• Software functionality that exercises control authority over potentially safety-significant hardware systems, subsystems, or components, allowing time for predetermined safe detection and intervention by independent safety mechanisms to mitigate or control the mishap or hazard. *(This definition includes the control of moderately complex system/software functionality, no parallel processing, or few interfaces, but other safety systems/mechanisms can partially mitigate. System and software fault detection and annunciation notifies the control entity of the need for required safety actions.)* • Software item that displays safety-significant information requiring immediate operator entity to execute a predetermined action for mitigation or control over a mishap or hazard. Software exception, failure, fault, or delay will allow, or fail to prevent, mishap occurrence. *(This definition assumes that the safety-critical display information may be time-critical, but the time available does not exceed the time required for adequate control entity response and hazard control.)*
3	Redundant Fault Tolerant (RFT)	• Software functionality that issues commands over safety-significant hardware systems, subsystems, or components, requiring a control entity to complete the command function. The system detection and functional reaction includes redundant, independent fault tolerant mechanisms for each defined hazardous condition. *(This definition assumes that there is adequate fault detection, annunciation, tolerance, and system recovery to prevent the hazard occurrence if software fails, malfunctions, or degrades. There are redundant sources of safety-significant information, and mitigating functionality can respond within any time-critical period.)* • Software that generates information of a safety-critical nature used to make critical decisions. The system includes several redundant, independent fault tolerant mechanisms for each hazardous condition, detection and display.
4	Influential)	• Software generates information of a safety-related nature used to make decisions by the operator, but does not require operator action to avoid a mishap.
5	No Safety Impacts (NSI)	• Software functionality that does not possess command or control authority over safety-significant hardware systems, subsystems, or components and does not provide safety-significant Information. Software does not provide safety-significant or time sensitive data or information that requires control entity interaction. Software does not transport or resolve communication of safety-significant or time sensitive data.

Bild 2:5 Kategorier för programvarukontroll

Software Safety Criticality Matrix				
	Severity Category			
Software Control Category	Catastrophic (1)	Critical (2)	Marginal (3)	Negligable (4)
1	SwCI 1	SwCI 1	SwCI 3	SwCI 4
2	SwCI 1	SwCI 2	SwCI 3	SwCI 4
3	SwCI 2	SwCI 3	SwCI 4	SwCI 4
4	SwCI 3	SwCI 4	SwCI 4	SwCI 4
5	SwCI 5	SwCI 5	SwCI 5	SwCI 5

Bild 2:6 Programvarusäkerhetskritisk matris

Den bedömda kritikalitetsklassen (SwCI) används sedan för att bestämma vilka aktiviteter som ska genomföras vid utvecklingen av programvaran. Vilka aktiviteter (Level of Rigor, LOR) som ska genomföras beror på det tekniska systemets art och ska överenskommas mellan utvecklande industri och FMV.

SwCI	Level of Rigor Tasks
SwCI 1	Program shall perform analysis of requirements, architecture, design, and code; and conduct in-depth safety-specific testing
SwCI 2	Program shall perform analysis of requirements, architecture, design; and conduct in-depth safety-specific testing
SwCI 3	Program shall perform analysis of requirements, architecture and conduct in-depth safety-specific testing
SwCI 4	Program shall conduct in-depth safety-specific testing
SwCI 5	Once assessed by safety engineering as Not Safety, then no safety specific analysis or verification i required

Bild 2:7 Nivå för val programvarurelaterade aktiviteter

För val av aktiviteter (LOR) och hur aktiviteterna ska genomföras, refereras dels till AOP-52 (Ammunition), dels till *Joint Software Systems Safety Engineering Handbook*. Se *avsnitt 2.16* respektive *2.17*.

2.16 AOP-52 (Ammunition)

Standarden AOP-52 edition 1, *Guidance on Software Safety Design and Assessment of Munition-related Computing Systems*, beskriver hela systemsäkerhetsprocessen och refereras av STANAG 4452 *Safety Assessment Requirements for Munition Related Computing Systems* samt MIL-STD 882E, appendix B.

Standarden är dock inte ett kravuppfyllnadsdokument utan ska ses som en vägledning och rekommendation. Standarden är en *Allied Ordnance Publication* framtagen av Nato Standardisation Agency (NSA). AOP-52 är inte ANSI *approved* eller DoD *adopted*, vilket innebär att Department of Defense (USA) inte använder detta som ett kravdokument. FMV ställer därför inte heller krav på att standarden AOP-52 ska följas av utvecklande industri. För särskild tillämpning se även Handbok Vapen- och ammunitionssäkerhet (H VAS).

Standarden AOP-52 begränsar sig till ammunitionsrelaterad programvara och ska ses som ett komplement till andra standarder såsom MIL-STD 882 och DEF-STAN 00-56. Standarden beskriver den totala verksamheten som ska bedrivas vid framtagning av programvara. Vidare anger den kopplingen mellan systemsäkerhetsprocessen och de unika aktiviteterna som ska bedrivas för programvaruframtagning. Standarden refererar till AOP-15 avseende hur man definierar den acceptabla olycksrisken för ammunitionen.

Centralt i standarden är informationen om hur man definierar kritikalitetsindex, *Software Safety Criticality Index* (SSCI). Kritikalitetsindexet fås genom att i matrisform kombinera olycksdefinitionen enligt AOP-15 edition 3, *Guidance on The Assessment of the Safety and Suitability for Service of Non-nuclear Munitions for Nato Armed Forces* och *Software Control Categories*. Denna anger hur fel i programvaran kan bidra till olyckor. Detta ligger sedan till grund för de aktiviteter och krav som styr framtagningen av programvara avseende utveckling, kodning, provning och integration av programvara i det tekniska systemet.

2.17 Joint Software Systems Safety Engineering Handbook

Handboken *Joint Software Systems Safety Engineering Handbook* har tagits fram i ett samarbete mellan Department of Defence (DoD), NASA, U.S. Army, U.S. Coast Guard, U.S. Navy, U.S. Marine Corps, U.S. Air Force, Missile Defense Agency och amerikansk försvarsindustri.

MIL-STD-882E refererar till handboken för utveckling av programvara i säkerhetskritiska tillämpningar. Handboken beskriver processer och aktiviteter för utveckling av programvara och handbokens processer interagerar med systemsäkerhetsarbetet för det tekniska systemet.

Handboken beskriver att man, under planeringsfasen, ska identifiera konstruktionskrav, definiera processaktiviteter och testaktiviteter, vilka ska inplaneras och genomföras. Handboken redogör för hur utvecklingsprocessen kan anpassas utifrån olika kritikalitetsklassificeringar. För en viss vald kritikalitetsklassificering

beskrivs inte en given uppsättning aktiviteter som ska genomföras. Istället beskriver handboken en stor mängd aktiviteter och krav varur ett antal relevanta aktiviteter bestäms per projekt beroende på det tekniska systemets art.

Val av aktiviteter (LOR) som ska genomföras beror på det tekniska systemets art och ska överenskommas mellan utvecklande industri och FMV.

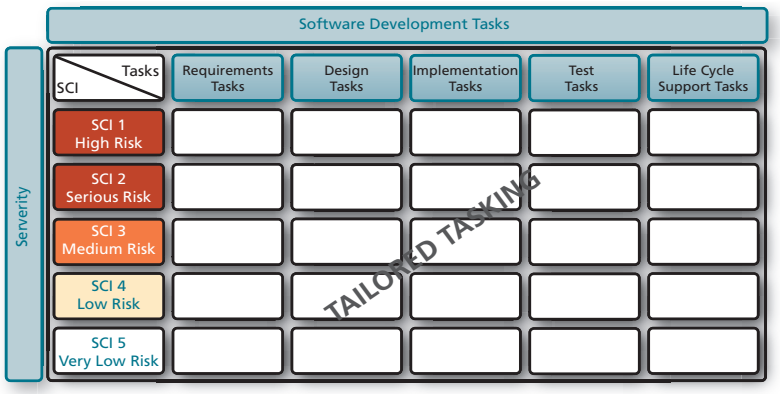


Bild 2:8 Principer för val av utvecklingstekniker beroende på kritikalitet

Design Requirements	Process Tasks	Test Tasks
Fault Tolerant Design	Design Reviews	Safety-Significant Function Testing
Fault Detection	Safety Reviews	Functional Thread Testing
Fault Isolation	Design Walkthroughs	Limited Regression Testing
Fault Annunciation	Code Walkthroughs	100% Regression Testing
Fault Recovery	Independent Reviews	Failure Modes and Effects Testing
Warnings Cautions, and Advisories	Independent Walkthroughs	Safety-CriticalInterfaceTesting
Redundancy	Traceability of Safety-Significant Requirements to Design	COTS, Government Off-the-Shelf Input, Output Test, and Verification
Independence	Traceability of Safety-Significant Requirements to Code	Independent Testing of Prioritized Safety-Related Functions
Functional Partitioning	Traceability of Safety-Significant Requirements to Test	Functional Qualification Testing
Physical Partitioning	Safety Test Results Review	Verification and Validation
Design Safety Standards	Software Quality Assurance Inspections and Audits	Independent Verification and Validation
Design Safety Guidelines	Traceability of Safety-Significant Requirements to Hazards	Full Screening of All COTS Features
Design Safety Lessons Learned	Specific Software Language Requirements	
Full COTS Features Disclosure and Analysis		

Bild 2:9 Exempel på några av de i handboken beskrivna utvecklingskrav och aktiviteter.

2.18 NASA Software Safety Guidebook (NASA-STD-8719.13)

NASA Software Safety Guidebook (NASA-STD-8719.13) togs fram för att ge specifik information och vägledning om processen med att skapa och försäkra sig om att programvara i säkerhetskritiska tillämpningar är tillräckligt säkra.

Handboken vänder sig till en bred målgrupp såsom systemsäkerhetsingenjörer, programutvecklare, kvalitetsingenjörer, projektledare och systemingenjörer. Introduktionen i handbokens inledning ger vägledning om vilka delar i handboken som är av särskilt intresse för de olika målgrupperna.

Handboken är avsedd att vara mer än bara en samling utvecklingsmetoder och analyser. Målet är att öppna för nya sätt att tänka på programvara ur säkerhetssynpunkt. Handboken pekar på saker att leta efter (och se upp för) i utvecklingen av säkerhetskritisk programvara. Handboken innehåller utvecklingsmetoder, säkerhetsanalyser och testmetoder som leder till förbättrad säkerhet i datorsystemet. Det finns även en genomgång av olika programmeringsspråk.

Handbokens fokus är främst utveckling av programvara i säkerhetskritiska tillämpningar. Mycket av informationen och vägledningen är även tillämplig för utveckling av uppdragskritisk/missionskritisk programvara.

2.19 Def Stan 00-56

Storbritanniens militära standard Def Stan 00-56 edition 4, *Safety Management Requirements for Defence Systems* anger särskilda krav för systemsäkerhetsverksamheten. Standarden ska främst tillämpas av utvecklande industri i samverkan med den brittiska militära myndigheten. Standarden definierar systemsäkerhet till att omfatta frihet från person- och egendomsskada. Den ställer krav på att systemsäkerhetsverksamhet ska bedrivas för det tekniska systemets under hela dess livslängd.

Standarden är indelad i två delar. Del 1 anger krav på verksamheter och del 2 är en vägledning till del 1. Del 2 innehåller även en vägledning för komplexa elektroniska system avseende säkerhetsverksamhet.

Centralt i standarden är begreppet *Safety Case* som anger den process som ska genomföras för att erhålla säkra tekniska system. Genom ett definierat *Safety Case* detaljregleras de ingående aktiviteterna. Resultatet av de genomförda aktiviteterna rapporteras i en *Safety Case Report*. Noterbart är att begreppet ALARP (As Low as Reasonably Practicable) är ett lagstadgat begrepp i Storbritannien.

Standarden hanterar elektronik och programvara som en del av det tekniska systemet. I del 2 finns en vägledning till hur kraven på aktiviteter enligt del 1 ska tillämpas. Det finns ett avsnitt om

systemsäkerhet för system som innehåller komplexa elektroniska enheter som avser både hårdvara och programvara. För denna elektronik ska man identifiera potentiella vådahändelser som denna kan åstadkomma eller bidra till, definiera motverkande kvav och riskminskande åtgärder samt redovisa bevis för att felmoder och felsannolikheter är relevanta.

Kritikalitetsnivån för elektronik och programvara ska definieras vid utveckling. Standarder såsom IEC 61508, RTCA DO-178C eller Def Aust 5679 *The Procurement of Computer-Based Safety-Critical Systems* bör användas. Def-Stan 00-56 innehåller dock ingen detaljbeskrivning om hur framtagning av programvaran ska ske.

3 Arbetsgång mellan Försvarsmakten, FMV och industrin

Detta kapitel beskriver arbetsgången från Försvarsmaktens behov, via utvecklande industrins arbete, till FMV:s överlämning av tekniska system till Försvarsmakten.

3.1 Övergripande processbild och olika perspektiv

Försvarsmaktens tekniska system och produkter inrymmer oftast stora energier som styrs och övervakas av olika datorsystem. Vägen till säkra tekniska system börjar genom kravställning från Försvarsmakten till FMV och vidare till utvecklande industri.

Arbetsgången mellan Försvarsmakten, FMV och utvecklande industri beskrivs övergripande i *bild 3:1*. Bildens huvudsyfte är att visa på viktiga steg och leveranser mellan de olika aktörerna, och i följande avsnitt förklaras de olika stegen. Genom att vid rätt tidpunkt ställa krav på bland annat prestanda, systemsäkerhet, metodik och dokumentation ges utvecklande industri möjlighet att arbeta strukturerat och kostnadseffektivt.

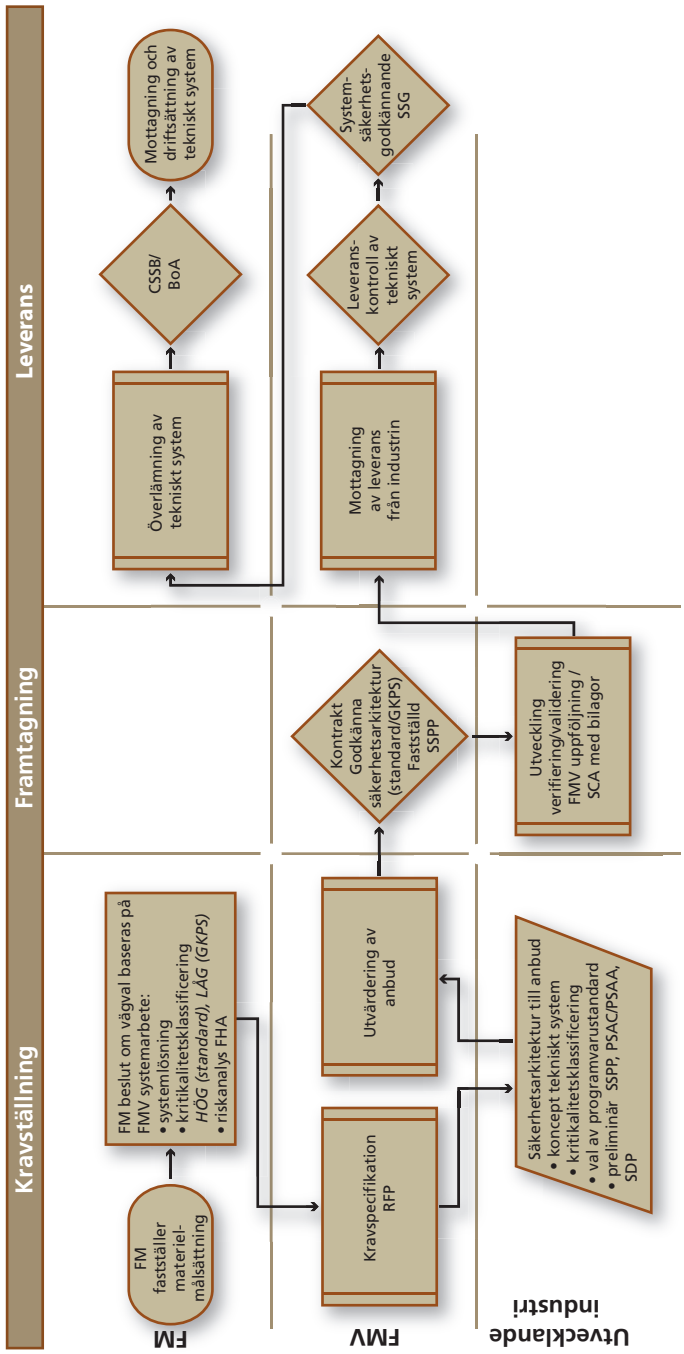


Bild 3:1 Förenklad processbild över arbetsgången mellan Försvarsmakten, FMV och utvecklande industri

Komplexiteten vid kravställning av tekniska system innehållande datorsystem kan variera mycket. *Bild 3:2* nedan pekar på ett antal olika aspekter som måste beaktas för en programvaras livscykel för en säker användning i militära tekniska system.

Då det inte finns någon etablerad programvarustandard som omfattar alla dessa aspekter beskriver handboken ett sätt att hantera dem. Detta syftar bland annat till att säkerställa att ställda säkerhetskrav uppfylls.

Arena	Läge	Uppgift	Komplexitet	Faser	Skyddsvärt	Konsekvens
Armé	Krig	Insats	System av system	Anskaffning	Person	Katastrofal
Flyg	Kris	Övning	Plattform	Drift	Egendom	Kritisk
Sjö	Fred	Utbildning	Produkt	Underhåll	Miljö	Allvarlig
Led			Programvara	Modifiering		Mindre allvarlig
				Avveckling		Negligerbar

Bild 3:2 Olika aspekter att beakta inför kravställning på tekniska system innehållande datorsystem

Den största påverkan på kravställningen, och i slutändan totalkostnaden för utveckling av programvaran, hänförs till vilken arena det tekniska systemet ska användas i, dess komplexitet samt konsekvensen för person, egendom och yttre miljö om en olycka skulle inträffa. Bilden ovan kan vara till stöd i dialogen mellan Försvarsmakten och FMV i de tidiga faserna. Den kan även användas som stöd vid en kontraktsgenomgång mellan FMV och utvecklande industri.

3.2 Försvarsmaktens målsättningsarbete

Försvarsmakten identifierar behov av ny förmåga, eller bibehållen förmåga genom materielomsättning, till ett eller flera olika förband samt om förmågan även ska kunna användas som stöd till det civila samhället i fredstid. Försvarsmakten fastställer Förbandsmålsättning, vilken bör inkludera krav på individrisk för personal inom aktuella förband. Motsvarande krav för egendom och yttre miljö ska också framgå. Därefter beställer Försvarsmakten ett systemarbete av FMV. I beställningen inkluderas även underlag gällande förutsättningar som krävs för att FMV ska kunna påbörja sitt systemarbete. Se *kapitel 6*.

FMV genomför systemarbete, bland annat utifrån perspektiven funktion, teknik, underhållslösning, kommersiellt och juridik. FMV följer härvid kraven i *kapitel 7* som en del i systemarbetet. Systemarbetet mynnar ut i ett eller flera möjliga alternativ till koncept och underhållslösningar. Efter Försvarsmaktens beslut om vilket alternativ som ska realiseras beställs ett utvecklingsuppdrag hos FMV.

Som en del i FMV:s beredningsarbete med Materielmålsättning gör FMV en inledande systemsäkerhetsanalys kallad *Functional Hazard Analysis* (FHA) i syfte att identifiera de dimensionerande olycksrisker. Resultatet redovisas för Försvarsmakten i syfte att säkerställa att det tekniska systemet kommer att uppnå de kravställda förmågorna.

Försvarsmakten fastställer Materielmålsättningen som bland annat innehåller FMV:s förslag på tolerabel risknivå för person, egendom och yttre miljö för det tekniska systemet. Av Materielmålsättningen ska även livslängd och driftprofil framgå. Vid överlämning av det tekniska systemet ska en återredovisning av kravuppfyllnad ske till Försvarsmakten.

3.3 FMV:s initiala systemsäkerhetsanalys

FMV genomför, tillsammans med Försvarsmakten, en *Funktionsinriktad systemsäkerhetsanalys* (FHA) på system- eller delsystemnivå. Som inledande riskanalys används metoden enbart på den högsta systemnivån. FHA används främst för att identifiera och klassificera systemfunktioner, samt för bedömning av konsekvenserna vid fel i dessa funktioner.

FHA används även för att identifiera yttre miljö- och hälsorelaterade konsekvenser som beror av funktionsrelaterade fel. Resultatet av riskanalysen används därefter för att bestämma vilka krav som ska ställas i upphandlingsunderlaget.

Följande arbetsgång kan tillämpas:

1. Ta fram en funktionell beskrivning av systemet.
2. Identifiera olycksrisker (topphändelser) som kan bli resultatet av utebliven funktion, nedsatt funktion, felaktig funktion eller oönskad aktivering av funktion.
3. Utvärdera de allvarligaste olycksriskerna (topphändelserna) som kan förknippas med varje identifierat fel hos en funktion.
4. Lista krav och förslag på riskminskande åtgärder, som när de är vidtagna, eliminerar eller minskar olycksrisken (topphändelsen). Utvärdera om de identifierade riskminskande åtgärderna kan implementeras i hårdvara, programvara eller liknade, beroende på hur kritiska dessa funktioner är.

En FHA-rapport bör innehålla följande information:

- En beskrivning av det tekniska systemet och dess huvudfunktioner.
- Identifierade olycksrisker (topphändelser) och deras allvarligaste konsekvenser på person, egendom och yttre miljö.
- Resultatet av riskanalysen där de identifierade olyckshändelserna med dess bedömda konsekvenser förtecknas i en Risklogg.
- Motivering om krav i upphandlingsunderlaget ska ställas på att en etablerad programvarustandard, tillämplig inom teknikområdet, ska användas, eller om *Grundkrav Programvarusäkerhet* (GKPS) kan anses vara tillräckligt.
- En beskrivning av den riskanalysmetod som använts.

3.4 FMV:s krav i förfrågningsunderlag till utvecklande industri

Utifrån resultatet av den funktionsinriktade systemsäkerhetsanalysen FHA använder FMV tillämpningsmatrisen för initial kritikalitetsklassificering av det tekniska systemet i *avsnitt 4.2*. Beroende på om det tekniska systemet kan orsaka allvarliga konsekvenser (*HÖG*) eller mindre allvarliga konsekvenser (*LÅG*) för person, egendom eller yttre miljö, görs en initial kritikalitetsklassificering enligt *bild 4:1*. Resultatet av kritikalitetsklassificeringen ger riktlinjer till FMV för vilka krav som ska ställas vid utveckling av programvaran. FMV ska även ställa krav på tolerabel risknivå samt ange sammanlagd drifttid för det tekniska systemet. Detaljerad arbetsgång beskrivs i *bilaga 3*.

3.5 Industrins anbud till FMV

FMV ställer alltid krav på att *Grundkrav Programvarusäkerhet* (GKPS) enligt *kapitel 8* ska uppfyllas och industrin ska därför alltid bekräfta detta i anbudet. Om FMV även har ställt krav på att etablerad programvarustandard ska följas, ska industrin i anbudet meddela vilken programvarustandard som industrin avser att följa samt ange motiv för detta.

I industrins anbud ingår att presentera en säkerhetsarkitektur för föreslaget tekniskt system. Vägledning och arbetsgång med att ta fram en säkerhetsarkitektur beskrivs i *avsnitt 4.3* samt med exempel i *bilaga 4*.

I anbudet ska alltid en preliminär *Systemsäkerhetsplan* (SSPP) finnas och en *Utvecklingsplan programvara* (SDP) bifogas. I tillämpliga fall ska även en *Certifieringsplan för programvara* (PSAC) eller *Acceptansplan för programvara* (PSAA) bifogas. Efterfrågad information i angivna dokument redovisas enligt dokumentlistan, *kapitel 9*.

3.6 FMV:s och industrins kontrakt och kontraktsgenomgång

FMV lägger beställning på utvecklande industri utifrån ställda krav i förfrågningsunderlaget. Av beställningen ska det i förekommande fall framgå vilken etablerad programvarustandard som industrin har åtagit sig att följa.

Under FMV:s och industrins kontraktsgenomgång fastställs *Systemsäkerhetsplanen* (SSPP), *Utvecklingsplan programvara* (SDP) och i förekommande fall, exempelvis *Certifieringsplan för programvara* (PSAC) alternativt *Acceptansplan för programvara* (PSAA).

Industrin förfinar säkerhetsarkitekturen och ska vid kontraktsgenomgången kunna motivera val av kritikalitetsklassificering för programvaran utifrån vald programvarustandard. Industrin ska redovisa om något enskilt krav i GKPS inte är tillämpbart och/eller kan uppfyllas på annat sätt. Industrin och FMV ska även överenskomma hur GKPS ska verifieras. Överenskommelser dokumenteras i protokoll mellan parterna.

Om industrin i sitt arkitekturarbete kan visa en systemlösning där programvaran har kritikalitetsklassificering ***LÅG***, kan parterna överenskomma om att *Grundkrav Programvarusäkerhet* (GKPS) är tillräckligt att följa. Sådan överenskommelse dokumenteras i kontraktet mellan FMV och industrin.

3.7 FMV:s insyn och uppföljning av industrins arbete

FMV bör ställa krav på särskild insyn i industrins utvecklings- och tillverkningsprocess. Detta görs lämpligen genom att kontraktera standarderna AQAP 2110/2210. Se *avsnitt 5.2*.

I samband med konstruktionsgenomgångar mellan FMV och industrin följs systemsäkerhetsarbetet upp utifrån överenskomna planer såsom SSPP och SDP, samt i förekommande fall i certifieringsdokumenten PSAC/PSAA.

Under utvecklingen ska dokumentationen successivt granskas av FMV. En tidig validering av krav och konstruktion bör planeras med avseende på systemsäkerhetskraven. Speciellt viktigt är detta vid utformning av användargränssnitt där operatörens förmåga att kunna identifiera ett uppkommet farligt fel eller farlig situation, och vidta åtgärder innan det kan leda till vådahändelser och olyckor.

FMV ska även ges möjlighet att medverka vid verifiering av det tekniska systemet hos industrin.

3.8 FMV:s leveranskontroll av tekniska system

Inför leverans av tekniska system till Försvarsmakten, granskar FMV industrins systemsäkerhetsutlåtande (SCA) med tillhörande bilagor. Som del av granskningen ingår att kontrollera att industrin har uppfyllt och verifierat de av FMV i beställningen ställda systemsäkerhetskraven, inklusive *Grundkrav programvarusäkerhet* (GKPS). Den dokumentation som överenskommits i SSPP, utifrån dokumentlistan i *kapitel 9*, granskas och godkänns av FMV. FMV bör före leverans delta vid validering av det tekniska systemet hos industrin.

3.9 FMV:s överlämning av tekniska system till Försvarsmakten

Innan FMV utfärdar systemsäkerhetsgodkännande (SSG), ska dialog föras med Försvarsmakten om aspekter kring systemuppdateringar i det tekniska systemet. Dessa aspekter framgår av *kapitel 6*. När samtliga frågetecken är utklarade överlämnar FMV det tekniska systemet till Försvarsmakten enligt ordinarie överlämningsrutin.

3.10 Försvarsmaktens mottagning och driftsättning av tekniska system

Baserat på FMV:s dokumentation avseende systemsäkerhet kan Försvarsmakten fatta *Centralt Systemsäkerhetsbeslut* (CSSB).

3.11 Systemuppdateringar under drift

Systemuppdateringar kan initieras dels av utvecklande industri utifrån ett produktansvar, dels av FMV på uppdrag av Försvarsmakten. Detta avser såväl rättningar av fel i programvara som funktionstillväxt i det tekniska systemet eller funktionsanpassning till omgivande system.

Systemuppdatering initierad av utvecklande industri för rättning av fel i programvara genomförs av utvecklande industri i samråd med FMV. Systemuppdatering i form av funktionstillväxt för programvara, genomförs av utvecklande industri enligt FMV beställning. Införande på förband sker i enlighet med av FMV fastställd Teknisk Order (TO). Undantag kan finnas om Försvarsmakten själva är tekniskt designansvariga.

Om det tekniska systemet innehåller tidigare utvecklad programvara (PDS) kan FMV välja att teckna ett underhållsavtal med leverantören av PDS. Detta görs för att kunna erhålla information om uppdateringar, samt för att få tillgång till dessa uppdateringar, inklusive viss dokumentation.

Varje ändring i programvaran i det tekniska systemet ska betraktas som en större ändring och ska följas av nya systemsäkerhetsbeslut i enlighet med H SystSäk.

Omställning av ändringsbara parametrar kan tillåtas om genomförda systemsäkerhetsanalyser har visat att detta inte förändrar bedömningen att tidigare identifierade olycksrisker. En ändringsbar parameter kan till exempel vara att förändra ett riskområde för viss ammunition. I detta fall kan det betraktas som en mindre ändring enligt H SystSäk.

3.12 Avveckling av programvara i tekniskt system

FMV:s avvecklingsskrivelse ska även beskriva hur programvarorna, deras utvecklingsmiljöer, samt hur programvarulicenser och underhållsavtalen ska hanteras.

4 Säkerhetsarkitektur och metodik

Syftet med detta kapitel är att beskriva vikten av att utarbeta en genomtänkt säkerhetsarkitektur för datorsystem utifrån Försvarsmaktens behov av tekniska system. Som stöd för detta presenteras en metodik för utveckling och provning. Denna involverar samtliga aktörer oavsett livscykelfas och systemnivå.

4.1 Tillämpningsmatris för initial kritikalitetsklassificering av det tekniska systemet

Försvarsmaktens Handbok Systemsäkerhet (H SystSäk) är en svensk anpassning av MIL-STD 882. Avsnitt 4.4 och Appendix B i MIL-STD 882 ersätts i huvudsak av denna handbok (H ProgSäk). Den underliggande standarden AOP-52 tillämpas inte, se *avsnitt 2.15*.

Nedan finns den tillämpningsmatris som beskriver kopplingen till den riskmatris som används i H SystSäk för att redovisa kvarstående olycksrisker för tekniska system och möjliga konsekvenser av vådahändelser med koppling till programvaran.

Initial kritikalitetsklassificering genomförs enligt *bild 4:1*. Genom att välja en lämplig säkerhetsarkitektur kan det säkerhetskritiska datorsystemets bedömda kritikalitetsnivå hållas låg, se *avsnitt 4.3*. Slutlig kritikalitetsklassificering genomförs efter arkitekturarbetet enligt *avsnitt 4.4*.

Utveckling av programvara sker i första hand genom att tillämpa generella eller sektorsspecifika etablerade programvarustandarder. Ett urval av programvarustandarder finns beskrivna i *kapitel 2*. Standarderna anger metoder för att reducera att systematiska fel införs under utvecklingen av programvaran.

Om utvecklande industri i sitt arkitekturarbete (med hjälp av tillförda säkerhetsfunktion, diversitet, redundans, övervakning med mera kan visa att systemets olycksrisker, som datorsystemet kan påverka, har låg eller negligerbar konsekvens för person-, ekonomi och/eller miljöskador är *Grundkrav Programvarusäkerhet* (GKPS) i *kapitel 8* tillräckliga att uppfylla. Användning av enbart GKPS ska överenskommas med FMV.

I de fall produkten kommer att användas fristående och är CE-märkt, eller kommer att CE-märkas, tillämpas *avsnitt 10.1–10.3* i denna handbok. Detta gäller även för tekniska system som godkänts av annan aktör, till exempel främmande makt eller samarbetsorgan inom Nato. Se *avsnitt 10.4*.

För initial kritikalitetsklassificering av det tekniska systemet, som genomförs av FMV, ska *bild 4:1*. Tillämpningsmatris för FMV:s initiala kritikalitetsklassificering av tekniska system tillämpas enligt beskrivningen nedan.

För tekniska system identifieras och analyseras de allvarligaste olycksriskerna för person, egendom och yttre miljö. För dessa olycksrisker görs en uppskattning av deras allvarligaste konsekvenser enligt ett av alternativen nedan:

a. Om konsekvenserna bedöms vara ***HÖG*** (hög, allvarlig eller medel) ska FMV i anbudsinfordran ställa krav på att utvecklande industri ska tillämpa en etablerad programvarustandard i utvecklingsarbetet parallellt med *Grundkrav Programvarusäkerhet* (GKPS) enligt *kapitel 8*.
b. Om konsekvenserna bedöms vara ***LÅG*** (låg eller ingen) ska FMV i anbudsinfordran ställa krav på att utvecklande industri alltid ska tillämpa *Grundkrav Programvarusäkerhet* (GKPS) enligt *kapitel 8*. Det står dock alltid utvecklande industri fritt att följa en etablerad programvarustandard parallellt med *Grundkrav Programvarusäkerhet* (GKPS).

Tillämpningsmatris kopplad till MIL-STD 882E för FMV:s initiala kritikalitetsklassificering av tekniska system			
Nivå avseende konsekvens	Beskrivning	Genomförande	FMV:s initiala kritikalitets-klassificering
Hög	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***katastrofal*** konsekvens för person, ekonomi och/eller miljö *(flera eller enstaka dödsfall, total systemförlust och/eller bestående miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för högsta kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	***HÖG*** FMV ställer krav på industrin att etablerad programvaru-standard ska följas.
Allvarlig	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***kritisk*** konsekvens för person, ekonomi och/eller miljö *(allvarliga och bestående personskador, omfattande ekonomisk och/eller miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för högre kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	
Medel	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***allvarlig*** konsekvens för person, ekonomi och/eller miljö *(allvarliga men inte bestående person-skador, betydande ekonom-isk och/eller miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för medelhög kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	
Låg	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***marginell*** konse-kvens för person, ekonomi och/eller miljö *(mindre allvarlig personskada, mindre ekonomisk och/eller miljöskada).*	Grundkrav för utveckling av programvara för lägsta tolerabla kritikalitetsnivå tillämpas (GKPS).	***LÅG*** FMV ställer krav på industrin att lägst GKPS ska användas. *(Industrin kan dock välja att följa en eta-blerad program-varustandard)*
Ingen	Tekniskt system innehållande programvara där konsekven-sen av olycka medför negligerbar konsekvens för person, ekonomi och/eller miljö.	Grundkrav för utveckling av programvara för lägsta tolerabla kritikalitetsnivå tillämpas (GKPS).	

Bild 4:1 Tillämpningsmatris kopplad till MIL-STD 882E för FMV:s initiala kritikalitetsklassificering av tekniska system

4.2 Datorsystemets egenskaper

Ett datorsystem med sin programvara har vissa unika egenskaper. Även kombinationer av identiska datorsystem eller av olika datorsystem påverkar såväl systemsäkerhet som tillgänglighet.

4.2.1 Programvarans egenskaper

En programvara har speciella egenskaper som skiljer sig från mekaniska och elektriska system. Nedan finns ett antal systemsäkerhetsrelaterade egenskaper listade.

En programvara:

- Innehåller enbart systematiska fel och har inga slumpartade fel, även om programvarans felyttringar kan uppfattas som slumpmässiga till exempel beroende på att insignalerna är slumpmässiga.
- Erhåller fel vid framtagning av kravspecifikation och/eller vid kodning. Dessa systematiska fel finns i konstruktionen från början men kan visa felyttring långt senare vid en förändrad användningsprofil eller användningssätt.
- Slits eller nöts inte ut med tiden.
- Olika delar kan kräva olika kritikalitet, där överordnade systemet ställer krav på programvarans kritikalitetsnivå. Högsta kritikalitet bestämmer hela programvarans samlade kritikalitet, enligt metod för respektive programvarustandarder i *kapitel 2*.
- Som genom redundans integreras i det tekniska systemet, minskar inte de systematiska felen men ger möjlighet till ökad tillgänglighet på systemnivå (se förklaring programvaruredundans).
- Kan genom diversitet integreras i det tekniska systemet vilket kan reducera de systematiska felen (se förklaring programvarudiversitet och funktionsövervakning).

4.2.2 Felupptäckt i system

För tekniska system är det viktigt att man har funktionalitet för att upptäcka slumpmässiga hårdvarufel. Fel i systemet som inte upptäcks i tid kan leda till fel i en säkerhetskritisk funktion. Felupptäckt kan åstadkommas på olika sätt och genom olika kombinationer av nedanstående tekniker.

- *Inbyggd test* (Build In Test, BIT) i form av *Säkerhetskontroll*, SK (Safety Check, SC/Power On Bit, PBIT) vid uppstart.
- *Funktionsövervakning*, FÖ (Functional monitoring, FM/Continuous BIT, CBIT) under drift, se exempel i *bild 4:11* och *bild B4:8*.
- *Funktionskontroll*, FK (Functional Check, FC/Initiated BIT, IBIT) som underhållsåtgärd för felverifiering före start eller vid underhåll av systemet.
- *Integritetskontroll*, checksumma på programvaran för att säkerställa att den inte har förändrats.
- *Felhantering*, fel som uppkommer under drift kan hanteras på ett sådant sätt att systemet fortsätter att fungera med reducerad funktionalitet eller prestanda.
- *Jämförare*, för att kunna välja en redundant eller diversifierad kanal.
- *Vakthund* (Watchdog) i datorsystemet, tillsammans med en strukturerad och deterministisk programvara, gör det möjligt att upptäcka fel i programvaruexekveringen.
- *Spänningsövervakning*, ett datorsystem där spänningsmatningen inte uppfyller definierade krav kan medföra att hela eller delar av datorsystemets hårdvara inte fungerar som avsett. Spänningsövervakning hanteras av speciellt utformade hårdvarukretsar som också kan utgöra villkor för Watchdogfunktionen.

4.2.3 Redundans respektive diversitet i datorsystem

Genom att föra in två identiska datorsystem med samma programvara för att lösa samma funktion (replika) kan man hitta slumpmässiga hårdvarufel i systemet. Genom att jämföra utdata från datorsystemen kan man se om resultatet skiljer sig för mycket från varandra och på så sätt avgöra om något är fel. I ett två-kanalssystem måste således båda kanalerna visa överensstämmelse för full funktionalitet i systemet, se *bild 4:2*. Redundans kan finnas för givare, ställdon, i datorsystemet med programvara samt på displayer för utdata till operatör.

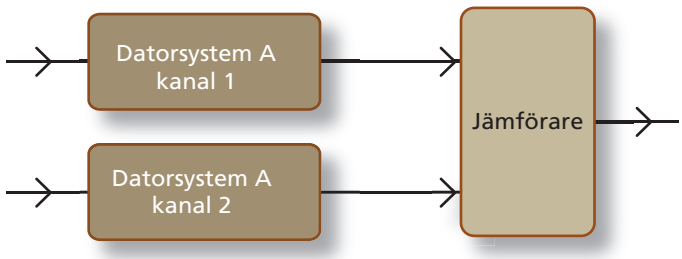


Bild 4:2 Redundant två-kanalssystem med identiska datorsystem och programvara med separata indatakanaler

Om man har tre identiska system för styrning, och ett av dessa går fel, så går det ofta att avgöra vilket av dessa som är fel, det vill säga två datorsystem visar liknade resultat och det tredje avviker från de övriga. Redundanta system med röstningsfunktion kan detektera slumpmässiga hårdvarufel och därmed höja både tillgänglighet och systemsäkerhet i ett tekniskt system. Se *bild 4:3*.

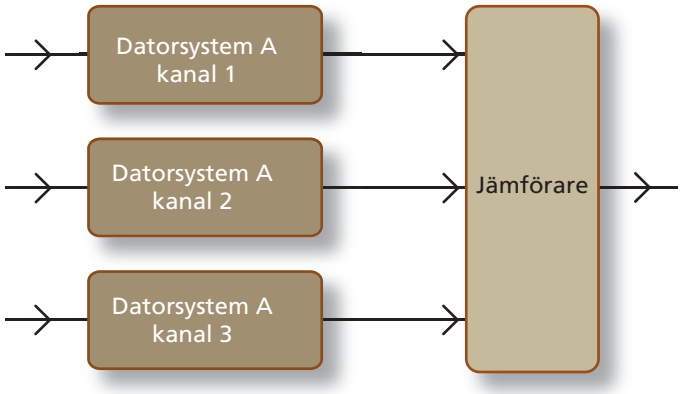


Bild 4:3 Redundant fler-kanalssystem med tre identiska datorsystem och programvaror med separata indatakanaler

Redundanta system med röstningsfunktion mellan tre olika datorsystem med diversitet kan förutom de slumpmässiga hårdvarufelen, även detektera systematiska fel i både program- och hårdvaran, se *bild 4:4*.

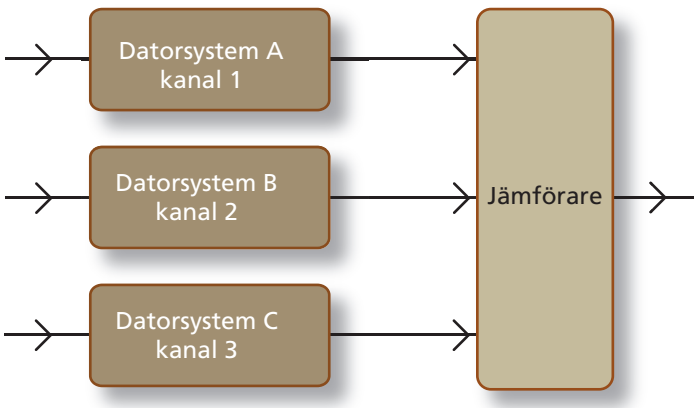


Bild 4:4 Redundant fler-kanalssystem med tre olika datorsystem och tre olika programvaror och separat indatakanaler

Genom att införa diversitet med olika programvaror i datorer för samma funktion ökar möjligheten att även hitta fel i programvaran. Diversitet kan föras in i konstruktionen på olika sätt under utvecklingen, dels via funktionell diversitet (det vill säga att man inte har fel i en gemensam kravbild), dels via konstruktionsdiversitet (inga gemensamma fel i metoder, använda verktyg med mera.) Valet av vilken diversitet som är lämplig för det aktuella tekniska systemet bör noga övervägas så att det inte leder till en för hög komplexitet. Det kan också vara svårt att visa att man också uppnått en diversitet för att kunna motverka systematiska fel.

I fler-kanalssystem så blir jämföraren den mest kritiska komponenten.

4.2.4 Felsäkert läge (Safe state) för tekniskt system

Då fel upptäcks i ett datorsystem behöver feltillståndet ofta tas om hand. Ett sätt är att försätta funktionen eller delfunktionen i felsäkert läge, så kallad *Safe State*. Felsäkert läge betyder ofta att systemet vid fel går in i ett tillstånd med reducerad funktionalitet eller prestanda.

Ett felsäkert läge varierar från tekniskt system till tekniskt system och kan därför inte definieras allmängiltigt. För varje funktion måste därför det felsäkra läget preciseras så långt detta är möjligt, det vill säga det läge där en vådahändelse på grund av fel i datorsystemet kan förhindras. Ett roterande system kan inta felsäkert läge då de mekaniska bromsarna aktiveras och styrningen från datorsystemet kopplas bort. Ett avfyringssystem kan inta felsäkert läge då energin till tändkretsen kopplas bort. Ett flygplan kan inta felsäkert läge på landningsbanan då start förhindras på grund av detekterat fel i datorsystemet. Om det finns driftfall där ett felsäkert läge inte kan definieras ska detta dokumenteras.

Varje tänkbart fel i systemet behöver analyseras med avseende på konsekvens och eventuell påverkan, samt hur felet ska hittas, hur det påverkar funktionen och hur det ska tas om hand. Ett felsäkert läge kan vara tillräckligt för att reducera konsekvensen av ett allvarligt fel så att vådahändelsen kan undvikas. För kritiska funktioner kan reserv- och/eller nödsystem behövas. Vid omstart ska systemet utgå från ett definierat säkert tillstånd.

4.3 Säkerhetsarkitektur, metodik och arbetsgång

För att kunna bestämma graden av programvarans påverkan på det slutliga tekniska systemet behöver programvaran kritikalitetsklassificeras. Handboken utgår från Försvarsmaktens princip för kravställning på förband och tekniska system och tar vid där en säkerhetsarkitektur för datorsystemet ska tas fram. Nedan föreslås en modell som bygger på samma principer som i en felträdsanalys. Andra modeller än nedan kan förekomma.

4.3.1 Olycksmodell

Olyckor är ofta mycket komplexa händelser sett till de orsaker och indirekta förhållanden som föranlett dem. Varje olycka är också en unik händelse. En olycksmodell kan därför aldrig till fullo beskriva alla enskilda olyckor utan endast uttrycka en generell bild. Dock kan olycksmodellen i *bild 4:5* utgöra ett stöd för tanken vid riskhantering.

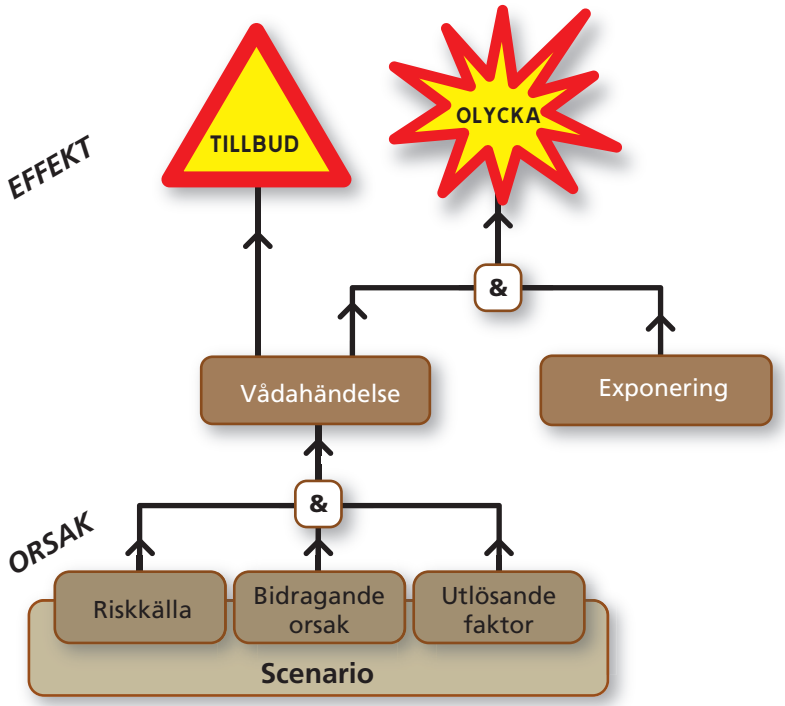


Bild 4:5 Förenklad olycksmodell enligt H SystSäk

Följande punkter förklarar begreppen i olycksmodellen och sätter dem i relation till varandra:

- En *Riskkälla* är något som kan skada person, egendom eller yttre miljö genom dess farliga egenskaper.
- Ett *Scenario* är en situation eller ett systemtillstånd där en riskkällas farliga egenskaper finns närvarande.
- En *Vådahändelse* är en oönskad händelse som inträffat oplanerat, av våda, det vill säga utan uppsåt, och som kan resultera

i ett tillbud om inget exponeras eller i en olycka om någon eller något exponeras.

- En *Vådahändelse* har alltid en eller flera *bidragande orsaker*. Den eller de direkta orsaker som framkallade vådahändelsen kallas *utlösande faktorer*.
- Ett *Tillbud* är en vådahändelse som inte resulterar i några skador.
- En *Olycka* är resultatet av en *Vådahändelse* då någon/något *Exponeras* för riskkällan och därvid skadas.
- *Konsekvensen* av en olycka utgörs av en uppkommen skada på person, egendom eller yttre miljö och redovisas för varje skadeklass i H SystSäk.

Med tekniska system avses alla typer av plattformar såsom flygplan, fartyg och stridsfordon, samt produkter såsom medicinteknisk utrustning och hushållsmaskiner.

I tekniska system ingår oftast någon styr- eller övervakningsfunktion som realiseras med hjälp av ett datorsystem. Datorsystemet är enligt definition säkerhetskritiskt om det styr eller övervakar energier, som vid ett okontrollerat förlopp kan orsaka vådahändelser och i förlängningen olyckor. Även datorsystem i säkerhets- och nödsystem ingår i denna kategori, även om de inte direkt styr farliga energikällor.

Enligt definitionerna ovan är riskkällor de energier som datorsystemet direkt eller indirekt styr eller övervakar. Brister i datorsystemets styr- eller övervakningsfunktioner kan här ses som bidragande orsaker till att vådahändelser inträffar.

Kraven på prestanda kan stå i motsatsförhållande till kraven på säkerhet eftersom komplicerade säkerhetsfunktioner kan medföra minskad taktisk förmåga och tillgänglighet. Vid konstruktionen av ett säkerhetskritiskt datorsystem ska därför målet alltid vara att hålla systemet inom krav på tolerabel risknivå under dess hela livslängd, utan att detta medför begränsningar i det taktiska användandet.

I fortsättningen diskuteras åtgärder för att kunna reducera sannolikheten för vådahändelse i ett tekniskt system. Kravet på tolerabel risknivå för det tekniska systemet bryts ner till krav på sanno-

likhet för respektive vådahändelse utifrån kravställd användningsprofil. Vid systemutformning ska dessa krav tidigt beaktas så att en systemstruktur kan erhållas där det finns rimliga förutsättningar att kunna påvisa och verifiera systemsäkerhetskraven.

Försvarsmakten anger krav på tolerabel risknivå för enskild olycksrisk utifrån en given driftprofil och användningsmiljö. Den tolerabla risknivån för enskild olycksrisk kopplas till sannolikhet för vådahändelsen genom att sannolikheten för exponering sätts = 1. Industrin utvecklar ett tekniskt system där sannolikheten för vådahändelse är så pass låg att kravet på tolerabel risknivå för olycksrisk uppfylls.

4.3.2 Kravnedbrytning av dimensionerande vådahändelser

Syftet med att göra en säkerhetsarkitektur är att minska datorsystemets kritikalitet i ett tekniskt system så långt som detta är praktiskt möjligt, det vill säga en sammanvägning av systemsäkerhets- och tillgänglighetskrav kopplat till kostnad.

Valet av säkerhetsarkitektur ska göras på sådant sätt att det inte ökar komplexiteten på det tekniska systemets utformning. En avvägning bör alltid göras så att de centrala säkerhetsprinciperna såsom enkelhet, oberoende och determinism uppnås. Detta underlättar förståelse av det tekniska systemets uppbyggnad, ger gynnsammare förutsättningar för verifieringen samt underlättar framtida systemuppdateringar.

En programvara har särskilda egenskaper och är i princip omöjlig att göra helt felfri för alla användningssätt och kombinationer av indata. Genom att använda sig av funktionsövervakning under datorsystemets olika driftmoder där jämförelse sker med ett förväntat resultat kan många slumpmässiga fel i datorsystemet identifieras tidigt innan de leder till ett farligt fel som påverkar systemets omgivning. Används diversitet i funktionsövervakningen kan även vissa systematiska fel identifieras och elimineras.

Nedan ges en modell för hur en kravnedbrytning kan genomföras. Utifrån ingångskravet på sannolikhet för vådahändelse (topphändelsen) görs en kravnedbrytning i en generell felträdsmodell till en ansatt felsannolikhet för respektive bashändelse.

Det ansatta nedbrutna kravet på felsannolikhet i bashändelsen får sedan utgöra ingångskrav vid val av lämpliga processer i konstruktionsarbetet. För hårdvara finns beräkningsmodeller för att kunna prediktera felsannolikheter, men för programvara (syste-

matiska fel) är detta inte möjligt utan där utgör istället det nedbrutna kravet ett ingångsvärde för val av utvecklingsmetoder med lämplig stringens.

Principen är att utgå från de mest kritiska vådahändelserna som kan inträffa i det tekniska systemet och låta detta påverka utformningen av säkerhetsarkitektur så att kritikalitetsnivån på datorsystemen blir så låg som är praktiskt möjligt. Tidigt i arkitekturarbetet kan en felträdmodell tas fram för det tekniska systemets mest kritiska vådahändelser.

Syftet med kravnedbrytningen är att tidigt i arbetet med säkerhetsarkitekturen kunna identifiera de delar som kommer att styra kravställningen på kritikalitetsnivån för datorsystemet, det vill säga både felsannolikheten för slumpmässiga hårdvarufel och stringens i utvecklingsmetodik för programvaran. Om oberoende säkerhetsfunktioner införs i det tekniska systemet, kan också kritikalitetsnivån sänkas i motsvarande grad för den säkerhetskritiska funktionen, och därmed också för datorsystemet.

En kravnedbrytning utförs för de mest kritiska vådahändelserna (sannolikheten P2) så att kravet på sannolikheten för olycka (med sannolikheten P1) för det tekniska systemet kan innehållas utifrån en given användningsprofil och givna operationsbetingelser.

Kravnedbrytningen kan presenteras i ett generellt felträd, se *bild 4:6*, och består minst av vardera en:

- Säkerhetskritisk funktion (med sannolikheten P4).
- Säkerhetsfunktion (med sannolikheten P5).

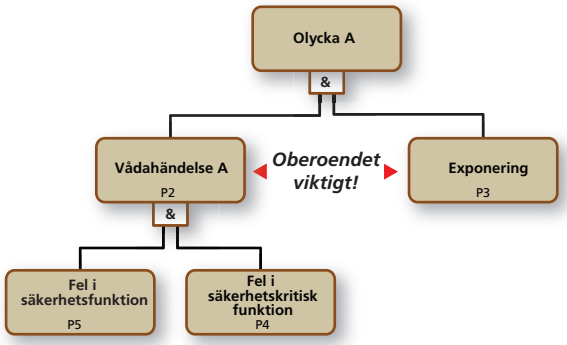


Bild 4:6 Generellt felträd för att beskriva relationer i olycksmodellen

Med säkerhetskritisk funktion (P4, datorsystemet) avses systemfunktion som vid farligt fel kan orsaka en vådahändelse. Det är i den säkerhetskritiska funktionen (P4) som det tekniska systemets logik finns som styr eller påverkar riskkällan och som ger systemet dess önskade funktion.

Med säkerhetsfunktion (P5) menas en tillförd fullständigt oberoende funktion vars enda syfte är att reducera sannolikheten för att vådahändelse (P2) ska inträffa vid farligt fel i den säkerhetskritiska funktion (P4).

Olyckan A (P1, topphändelsen) inträffar endast om vådahändelsen A (P2) inträffar samtidigt som något skyddsvärt exponeras (P3) såsom person, egendom eller yttre miljö, samtidigt som fel i både säkerhetsfunktion (P5) och fel i säkerhetskritisk funktion (P4) uppstår. Exponering påverkas av användningsprofilen vilken definieras av Försvarsmakten och kan förändras under ett tekniskt systems livslängd.

I kravnedbrytningen ansätts därför initialt en konservativ sannolikhet för exponering (sannolikhet = 1), vilket ger sannolikheten för olycka = sannolikheten för vådahändelse, det vill säga P1=P2, se *bild 4:7*.

Förenklat antagande för exponering av person, egendom och yttre miljö:

Inledningsvis antas att sannolikheten för exponeringen (P3) = 1. Detta antagande kan ibland vara för konservativt. Om antagandet leder till orimliga krav avseende sannolikheten för vådahändelse tillika olycka så ska en analys genomföras för att definiera en för tillämpningen realistisk exponering. Den nya ansatta exponeringen ska överenskommas med FMV.

Kravet på vådahändelse A (P2) bryts ned på säkerhetsfunktion (P5) och säkerhetskritisk funktion (P4), se *bild 4:7*.

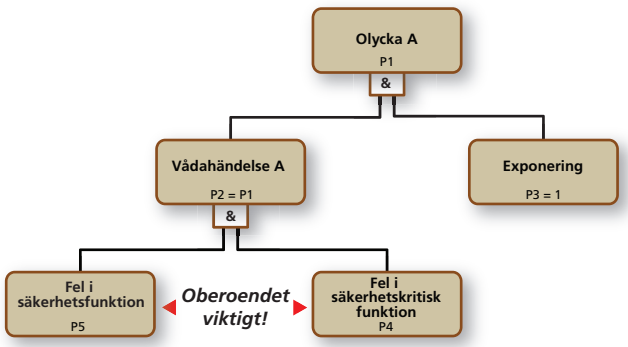


Bild 4:7 Kravnedbrytning av vådahändelse på säkerhetsfunktion och säkerhetskritisk funktion

Enheten för sannolikhet måste vara definierad utifrån Materielmålsättningen, det vill säga per individ av tekniskt system per tidsenhet avseende dödsfall, egendomsförlust eller allvarlig miljöskada (endast allvarligaste konsekvens, skadeklass I i H SystSäk). Notera att felfrekvens eller felsannolikhet kan anges på olika sätt såsom per användningstillfälle, per timme, per år eller per livslängd.

Vid konstruktionen av säkerhetsfunktionen bör enkelhet samt kända beprövade teknologier företrädesvis användas. Om säkerhetsfunktionen kan realiseras med delsystem där stor erfarenhet finns sedan tidigare och där felmoder och felfrekvenser är kända, underlättar detta också vid verifieringen av kravet.

Vid utvecklingen av den säkerhetskritiska funktionen ska målet givetvis resultera i en så låg sannolikhet för farliga fel som praktiskt är möjligt, men detta kan vara svårt att verifiera om den säkerhetskritiska funktionen är realiserad i ett datorsystem med många samverkande programvaror. Ur systemsäkerhets- och verifieringssynvinkel är det oftast en bättre strategi att allokera säkerhetskraven till systemets säkerhetsfunktioner.

4.3.3 Kravnedbrytning av vådahändelsen

Vid kravnedbrytningen kan den säkerhetskritiska funktionen också delas upp i ett antal oberoende redundanta diversifierade funktioner. Se *bild 4:8* för en fler-kanalig systemarkitetur, det vill säga det måste vara ett samtidigt farligt fel i de båda kanalerna A1 och A2 för att farligt fel ska uppstå i den säkerhetskritiska funktionen. På detta sätt kan det nedbrutna kravet på säkerhetskritisk funktion i det ideala fallet ytterligare brytas ned på oberoende delfunktioner. En omfördelning kan då tidigt göras i systemdesignen om orimliga eller svårverifierbara krav identifierats.

Utifrån denna kravnedbrytning på säkerhetsfunktion, säkerhetskritisk funktion, redundans och diversitet styrs sedan valet av säkerhetsarkitekturen för att kunna säkerställa att kravet på vådahändelse kan inrymmas.

Ett fullständigt oberoende är praktiskt svårt att realisera, det viktiga är att de beroenden som kan finnas är identifierade.

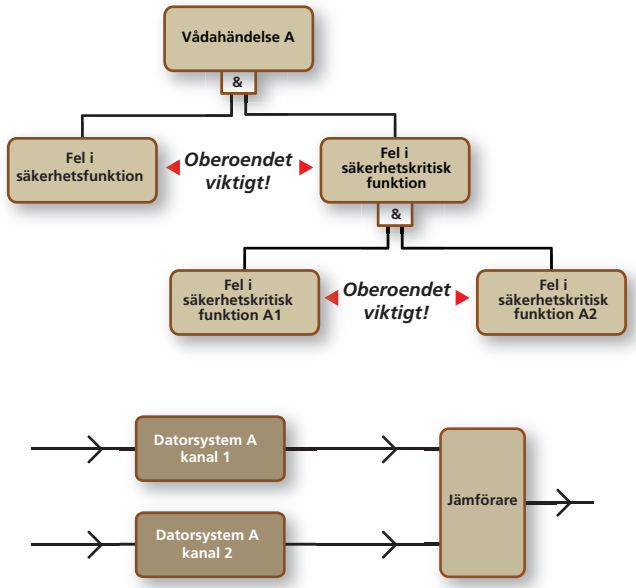


Bild 4:8 Säkerhetskritiskt system, fler-kanaligt redundant system (replika)

Om en-kanalig säkerhetsarkitektur används bör kravnedbrytningen fördelas så att säkerhetsfunktionen tar så stor del av kravet som möjligt. Detta för att säkerhetsfunktionen är enklare att verifiera än den säkerhetskritiska funktionen, se *bild 4:9* nedan.

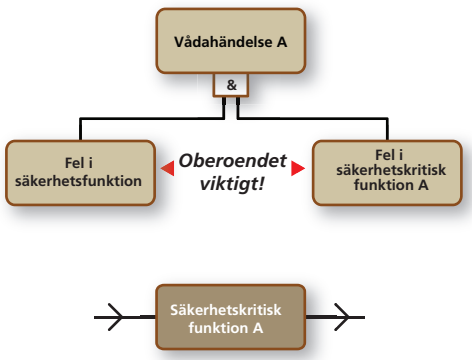


Bild 4:9 Säkerhetskritiskt system, en-kanaligt

4.3.4 Generiskt felträd för kravnedbrytning av vådahändelse

I den fortsatta beskrivningen används exemplet för ett en-kanaligt säkerhetskritiskt system. Detta är också tillämpbart i varje gren under säkerhetskritisk funktion i det fler-kanaliga exemplet enligt *bild 4:8*.

I den fortsatta nedbrytningen så anges den säkerhetskritiska funktionen av tre grenar i felträdet. Dessa är ställdon, givare och datorsystem. Alla grenar kan var och en för sig, direkt eller indirekt, orsaka ett farligt fel i den säkerhetskritiska funktionen, därav ”ELLER”-grinden, se bild *bild 4:10*.

Ställdonet i felträdet symboliserar datorsystemets koppling mot riskkällan. Det är via ställdonet som datorsystemet styr eller påverkar sina anslutna energier. Ett farligt fel i den säkerhetskritiska funktionen kan direkt vara orsakat av ett farligt fel i ställdonet, det vill säga datorsystemet styr ställdonet på avsett sätt, men felet i ställdonet leder till ett farligt fel i den säkerhetskritiska funktionen.

Givare i felträdet symboliserar datorsystemets återkoppling av hur riskkällan styrs. Ett fel i givaren leder till att datorsystemet får en felaktig återkoppling av tidigare utförda styrningar via ställdonet. Fel i givaren kan resultera i att datorsystemet styr ställdonet på ett felaktigt sätt så att ett farligt fel uppstår i den säkerhetskritiska funktionen.

Datorsystemet i felträdet symboliserar både datorsystemets hård- och programvara. Ett fel i datorsystemet kan resultera i att ställdonet styrs på ett okontrollerat sätt, vilket då kan medföra ett säkerhetskritiskt fel i den säkerhetskritiska funktionen.

Bild 4:10 Reducerat generiskt felträd, en-kanaligt säkerhetskritiskt system

För att ytterligare reducera sannolikheten för vådahändelse, tillförs i nästa steg övervakning/diagnostik av säkerhetsfunktion, ställdon och givare. Övervakningens syfte är att kunna detektera fel i respektive övervakad del innan felen leder till ett farligt fel för att på så sätt ytterligare kunna reducera kritikalitetsnivån på det säkerhetskritiska datorsystemet. Felträdet enligt *bild 4:10* utökas då enligt *bild 4:11* nedan.

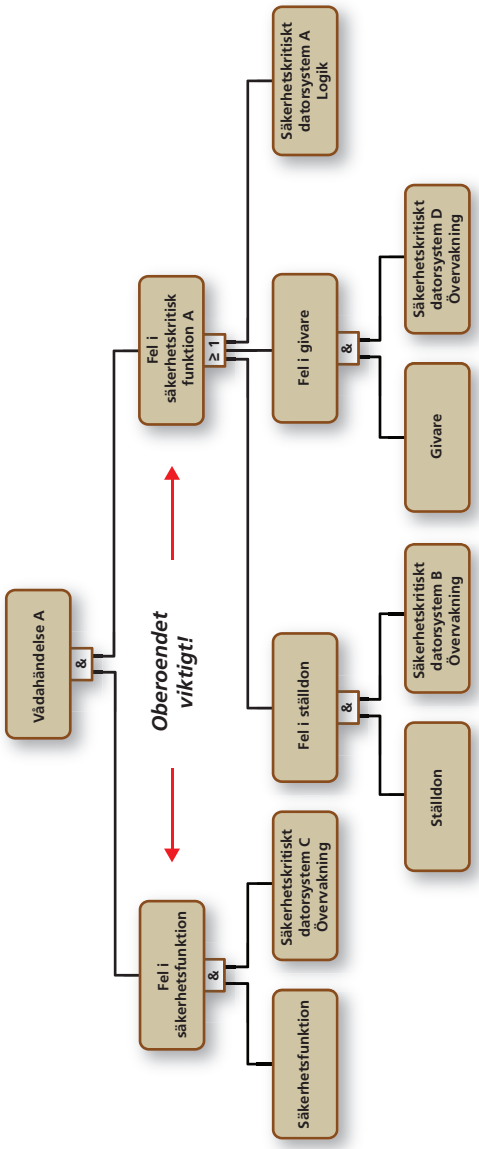


Bild 4:11 Reducerat generiskt felträd för ett en-kanaligt säkerhetskritiskt system med oberoende övervakning

Händelsen *Fel i säkerhetsfunktion* definieras här bestående av två bashändelser. Dels en *Säkerhetsfunktion*, dels en oberoende övervakningsfunktion (*Säkerhetskritisk Datorsystem C övervakning*). Övervakningen tillförs med syfte att detektera slumpmässiga farliga fel i *Säkerhetsfunktion*. Händelsen *Fel i säkerhetsfunktion*

kan endast uppstå om det samtidigt är ett farligt fel i både *Säkerhetsfunktion* och att den oberoende övervakningen (datorsystem C) inte kan detektera det farliga felet. På samma sätt hanteras sedan *ställdon* och *givare* där syftet med den oberoende övervakningen är att slumpmässiga farliga fel ska detekteras innan de leder till att *Säkerhetskritiskt Datorsystemet A* utför en farlig styrning på grund av fel i *ställdon* eller *givare*.

En förutsättning är att den tillförda övervakningen i *Datorsystem C* kan anses vara oberoende av fel i det säkerhetskritiska *Datorsystemet A*, *B* och *D*.

Under händelsen *Fel i säkerhetskritisk funktion A* kommer i realiteten det *säkerhetskritiska Datorsystemet A* att bidra med den större delen av det nedbrutna delkravet. Det är i denna gren som systemets logik är realiserad och där den mest komplexa funktionaliteten finns i både hård- och programvara. Ur systemsäkerhetssynpunkt är därför målet att hålla det nedbrutna kravet på denna del så rimlig som möjligt för att underlätta verifieringen av kravet på vådahändelse A.

Tillförs övervakning för att detektera slumpmässiga farliga fel i ställdon och givare kan dessa två grenars bidrag till den sammanlagda felsannolikheten för farligt fel i den säkerhetskritiska funktionen reduceras ytterligare.

När alla vådahändelser för skadeklass I (katastrofal konsekvens för person, ekonomi och/eller miljö) har brutits ned på var sitt felträd, kan det dimensionerande felträdet identifieras. Möjliga realiserbara säkerhetsfunktioner har identifierats i samtliga felträd och en rimlig ansats har då gjorts på en verifierbar nivå på sannolikheten för slumpmässigt farligt fel i säkerhetsfunktionen. Det som blir kvar avseende det nedbrutna kravet på den säkerhetskritiska funktionen för det strängaste kravet blir sedan dimensionerande i utvecklingen av det *säkerhetskritiska Datorsystem A*.

Då *Datorsystem A* kan ingå i flera felträd för olika vådahändelser blir detta krav slutligen styrande för utvecklingen av *Datorsystem A*.

Eftersom både ställdon och givare ligger under ELLER-grinden måste deras bidrag till felsannolikheten också hållas lägre än kravet på *Fel i säkerhetskritisk funktion A*. Beroende på det ned-

brutna kravet på felsannolikhet för ställdon och givare så tillförs också krav på oberoende övervakning i *Datorsystem B* och *D*. På samma sätt hanteras också kravet på övervakning av säkerhetsfunktionen i *Datorsystem C*.

Observera att kraven enligt GKPS omfattar både krav på hårdvara för att reducera sannolikheten för slumpmässiga fel, samt krav på stringens i utvecklingsmetoder för programvaruutveckling i syfte att begränsa införandet av systematiska fel. Kravnedbrytningen av sannolikhet för vådahändelse är endast giltig för de slumpmässiga hårdvarufelen. GKPS kritikalitetsklassificering ***LÅG*** definierar den minsta delmängd av de krav som ska begränsa införandet av systematiska fel.

Typexempel finns i *bilaga 4*.

4.4 Kritikalitetsklassificering av det tekniska systemet

FMV har inför upphandling genomfört en initial kritikalitetsklassificering enligt avsnitt 4.1 av det tekniska systemet för en systemindivid. Om FMV:s *Functional Hazard Analyzis* (FHA) initiala kritikalitetsklassificering anger ***LÅG***, räcker det att ställa krav enligt Grundkrav Programvarusäkerhet (GKPS). Om FHA:s initiala kritikalitetsklassificering anger ***HÖG***, ska krav också ställas att valfri etablerad programvarustandard, tillämplig inom teknikområdet, ska tillämpas.

Utvecklande industri ska därefter ta fram ett koncept på säkerhetsarkitektur och utvecklingsprocess för att kunna balansera och reducera kravet på sannolikheten för vådahändelse. Den genomförda kravnedbrytningen är dock oberoende av vald programvarustandard.

För att Försvarsmaktens krav på tolerabel risknivå ska inrymmas, ska FMV och utvecklande industri komma överens om vilken kritikalitetsnivå som är nödvändig att uppnå för datorsystemet. Om kraven eller den valda systemlösningen är sådan att det nedbrutna kravet på vådahändelsen för farligt fel i säkerhetskritisk funktion kan ansättas en felsannolikhet som lägst 10^{-1}/system/år, då är enbart grundkraven (GKPS) tillräckliga. Om det nedbrutna

kravet istället är lägre än 10^{-1}/system/år, då ska också en valfri etablerad programvarustandard, tillämplig inom teknikområdet tillämpas.

Grundkraven GKPS innehåller krav på utformning av datorsystemet för att motverka slumpmässiga hårdvarufel samt krav på utvecklingsmetodik för att reducera införande av systematiska fel i datorsystemets hård- och programvara för kritikalitetsklassificering ***LÅG***. Den ansatta nivån för GKPS avseende felsannolikhet 10^{-1}/system/år är vald så att den ligger under kravet för SIL 1, enligt IEC 61508. Se *tabell 4:1* nedan.

Genom att införa flera oberoende säkerhetsfunktioner kan kraven på den säkerhetskritiska funktionen sänkas.

> Att införa flera redundanta oberoende säkerhetsfunktioner i syfte att enbart behöva tillämpa GKPS är inte tillåtet.

För ett tekniskt system i kontinuerlig drift avser som strängast en ansatt felsannolikhet på 10^{-1}/system/år (eller 10^{-5}/system/timme), vilket då motsvarar en sammanlagd drifttid av datorsystemet på 10 000 timmar, det vill säga cirka 1 år. Om en annan sammanlagd drifttid används så ska också felsannolikhetskravet räknas om enligt *tabell 4:1*. Nedan ges en omräkningstabell för lägsta ansatta felsannolikhet (10^{-5}/system/timme) beroende på sammanlagd drifttid.

Tabell 4:1 Omräkningstabell, tillämpning av GKPS för kontinuerlig drift

System i kontinuerlig drift Sammanlagd drifttid under livslängden	Lägsta tillåtna ansatta sannolikhet för fel i säkerhetskritisk funktion för kritikalitetsnivå ***LÅG***
≤ 100 h	1×10^{-3} (p)
< 500 h	5×10^{-3} (p)
< 1 000 h	1×10^{-2} (p)
< 5 000 h	5×10^{-2} (p)
< 10 000 h	1×10^{-1} (p) (1 års kontinuerlig drift = 8 760 h) (1 år ~ 10 000 h)
< 50 000 h	5×10^{-1} (p)
≥ 100 000 h	= 1

Om kravnedbrytningen ger en lägre felsannolikhet än 10^{-1}/system/år (10^{-5}/system/timme) enligt ovan angivna förutsättningar, måste grundkraven GKPS kompletteras med krav enligt överenskommen etablerad programvarustandard.

För en funktion i ett tekniskt system som påkallas vid behov (Demand mode), till exempel nödsystem, räddningssystem eller system med korta drifttider, kan ansatt felsannolikhet som lägst vara 10-1/system/år (se IEC61508, del 1 tabell 2). I detta fall är GKPS tillräckligt. För denna typ av system ska inte *avsnitt 4:1* tillämpas.

I *bild 4:12* finns kritikalitetsnivåer för olika programvarustandarder angivna. En direkt jämförelse mellan de olika standardernas kritikalitetsnivåer kan inte göras. I ett tekniskt system kan det finnas olika kritikalitetsnivåer på ingående system, samt även olika standarder som används under utvecklingsarbetet. Om detta föreligger, så måste FMV och industrin överenskomma hur en kritikalitetsmatris ska tillämpas för det aktuella projektet, med ett ställningstagande om hur olika använda standarder under utvecklingen förhåller sig till varandra.

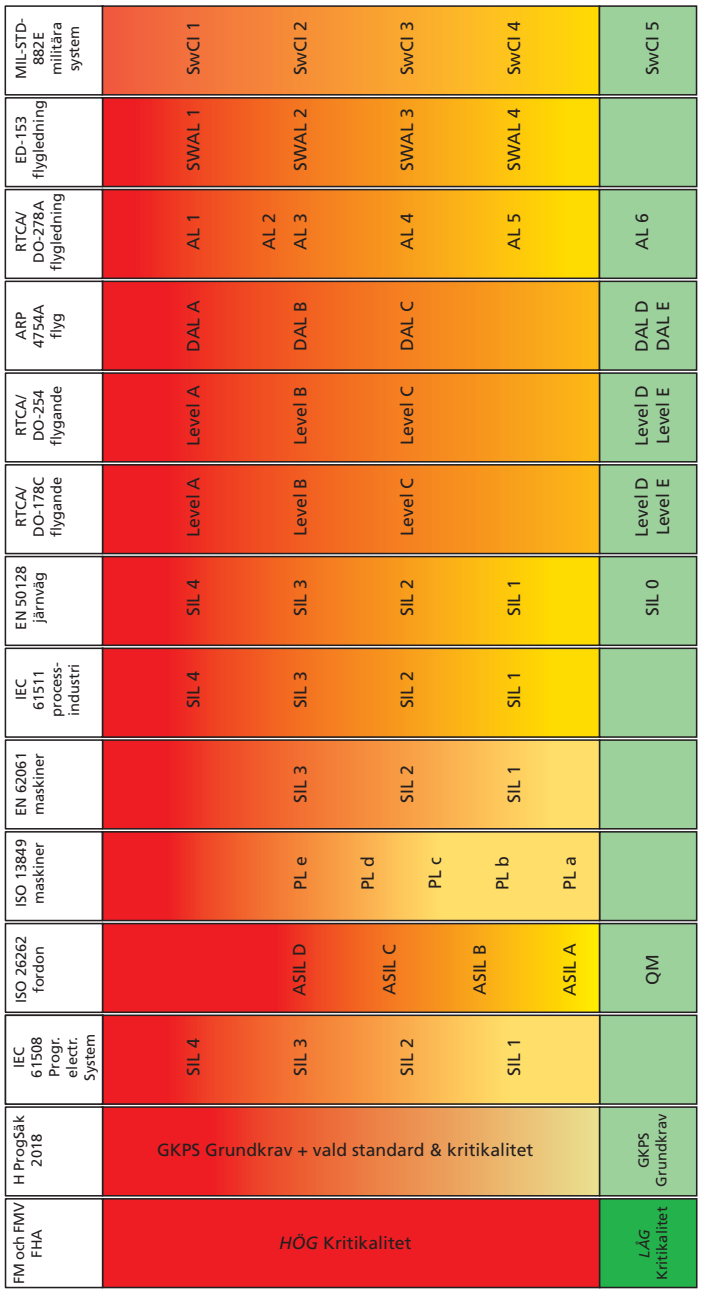


Bild 4:12 Kritikalitetsnivåer för olika programvarustandarder

En direkt jämförelse mellan de olika standardernas kritikalitetsnivåer kan inte göras.

4.5 Data

Med data avses här information, ofta lagrad som filer eller databaser, som programvaran använder då den ger funktion eller genererar annan information.

Bland typer av data märks:

- Omvärldsinformation: Till exempel terrängdata (kartor med mera), information om vägar (användning, egenskaper), information om luftrum och flygplatser.
- Kalibreringsdata: Till exempel motorvärden för ett motorreglersystem för att ge rätt funktion hos en motorindivid, information om accelerometrars kalibrering och orientering för ett navigeringssystem.
- Konfigurationsdata: Till exempel vilket utförande enheter som ingår i ett system har, och därmed vilka funktioner en programvara ska styra och på vilket sätt.
- Parametrar: Värden på parametrar som styr programvarans funktion, till exempel hur data från sensorer tolkas och hanteras, vilket kan variera för olika installationer av ett system.

Data kan både vara inflöde till de funktioner som datorsystemet eller programvaran har, och kan styra vilka funktioner datorsystemet (programvaran) ska ha.

Ovanstående resonemang innebär att data har inverkan på datorsystemets funktion och därmed dess säkerhet om datorsystemet är systemsäkerhetspåverkande. Man behöver försäkra sig om att data som kan påverka säkerhetsnivån är tillräckligt bra och har rätt kvalitet.

Inom flygområdet finns krav på kvalitetssäkring av *aeronautical data*, luftrumsdata med mera eftersom den typen av data genereras och hanteras av flera olika aktörer. Hur data kan kvalitetssäkras beskrivs i standarden RTCA DO-200B. I standarden ställs krav på de processer (inklusive verktyg med mera) som används

för att generera och hantera data. Utgångspunkten är att krav på data formuleras utifrån deras kritikalitet. Krav på data formuleras med avseende på följande egenskaper:

- noggrannhet (accuracy)
- upplösning (resolution)
- assuransnivå (assurance level)
- spårbarhet (traceability)
- aktualitet (timeliness)
- fullständighet (completeness)
- format (format).

Med assuransnivå avses vilken kravnivå som ställs på de arbetsprocesser som tillämpas för att skapa och hantera data. Har data hög säkerhetspåverkan ställs högre krav på vilka aktiviteter som ska genomföras och hur de ska dokumenteras och kvalitetssäkras. Krav ställs även på de tekniska system som hanterar data. Metodiken för att formulera krav på data kan vara till hjälp också inom andra teknikområden.

4.6 Underhållsutrustningar

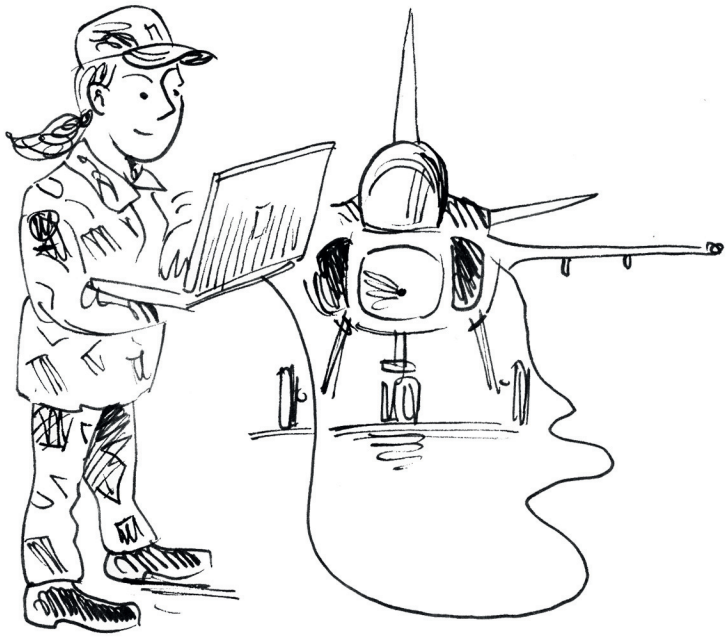
I det tekniska systemet ingår underhållsutrustningar och dessa ska analyseras i det ordinarie systemsäkerhetsarbetet. Ett väl konstruerat underhållskoncept underlättar både framtagning och vidmakthållande av ett tekniskt system.

Denna handbok ger särskild vägledning för externt ansluten utrustning som kan användas vid programvaruuppdatering, hantering av ändringsbara parametrar, programladdning av kod och data, utläsning av loggar samt felsökning.

All hantering av signalskyddsparametrar (krypto) behandlas utanför denna handbok.

Underhållsutrustning ska tas fram samtidigt med utvecklingen av det tekniska systemet eftersom anpassningar i gränsytor kan behöva konstrueras och anpassas. Underhållsutrustningen ska genom dess gränsytor till det tekniska systemet både kunna ge stimuli till, samt avläsa, testpunkter i systemfunktioner.

Vådahändelser som uppkommer vid användning av underhållsutrustning tillsammans med det tekniska systemet ska tas med i det inledande systemsäkerhetsarbetet. Vådahändelser kan även uppstå vid fel i, eller vid felaktig hantering av underhållsutrustning.



Underhållsutrustningen kan också meranvändas vid verifiering av det tekniska systemet då utrustningen kan interagera med systemfunktioner då systemet är i drift. Exempelvis kan utrustningen injicera fel för att verifiera att systemets ordinarie säkerhetsfunktion detekterar felet och utlöser eventuella fysiska skyddsfunktioner.

Underhållsutrustningen bör också kunna läsa ut alla systemloggar och spara dessa i en databas för senare analys av förekomst av feltyper och feltillstånd. För att kunna analysera loggar i efterhand måste dock en definierad systemtid finnas som på något sätt kan refereras till en känd tidbas, exempelvis *Coordinated Universal Time* (UTC-tid).

Om datorsystemet är kravställt med en så kallad reprisfunktion, exempelvis för återuppspelning av ett operatörsförlopp så bör registrerade fel i underhållsfunktionen direkt kunna kopplas mot reprisfunktionen.

För att underhållsutrustningen ska få kopplas in till, och därigenom få tillgång till informationsutbyte med, det tekniska systemet krävs att systemsäkerhetsanalys är genomförd och att systemsäkerhetsgodkännande är utfärdat för ändamålet.

Ändringar som görs i det tekniska systemet ska även loggas av både underhållsutrustningen och det tekniska systemet. Detta måste även omhändertas ut ett informationssäkerhetsperspektiv.

5 Livscykelhantering och kvalitetsstyrning

Programvara, inkluderat kod, data, dokumentation och utvecklingsmiljöer, måste hanteras i de olika delarna av det tekniska systemets livscykel såsom utveckling, systemuppdatering och avveckling. Det är viktigt att bevara utvecklingsmiljön och kompetens om projektet för framtida programvaruuppdateringar. Under avvecklingsfasen ska genomgång av programvarulicenser ske och eventuell utvecklingsmiljö avvecklas.

5.1 Verksamhetsledningssystem

Alla aktörer ska ha ett verksamhetsledningssystem för att kunna bedriva en kvalitetssäkrad verksamhet som kan vara baserat på en eller flera olika standarder. Nedan beskrivs ett urval av verksamhetsstandarder som särskilt belyser programvaruutveckling. Tillsammans utgör dessa tre standarder det som närmast kanske kan kallas en allmänt vedertagen praxis för hur programvaruutveckling kan beskrivas.

Standard ISO/IEC 15288 beskriver livscykelprocesser för system. För programvaruutveckling finns standard ISO/IEC 12207 som beskriver livscykelprocesser för programvara. Utvärderingar av processer inom informationsteknologi beskrivs av standard ISO/IEC 15504. Det som behandlas av standarderna är tillämpbart i många branscher.

De tre standarderna ISO/IEC 15288, ISO/IEC 12207 och ISO/IEC 15504 relaterar till varandra och beskrivs översiktligt nedan. Det finns möjlighet till oberoende certifieringar för olika verksamhetsledningssystem.

5.1.1 ISO/IEC 15288 Systems and software engineering - System life cycle processes

Standarden beskriver livscykelprocesser generellt för olika typer av tekniska system och är ett ramverk. I Annex B ges koppling till ISO/IEC 15504 Part 2. Syftet med standarden är att kunna göra utvärderingar av livscykelprocessen med stöd från ISO/IEC 15504. I Annex E ges en jämförelse mellan processer i ISO/IEC 15288 och i ISO/IEC 12207. ISO/IEC 15288 refererar till:

- ISO/IEC 15504 Part 2
- ISO/IEC 12207.

5.1.2 ISO/IEC 12207 System- och programvarukvalitet

Syftet med ISO/IEC 12207 är att vara en programvaruspecialisering av de generella livscykelprocesserna i ISO/IEC 15288. De två standarderna är harmoniserade med varandra så att de kan användas samtidigt. Nivån i ISO/IEC 12207 är relativt allmän och detaljer såsom specifika metoder och procedurer ingår inte. I Annex B ges koppling till ISO/IEC 15504 Part 2. Syftet är att hantera utvärdering (process assessment) med stöd från ISO/IEC 15504. Tabell B.2 i standarden listar samtliga processer.

I Annex D ges en jämförelse mellan processer i ISO/IEC 15288 och ISO/IEC 12207. ISO/IEC 12207 refererar till:

- ISO/IEC 15288
- ISO/IEC 15504 Part 2.

5.1.3 ISO/IEC 15504, Information Technology

Syftet med standarden är att kunna göra utvärderingar av livscykelprocessen. Standarden består av fem delar:

- Part 1: Concepts and vocabulary
- Part 2: Performing an assessment
- Part 3: Guidance on performing an assessment
- Part 4: Guidance on use for process improvement and process capability determination
- Part 5: An exemplar Process Assessment Model.

Part 1 innehåller definitioner. Part 2 innehåller krav för utvärdering, genomförande, hantering och klassificering (bland annat Level 1-5). Övriga delar finns som stöd. I Part 5 avsnitt 4.2.1. listas alla processer. ISO/IEC 15504 refererar till:

- ISO/IEC 15288 från Part 1-4
- ISO/IEC 12207 från Part 1-5.

5.2 Kvalitetsledning för försvarsmateriel

ISO 9001 är den vanligast förekommande standarden för kvalitetsledning. The Allied Quality Assurance Publications (AQAP) är standarder för kvalitetsledningssystem. Standarderna har utvecklats av Nato för kvalitetssäkring av försvarsmateriel och de kan användas av alla Nato-länder och deras samarbetspartner. Krav på att utvalda AQAP-standarder ska följas kan därmed ställas i kontrakt med utvecklande industri. AQAP-systemet beskrivs utförligt i STANAG 4107. Det finns för närvarande två huvudtyper av AQAP-standarder. Dels avtalsenliga som är skrivna som en teknisk specifikation, dels vägledande standarder. Om FMV i beställning har ställt krav på att leverantören ska följa AQAP 2110/2210 blir de tvingande.

5.2.1 AQAP 2110, NATO Quality Assurance Requirements for Design, Development and Production

Försvarsstandarden AQAP 2110 är särskilt utformad för leverantörer av militära tekniska system, produkter samt tjänster. AQAP 2110 innehåller Nato:s tilläggskrav utöver krav i ISO 9001 för kvalitetsledning vid konstruktion, utveckling och tillverkning. Krav på att AQAP 2110 ska följas är lämpligt att ställa om den utvecklande industrin redan uppfyller kraven i ISO 9001. Att ställa krav på att AQAP 2110 ska följas ger bland annat FMV en särskild insynsrätt i industrins arbete under projektets genomförande.

5.2.2 AQAP 2210, NATO Supplementary Software Quality Assurance Requirements to AQAP 2110

Försvarsstandarden AQAP 2210 är avsedd att användas som ett komplement till AQAP 2110 i projekt som även omfattar programvaruutveckling. AQAP 2210 innehåller särskilda krav på leverantörens kvalitetsledningssystem samt tillhörande krav för konfigurationsledning.

AQAP 2210 innehåller projektorienterade krav för att hantera kvaliteten på processen vid programvaruutveckling. Såväl administrativa och tekniska processer måste behandlas för att:

- upprätta synligheten för programvaruutvecklingsprocessen
- identifiera programvaruproblem så tidigt som möjligt i programvarans livscykel
- ge data till kvalitetskontroll för att snabbt genomföra effektiva korrigerande åtgärder
- bekräfta att kvalitet ingår under utvecklingsprocessen för programvara
- ge försäkran att programvaran som produceras uppfyller kontraktsenliga krav
- se till att lämpligt programvarustöd ges till aktiviteter på systemnivå i avtalet samt se till att ta itu med säkerhetskrav samt villkoren för projektet.

Utöver ovanstående kan även användning av ett internationellt samarbetsavtal inom kvalitetsområdet (GQA) användas. Se även AQAP 2070.

5.3 Konfigurationsledning (ISO 10007:2003, IDT)

Konfigurationsledning (Configuration Management, CM) är en metodik som tillämpar teknisk och administrativ styrning på konfigurationsobjekt (Configuration Item, CI) med dess konfigurationsinformation under ett tekniskt systems hela livcykel. Konfigurationsledning kan tillämpas för att uppfylla de krav på identifiering och spårbarhet som specificeras i ISO 9001. Information om konfigurationsledning finns i ISO 10007:2003.

Metodiken används för att upprätta, dokumentera och upprätthålla ett tekniskt systems fysiska och funktionella krav, prestanda, funktion samt fysiska komponenter med dess krav, design och operativ information. Valet av konfigurationsobjekt och deras inbördes samband baseras på beslutad systemdefinition. Fastställda kriterier bör användas när konfigurationsobjekt identifieras och kriterierna bör väljas så att deras funktionella och fysiska egenskaper kan hanteras separat i avsikt att konfigurationsobjektens totala prestanda i slutanvändningen uppnås.

Konfigurationsledning vid programvaruutveckling ska utgöra ett stöd för verksamheten och säkerställa att:

- Programvarans status och historik för ett tekniskt system dokumenteras under hela dess livscykel.
- Det finns en godkänd och låst struktur för programvaran där endast godkända ändringar tillåts genomföras.
- Det finns spårbarhet för alla händelser och beslut rörande allt som ingår i ett programvarusystem, såsom avvikelsehantering, problemrapporter och eventuell ändringsbegäran.

Konfigurationsinformation ska vara relevant, spårbar och uppdaterad.

5.4 Programvaruutvecklingsmiljöer

Under genomförandet av ett utvecklingsprojekt för programvara krävs både en kvalificerad utvecklingsmiljö och kompetens för att hantera densamma. Till verktyg hör även de utrustningar som krävs för verifiering av programvaran, såsom riggar, simulatorer, inklusive systemsimulatorer. Även utrustning för dataförsörjning och konfigurationsledning kan krävas. Det är viktigt att avtal finns med leverantören av utvecklingsmiljön, så att fel som upptäcks rapporteras och att rättningar kan tas fram. Vid ändringar i utvecklingsmiljön kan ny kvalificering erfordras när programvaran uppdateras. Detta regleras i använd utvecklingsstandard.

Inom ramen för FMV:s tekniska designansvar ingår även att skapa förutsättningar för uppkomna behov och planerade framtida systemuppdateringar i det tekniska systemet. FMV kan behöva kontraktera utvecklande industri för att på lämplig nivå vidmakthålla utvecklingsmiljön och kompetens under det tekniska systemets livslängd. Det kan till och med krävas lokaler för att ha utrustningen uppställd.

6 Förutsättningar från Försvarsmakten

Detta kapitel beskriver de förutsättningar som FMV behöver ha från Försvarsmakten och som erfordras för att uppnå tillräcklig systemsäkerhet i tekniska system. Försvarsmakten behöver ange användningsmiljö, operationsbetingelser, tolerabel risknivå och krav på administration vid förvaltning av det tekniska systemet. Vissa förutsättningar bör besvaras av Försvarsmakten innan ett anskaffningsuppdrag ges till FMV. Svar på förvaltningskrav ska finnas innan det tekniska systemet överlämnas till Försvarsmakten inför användning då detta kan påverka innehållet i FMV:s systemsäkerhetsgodkännande.

6.1 Förutsättningar inför utveckling av tekniska system

Handbok Systemsäkerhet (H SystSäk) beskriver systemsäkerhetsverksamheten för ett tekniskt systems livscykel hos de olika aktörerna. Det är Försvarsmakten som beslutar om tolerabel risknivå för tekniskt system. Ingångsvärde för systemsäkerhetskrav kan vara tidigare erfarenheter om systemets användnings- och omgivningsmiljö, samt de operationsbetingelser som gällde under utbildning, övning och insats. Genom lämplig arkitektur och motiverad kritikalitetsklassificering av datorsystem kan krav på tolerabel risknivå för det nya tekniska systemet uppfyllas.

Försvarsmakten ska definiera förbandets förmågor. Utifrån dessa behov utarbetar FMV en materielmålsättning för aktuellt tekniskt system, vilken fastställs av Försvarsmakten. Det förutsätter goda kontakter mellan alla involverade parter, inklusive slutanvändarna. Detta för att rätt tekniskt system ska kunna upphandlas och för att konstruktion, verifiering och validering samt driftsättning ska kunna utföras på ett kostnadseffektivt sätt.

För att överordnade behov på högsta systemnivå ska kunna uppfyllas, behövs en helhetsbild över sammanhang, användnings- och omgivningsmiljö, samt operationsbetingelser där det tekniska systemet ska användas och vilka uppgifter det ska utföra. Om det

tekniska systemet är avsett att användas i både militära användningsmiljöer och som stöd till samhället i fredstid så ska detta framgå av materielmålsättningen. Nedanstående förenklade exempel kan användas som modell för att beskriva funktionsinriktade prestandakrav på det tekniska systemet.

Förenklat exempel

Försvarsmakten behöver ett nytt luftvärnssystem. Systemet ska primärt användas under strid, men det ska även kunna stödja samhället i fredstid, till exempel under större evenemang om terroristhot bedöms föreligga. Försvarsmakten behöver besvara om luftvärnssystemet ska ge verkanseld mot alla luftfartyg eller om verkanseld endast får ske mot luftfartyg som med säkerhet bedöms som fientliga. Kritikalitetsklassificering av datorsystemet och säkerhetsarkitektur kommer att påverka kostnaden väsentligt vid utveckling av programvaran till luftvärnssystemet.

Systemsäkerhetskraven för datorsystemet formuleras med syfte att det slutliga tekniska systemet ska uppfylla Försvarsmaktens krav på tolerabel risknivå. Specificerad funktionalitet ska balanseras mot identifierade övergripande olycksrisker. Innan Försvarsmakten beställer utveckling av FMV ska korrekta och balanserade systemsäkerhetskrav finnas. Utifrån krav som Försvarsmakten ställer på tolerabel risknivå kan FMV bryta ner kraven på datorsystemet enligt modellen som beskrivs i *kapitel 4*.

FMV ska hos Försvarsmakten efterfråga driftserfarenheter från tidigare motsvarande tekniska system. FMV kan även medverka vid brukarmöten, eller att ta direktkontakt med brukare för att få en helhetsbild av möjliga olycksrisker.

FMV bör av Försvarsmakten efterfråga om det finns direktiv gällande hur programvarulicenser och underhållsavtal hanteras för att passa in mot annan inköpt programvara. Om Försvarsmakten redan har fleranvändarlicenser på en mängd programvaror, så är det bra om detta är känt vid upphandling av ny programvara. Eventuellt vill Försvarsmakten själva köpa in licenser och rättigheter.

Ett annat alternativ är att licenser ska ingå vid upphandlingen. Då det finns flera vägar att gå vad det gäller licenser är det viktigt att FMV klarlägger hur denna fråga bör lösas, så att licensfrågan inte blir onödigt krånglig eller kostnadsdrivande.

Nedanstående förutsättningar löper över materielens hela livscykel från behov till och med avveckling. FMV ska tillfråga beställare vid Försvarsmakten om inriktningar enligt nedan.

2.601.01-A	FMV skall begära att Försvarsmakten preciserar sammanhang, användnings- och omgivningsmiljö samt operationsbetingelser för det tekniska systemet. *Kommentar*: Detta gäller både för militär användning och i förekommande fall för stöd till samhället i fredstid.
2.601.02-A	FMV skall begära att Försvarsmakten definierar övergripande funktionsinriktade prestandakrav för tekniskt system.
2.601.03-A	FMV skall begära att Försvarsmakten definierar tolerabel risknivå för det tekniska systemet under hela dess livslängd.
2.601.04-A	FMV skall begära att Försvarsmakten tillgängliggör drifterfarenheter från tidigare motsvarande tekniska system.

6.2 Förutsättningar under utveckling av tekniska system

Under tiden som det tekniska systemet utvecklas, kan vissa olycksrisker identifieras som är svåra att nedbringa till tolerabel risknivå. Genom dialog mellan FMV och Försvarsmakten, inklusive medverkan av Försvarsmakten utsedd slutanvändare, kan dessa problem hanteras på ett tidigt stadium.

I god tid innan överlämning till Försvarsmakten behöver FMV få kännedom om vilken aktör som ska vara tekniskt designansvarig.

2.602.01-A	FMV skall av Försvarsmakten efterfråga vilken aktör som utses till att vara tekniskt designansvarig organisation. *Kommentar*: Om annan aktör än FMV ska vara tekniskt designansvarig behöver detta framgå av FMV:s Systemsäkerhetsgodkännande (SSG).

6.3 Förutsättningar inför överlämning och användning

Inför överlämning ska Försvarsmakten meddela FMV om hur rapportering och uppföljning av driftserfarenheter och avvikelser ska ske om inte ordinarie rapporteringssystem ska användas. Utöver detta behöver FMV veta hur Försvarsmakten avser införa systemuppdateringar på överlämnad materiel. Detta är särskilt viktigt att klargöra inför en eventuell insats, samt om det kommer att finnas andra inskränkningar som behöver omhändertas i FMV:s systemsäkerhetsgodkännande.

Systemuppdatering kan ske direkt av utvecklande industri. Det kan även ske genom att FMV utfärdar en Teknisk Order (TO) som reglerar vem som gör detta och hur systemuppdateringen ska gå till, inklusive kontrollinstruktion för att verifiera att uppdateringen blev korrekt utförd. Vid alla systemuppdateringar ska nya säkerhetsbeslut tas.

2.603.01-A	FMV skall begära att Försvarsmakten har ett avvikelserapporteringssystem för tekniska system där avvikelser kan rapporteras. *Kommentar*: Om annat avvikelserapporteringssystem än Försvarsmaktens ordinarie ska användas, behöver FMV ha kännedom om detta.
2.603.02-A	FMV skall begära att Försvarsmakten följer de anvisningar som överlämnas avseende handhavande, vidmakthållande/underhåll samt rutiner för att genomföra systemuppdateringar på överlämnad materiel. *Kommentar*: Om annan aktör än FMV ska vara Tekniskt designansvarig behöver Försvarsmakten delge FMV detta.
2.603.03-A	FMV skall utifrån Försvarsmaktens ställda krav specificera vilka inskränkningar och krav som gäller för personal som hanterar, vidmakthåller/underhåller eller genomför systemuppdateringar på överlämnad materiel. *Kommentar*: Detta gäller särskilt för insats då systemuppdatering kan behöva genomföras av Försvarsmaktens egen personal.

6.4 Förutsättningar för vidmakthållande

Vid användning och underhåll av tekniska system kan rapporter om avvikelser finnas. Uppföljning av dessa och förslag till åtgärder kan hanteras i Arbetsgrupp för systemsäkerhet (SSWG). Vid behov kan FMV begära deltagande av Försvarsmakten i SSWG för att gemensamt hitta förslag till lösningar.

2.604.01-A	FMV skall hos Försvarsmakten begära avvikelserapporter för det tekniska systemet. *Kommentar*: Informationen kan överlämnas till Arbetsgrupp för systemsäkerhet (SSWG).
2.604.02-A	FMV skall begära att Försvarsmakten deltar i Arbetsgrupp för systemsäkerhet (SSWG).

6.5 Förutsättningar inför avveckling

Försvarsmakten fattar beslut om avveckling. Beslutet omfattar tekniska system (eller delar av sådana) med ingående datorsystem med programvara. Beslutet ska även omfatta de resurser som används för stöd till utveckling och vidmakthållande av systemen såsom utvecklingsmiljöer och övriga stödsystem. Följande ingår i det som ska hanteras:

- Utvecklingsmiljö för programvaran såsom utvecklingsverktyg, riggar, simulatorer, lokaler, användarlicenser och uppdateringar för programvaran.
- Avtal för personella resurser som underhåller verktyg, riggar, simulatorer, konfigurationsledningsverktyg med mera.
- Resurser och avtal för dataförsörjning.
- Hemlig information i exempelvis dokumentation, hårddiskar och datorer.

Observera att utrustning och dokumentation kan finnas både hos Försvarsmakten, FMV och hos utvecklande industri.

7 VERKSAMHETSKRAV PÅ FMV

Detta kapitel innehåller krav och vägledningar för arbete inom FMV med arkitekturarbete, utformning av underlag för upphandling, uppföljning av industrins arbete och för vidmakthållande. Detta kapitel utgör därmed krav på FMV:s arbetssätt. Samverkan med Försvarsmakten beskrivs i ***kapitel 6***.

Utveckling av tekniska system med omfattande innehåll av programvara ställer krav på ett väl strukturerat arbetssätt med säkerhetsbefrämjande aktiviteter och tekniker för att undvika systematiska fel såsom felaktig kravställning med åtföljande stora kostnader för rättning. Kostnaderna för omkonstruktion av tekniskt system med säkerhetskritisk programvara tenderar att bli höga på grund av stora kostnader för nödvändig testning och dokumentation.

7.1 FMV:S ARBETE UNDER LIVSCYKELN

Verksamheten vid FMV ska planeras så att rätt systemsäkerhetsverksamhet, inklusive programvarusäkerhet, genomförs i samtliga livscykelskeden för respektive tekniskt system under koncept, utveckling, produktion, vidmakthållande och avveckling. FMV:s arbete ska ske i enlighet med gällande systemsäkerhetsledningsplan (SSMP) för systemet, eller systemsäkerhetsplanen för det enskilda projektet (SSPP) om sådan finns. SSMP och SSPP ska omfatta programvarusäkerhet. FMV genomför SSWG som ska behandla bland annat programvara i säkerhetskritiska tillämpningar. FMV begär deltagande av Försvarsmakten i SSWG för att gemensamt hitta förslag till lösningar, för att ge Försvarsmakten möjlighet att avge nödvändiga avdömningar och fatta beslut.

Arbetet på FMV med systemet och i det enskilda projektet ska omfattas av krav på kvalitetssäkring i enlighet med FMV:s interna arbetssätt. Se vidare H SystSäk.

2.701.01-A	FMV:s systemsäkerhetsledningsplan (SSMP) skall omhänderta krav på programvarusäkerhet. *Kommentar*: FMV:s SSMP ska omhänderta Försvarsmaktens krav på tolerabel risknivå för det tekniska systemets alla systemnivåer. I de fall FMV utarbetar en intern SSPP för ett projekt ska den även omfatta programvarusäkerhet.
2.701.02-A	Programvarusäkerhetsfrågor skall omhändertas av Arbetsgrupp systemsäkerhet (SSWG).

7.2 Konceptskede före Försvarsmaktens utvecklingsuppdrag till FMV

FMV och Försvarsmakten identifierar tillsammans vilka lösningar som finns i form av tekniska system och tjänster utifrån krav på efterfrågade förmågor på förband. FMV genomför utifrån det ett arkitekturarbete där tekniska system utformas som levererar rätt funktionalitet och uppfyller tillämpliga icke-funktionella krav samt uppfyller krav på tolerabel risknivå. I arkitekturarbetet identifierar och kravställer FMV vilka produkter (tekniska system eller produkter ingående i tekniska system) som ska anskaffas.

7.3 Utveckling, produktion och anskaffning

Utvecklande industri ska arbeta i enlighet med den obligatoriska systemsäkerhetsplanen (SSPP), som överenskommits med FMV vid kontraktsgenomgången. Industrins utvecklingsarbete ska även omfattas av ett kvalitetsarbete i överensstämmelse med standarderna AQAP 2110/2210 om inte annat överenskommits. Se vidare i *avsnitt 5.2*.

Vid kontraktsgenomgång mellan FMV och utvecklande industri ska ett protokoll upprättas. Av detta ska det framgå vilken programvarustandard och vilken kritikalitetsnivå som utvecklande industri kommer att följa vid utveckling av datorsystemet. Av protokollet ska det också framgå att industrin kommer att uppfylla GKPS (kraven i *kapitel 8*).

Industrins systemsäkerhetsplan (SSPP) ska utformas enligt H SystSäk och den ska även ange på vilket sätt kraven i *kapitel 8* kommer att uppfyllas. Vidare ska även en *Utvecklingsplan programvara* (SDP) bifogas. I de fall en programvarustandard krävs tillkommer ytterligare aktiviteter. Planerna ska omfatta aktiviteter under programvarans samtliga livscykelskeden, såsom kravhantering, konfigurationsledning, kodning, återanvändning, test och dokumentation. Planerna ska även omfatta hur de olika aktiviteterna följs upp, redovisas och levereras.

FMV ska säkerställa att industrin under utveckling använder ett system för avvikelserapportering där avvikelser i arbetsprocesser och avvikelser i förväntad funktion hos programvaran registreras och följs upp kontinuerligt (felrapporter, *problem reports*). Systemet för avvikelserapportering ska möjliggöra analys av såväl enskilda avvikelser som statistisk analys av den totala mängden avvikelser. Systemet ska också möjliggöra identifiering av programvarans konfigurationsstatus kopplat till respektive felrapport. Det kan ske med hjälp av ett konfigurationsledningssystem. Industrin använder de stödsystem, verktyg, som man normalt använder i sin verksamhet om de uppfyller de krav som ställs av FMV.

Industrin ska visa att eventuella fel genererade av utvecklingsmiljön kan upptäckas vid efterföljande testning.

FMV ska utfärda systemsäkerhetsgodkännande (SSG) för det kompletta tekniska systemet som överlämnas till Försvarsmakten. Detta förutsätter att FMV ser till att leverantörer av ingående system redovisar de systemsäkerhetsanalyser och riskreducerande åtgärder som omfattar programvara för de datorsystem som ingår. Det innebär också att FMV ska se till att leverantören för programvara som har kritikalitetsnivån ***LÅG*** enligt *avsnitt 4.1* visar överensstämmelse med grundkraven i denna handbok.

För programvara som har en påverkan på systemsäkerhet som ger kritikalitetsnivån ***HÖG*** ska industrin, förutom grundkraven (GKPS), även visa överensstämmelse med etablerad och med FMV överenskommen programvarustandard.

Oberoende granskning av det tekniska systemet ska genomföras av utvecklande industri i enlighet med den programvarustandard som tillämpas för utvecklingsarbetet och enligt H SystSäk för aktiviteter som återfinns där. Oberoende granskning definieras olika beroende på vilken programvarustandard som tillämpas, vilken uppgift som ska utföras och den kritikalitetsnivå med vilken man ska visa överensstämmelse. Den vanligaste innebörden är att dokumentgranskning eller annan aktivitet genomförs av någon som inte har deltagit i utvecklingen av programvaran inklusive dess dokumentation.

FMV kravställer genom beställning omfattningen av industrins arbete under vidmakthållande och drift. FMV ordnar så att industrin får tillgång till de data från användning i Försvarsmakten och data från eventuella andra intressenter som behövs för de analyser som ska göras.

Systemsäkerhetsarbetet ska säkerställa att systemet vid programvaruuppdatering fortsatt uppfyller av Försvarsmakten kravställd risknivå. Exempel på risker vid arbete på systemet kan vara att skyddsanordningar eller andra komponenter är bortmonterade då man vill provköra systemet vilket då kan utsätta personal för olycksrisker.

Då ett tekniskt system ändras ska ett förnyat systemsäkerhetsarbete genomföras. FMV identifierar berörda industrier som ges i uppgift att genomföra systemsäkerhetsarbete för respektive delområde för att ge underlag för ett förnyat systemsäkerhetsgodkännande (SSG) för hela systemet, i förekommande fall baserat på förnyade systemsäkerhetsutlåtanden (SCA) från industrin.

2.703.01-A	FMV skall säkerställa att SSPP omfattar programvarurelaterade systemsäkerhetsaktiviteter innan kontrakt tecknas. *Kommentar*: SSPP ska vara utformad i enlighet med H SystSäk och innehålla alla nödvändiga aktiviteter och metoder för att genomföra programvarusäkerhetsarbetet och i förekommande fall i enlighet med överenskommen programvarustandard.
2.703.02-A	FMV skall för programvara med initial kritikalitetsklassificering ***HÖG*** överenskomma med industrin om etablerad programvarustandard inklusive kritikalitetsnivå, tillämplig inom teknikområdet, med vilken industrin ska visa överensstämmelse. *Kommentar*: Vid kritikalitetsnivå ***LÅG*** räcker grundkraven (GKPS).
2.703.03-A	FMV skall säkerställa att det av protokollet från kontraktsgenomgången framgår vilka eventuella avsteg från GKPS som överenskommits. *Kommentar*: Av protokollet ska det framgå att industrin kommer att uppfylla övriga krav enligt GKPS. Flera kontraktsgenomgångar kan genomföras under tiden för genomförande av projektet.

2.703.04-A FMV skall säkerställa att industrin redovisar avvikelser med betydelse för systemsäkerhet som identifierats under utveckling och drift samt den totala mängden avvikelser.

Kommentar: Redovisningen ska vid leverans åtminstone omfatta vilka avvikelser som är öppna eller stängda från och med verifierande provning. FMV ska ha omhändertagit eventuella öppna anmärkningar i sitt systemsäkerhetsarbete åtminstone genom att ta ställning till att de inte föranleder åtgärder med avseende på systemsäkerhet.

2.703.05-A FMV skall kravställa att industrin visar överensstämmelse med grundkraven (GKPS) i denna handbok för alla programvaror oavsett kritikalitetsnivå.

2.703.06-A FMV skall säkerställa att industrin kan ge stöd för analys och åtgärder för uppkomna systemsäkerhetsproblem under systemets livslängd.

Kommentar: FMV ska beställa stöd från tillverkaren i enlighet med den omfattning och den tid FMV och Försvarsmakten har behov av. Hänsyn ska tas till det tekniska systemets egenskaper och förväntade livslängd.

2.703.07-A FMV:s och industrins systemsäkerhetsarbete inklusive programvarusäkerhet skall vara slutfört och systemsäkerhetsgodkännande (SSG) skall vara fastställt före överlämning till Försvarsmakten.

7.4 Användning och systemuppdateringar

Under användning och underhåll rapporterar användaren avvikelser i funktionen hos det tekniska systemet. Rapporterna analyseras av FMV och av respektive industri för att identifiera behov av åtgärder på grund av brister i funktion eller systemsäkerhet.

För att kunna analysera inträffade avvikelser behöver FMV ställa krav på att utvecklande industri har tillgång till utvecklingsmiljöer för testning. Detta inkluderar även system för hantering av avvikelserapportering. Omfattningen av industrins åtagande anpassas efter programvarans funktion och förväntade livslängd.

I de fall utvecklande industri har för avsikt att använda tidigare utvecklad programvara (PDS) från egen underleverantör så bör FMV ställa krav på utvecklande industri att genom särskilt underhållsavtal säkerställa tillgång till framtida systemuppdateringar.

Fastställd programvara kan behöva ändras. Ändringar ska behandlas på samma sätt som nyutveckling av programvara med den kritikalitetsnivå som den ändrade programvaran får. Ett systemsäkerhetsarbete ska genomföras och ett nytt Systemsäkerhetsgodkännande (SSG) baserat på Systemsäkerhetsutlåtande (SCA) ska utfärdas.

2.704.01-A	FMV skall säkerställa att industrin har tillgång till programvarans utvecklingsmiljö under produktens hela livscykel i den omfattning som behövs. *Kommentar*: Omfattning regleras av FMV beställning.
2.704.02-A	Uppdatering av systemsäkerhetsgodkännande (SSG) skall alltid göras vid förändring eller modifiering av ett tekniskt system. *Kommentar*: Se H SystSäk avseende Systemsäkerhetsgodkännande (SSG).

7.5 Avveckling av tekniskt system

Vid avveckling av det tekniska systemet ska programvarurelaterad verksamhet och materiel omhändertas. Exempel på detta är att bevarade utvecklingsmiljöer, eventuella licenser för programvara eller programvarubaserade stödsystem ska identifieras och avvecklas. För att möjliggöra avveckling ska förnödenheter som programvaror och datorer, även de som ingår i utvecklingsmiljöer och testmiljöer, registreras i relevanta förvaltningsstödsystem redan vid leverans från industrin.

2.705.01-A	Förnödenheter som programvaror och datorer skall registreras i relevanta stödsystem i samband med leverans från industrin. *Kommentar*: Detta gäller även förnödenheter som överförts i statens ägo, men som finns kvar hos industrin.
2.705.02-A	Avveckling av tekniskt system (eller del av tekniskt system) skall omfatta de resurser som används för stöd till utveckling och vidmakthållande av systemen. *Kommentar*: I det som ska hanteras ingår utvecklingsmiljöer för programvaran, avtal för verksamhet inklusive personal och dataförsörjning med mera

8 Grundkrav (GKPS) på utvecklande industrin

Detta kapitel innehåller de grundläggande kraven på programvarusäkerhet (GKPS), som ska uppfyllas av utvecklande industri vid utveckling och uppdatering av programvara i datorsystem oavsett kritikalitetsnivå.

8.1 Krav inför framtagning av tekniska system

Utveckling av programvara i ett säkerhetskritiskt datorsystem ställer krav på ett strukturerat arbetssätt och tekniker för att bland annat skapa robusthet och undvika systematiska fel i konstruktionen.

Syftet med att formulera systemsäkerhetskrav för programvaran är att det slutliga tekniska systemet ska uppfylla krav på tolerabel risknivå. Utvecklande industri ska för FMV redovisa resultatet av de systemsäkerhetsanalyser och riskreducerande åtgärder som vidtagits i det tekniska systemet. Det styrande dokumentet är den överenskomna Systemsäkerhetsplanen (SSPP).

För en bedömd initial kritikalitetsklassificering ***LÅG*** (enligt *bild 4:1*) ska samtliga nedanstående krav uppfyllas av utvecklande industri. För initial kritikalitetsklassificering ***HÖG*** tillämpas dessutom generella eller sektorsspecifika etablerade programvarustandarder. Dock ska utvecklande industri alltid skriftligt redovisa för FMV hur kraven enligt detta kapitel kommer att uppfyllas. Detta kan görs i Systemsäkerhetsplanen (SSPP) eller i P*lan for Software Aspects of Certification* (PSAC)/*Plan for Software Aspects of Approval* (PSAA).

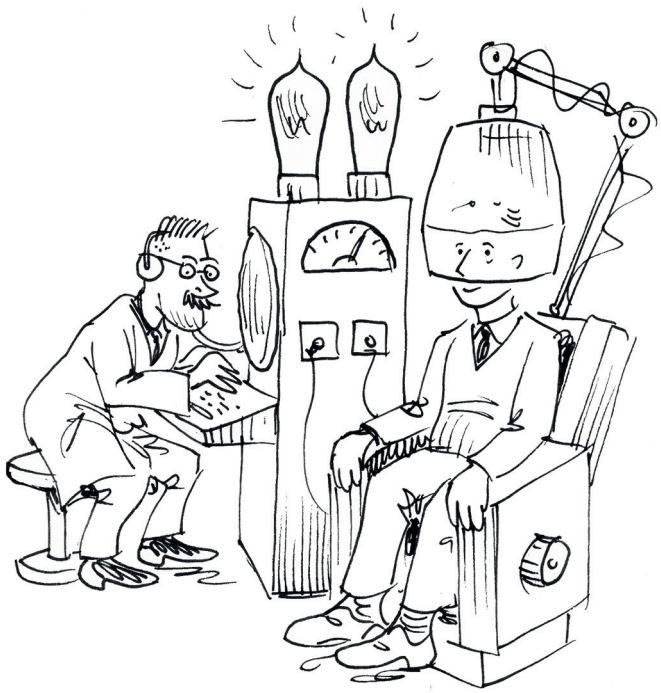
Om utvecklande industri i sin preliminära kritikalitetsklassificering kan visa att de olycksrisker som kan påverkas av det tekniska systemets programvara har låg eller negligerbar konsekvens för person, ekonomi och/eller miljö är grundkraven (GKPS) i detta kapitel tillräckliga att uppfylla.

Grundkraven (GKPS) ger förutsättningar för att uppnå kravställd tolerabel risknivå. Detta leder till att industrin under utvecklingen av det tekniska systemet identifierar och åtgärdar systematiska fel. Därmed kan utvecklings- och vidmakthållandekostnaderna reduceras under hela livscykeln, samtidigt som kravställd tolerabel risknivå upprätthålls över livstiden.

8.1.1 Krav på kompetens hos personal

Personal som utvecklar datorsystem och programvara ska ha god kännedom om etablerad utvecklingsteknik, säkerhetsarkitektur, metoder, verktyg och programmeringsspråk, samt ha kunskap och erfarenhet av tillämpliga programvarustandarder inom teknikområdet för liknande tekniska system. För initial kritikalitetsklassificering ***HÖG*** tillkommer utökade kompetenskrav enligt vald programvarustandard.

2.801.01-A	Roller inklusive erforderlig kompetensnivå skall överenskommas med FMV. *Kommentar*: Kompetensprofil för personal som medverkar i utvecklingen av det tekniska systemet såsom projektledare, tekniskt ansvarig för systemarkitektur, verifieringsansvarig samt oberoende granskare dokumenteras.
2.801.02-A	Minst två personer skall ha kännedom om motiv för vald systemarkitektur. *Kommentar*: Valet av systemarkitektur utifrån genomförd kravnedbrytning av dimensionerade vådahändelser ska vara känd av minst två personer.
2.801.03-A	Utvecklande industri skall utse en kontaktperson för programvarusäkerhet. *Kommentar*: Denna person säkerställer att för projektet överenskommet arbetssätt och metodik för systemsäkerhetsarbetet följs samt ansvarar för verifieringen av grundkraven (GKPS) och redovisar att dessa krav är uppfyllda.



8.1.2 Krav på verksamhets- och systemsäkerhetsledning

Industrins verksamhetsledningssystem syftar till att bedriva en kvalitetssäkrad verksamhet. Utvecklande industri ska tillämpa AQAP 2110/ 2210 så att FMV ges särskild insynsrätt i utvecklingsarbetet av det tekniska systemet. I kontraktet mellan FMV och industrin överenskoms om vilka delar som ska tillämpas i standarderna.

Industrins systemsäkerhetsarbete vid utveckling styrs av överenskommen systemsäkerhetsplan (SSPP), se *kapitel 9*. Denna plan ska godkännas av FMV före projektstart. I förekommande fall kan programvaruutvecklingen beskrivas i en programutvecklingsplan *Software Development Plan* (SDP) eller i PSAC/PSAA.

Omfattningen av *Systemsäkerhetsplanen* (SSPP) eller programutvecklingsplanen *Software Development Plan* (SDP) beror på det tekniska systemets komplexitet och kan behöva omprövas under projekttiden beroende på förändrade förutsättningar.

I det fall där säkerhetsarkitekturen kräver initial kritikalitetsklassificering ***HÖG***, ska utvecklande industri redovisa en anpassning mot överenskommen etablerad programvarustandard. Detta kan hanteras i *Software Development Plan* (SDP), se *kapitel 9*.

2.801.04-A	Industrin skall följa AQAP 2110. *Kommentar*: Detta gäller främst insynsrätten.
2.801.05-A	Industrin skall följa AQAP 2210.
2.801.06-A	Industrin skall ta fram en Systemsäkerhetsplan (SSPP). *Kommentar*: Systemsäkerhetsplanen (SSPP) ska även omfatta erforderliga aktiviteter såsom kravdokument, testplaner och testprocedurer för programvaruutveckling, samt en beskrivning av överenskomna utvecklingsverktyg.
2.801.07-A	Utvecklande industri skall i Systemsäkerhetsplanen (SSPP) redovisa hur GKPS kommer att uppfyllas.
2.801.08-A	Systemsäkerhetsanalys skall omfatta datorsystemets påverkan på det tekniska systemets hela livscykel. *Kommentar*: Analysen ska utföras iterativt under utvecklingen, från kravnedbrytning till avslutad verifiering.

8.1.3 Krav på utformning av säkerhetsarkitektur

Vid konstruktion av säkerhetskritiska datorsystem är det viktigt att utgå från det tekniska systemets mest kritiska vådahändelser för skadeklass I, i H SystSäk, och låta dessa påverka utformning av säkerhetsarkitekturen. Om skadeklass I inte kan inträffa används istället skadeklass II.

Målet är att tidigt i systemutvecklingen kunna fastställa en säkerhetsarkitektur som kan ge förutsättningar för att erhålla en så låg kritikalitetsnivå för datorsystemet och ingående programvara som möjligt. Se metodik i *avsnitt 4.1–4.4*.

Identifiering av möjliga vådahändelser hos det tekniska systemet är ett arbete som påbörjas tidigt och pågår under hela utvecklingstiden. Innan val görs av säkerhetsarkitektur ska denna vägas mot erforderliga systemsäkerhets- och prestandakrav. Som säkerhetsprincip ska enkelhet i konstruktionen eftersträvas. Konstruktionsprinciperna ska redovisa vilka strategier för felupptäckt, feltolerans och felsäkerhet som ska tillämpas. Redovisningen ska även omfatta verifieringsmetoder och godkännandekriterier.

Under utvecklingsarbetet kan nya vådahändelser identifieras och en omprövning av vald säkerhetsarkitektur kan därför bli nödvändig. Vid långa utvecklings- och vidmakthållandetider är det därför viktigt att dokumentera konstruktionsbesluten så att en omprövning inte blir personberoende.

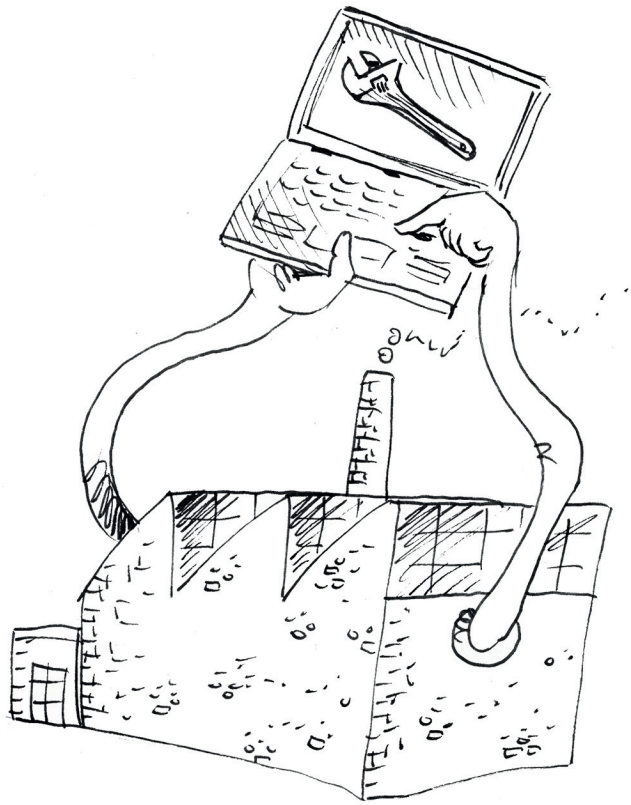
2.801.09-A	För datorsystemet skall säkerhetsarkitektur och konstruktionsprinciper dokumenteras och redovisas. *Kommentar*: Industrin ska presentera en säkerhetsarkitektur enligt avsnitt 4.3 vilken redovisas i Systemspecifikation /System, Subsystem Specification (SSS).
2.801.10-A	Konstruktionsprinciperna skall fastlägga vilka strategier för felupptäckt, feltolerans och felsäkerhet som tillämpas. *Kommentar*: Redogörelsen ska ange valda konstruktionsprinciper med motiveringar för gjorda val.
2.801.11-A	Konstruktionsbeslut avseende vald säkerhetsarkitektur skall dokumenteras och inkludera förutsättningar, antaganden samt motiveringar för valda konstruktionsalternativ.

8.1.4 Krav på utvecklingsverktyg

Val av utvecklingsverktyg ska överenskommas med FMV innan systemsäkerhetsplanen (SSPP) fastställs. De överenskomna verktygen för kravspårning, konfigurationsledning, avvikelserapportering och testdata ska fungera i en verktygskedja och vara utformade på ett sådant sätt att information kan utbytas mellan FMV och utvecklande industri. Detaljer kring detta ska framgå i avtalet mellan FMV och utvecklande industri.

Under utveckling är insyn och transparens viktigt för både FMV och utvecklande industri. Detta medför en ökad delaktighet och ger förutsättningar för att kunna göra korrekta och gemensamma prioriteringar av åtgärder under utveckling och vidmakthållandet av det tekniska systemet.

2.801.12-A	Verktyg för kravspårning skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för kravspårning enligt IEC 12207.
2.801.13-A	Verktyg för konfigurationshantering skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för konfigurationshantering enligt IEC 12207.
2.801.14-A	Verktyg för avvikelserapportering skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för felrapportering enligt IEC 12207.
2.801.15-A	FMV skall ges tillgång till information för kravspårning, konfigurationsledning, avvikelserapportering och testdata. *Kommentar*: FMV behöver se till att förutsättningar finns för att kunna hantera och läsa informationen.



8.1.5 Krav på dokumentation

Under utvecklingsarbetet tar utvecklande industri fram en stor mängd dokumentation. De dokument som ska ingå vid leverans av komplett tekniskt system, ska överenskommas med FMV. Detta ska finnas dokumenterat i systemsäkerhetsplanen (SSPP), vilket är ett obligatoriskt dokument. Dokumentationen kan även bestå av automatgenererade rapporter från utvecklingsverktygen, men innehållet ska omfatta kravställd information. Förslag på dokumentlista finns i *kapitel 9*.

En digital plattform för leverans av dokumentation behöver överenskommas mellan FMV och utvecklande industri. Plattformen bör också utgöra ett stöd för konfigurationsledning av tillhörande dokumentation under det tekniska systemets livstid.

Beakta att tiden för utveckling och vidmakthållande kan vara så lång att informationstillgångarna måste vara oberoende av vald plattform. Hänsyn måste också tas till det tekniska systemets säkerhetsskyddsplan och informationssäkerhetsklassificering vid val av plattform.

2.801.16-A En dokumentlista skall överenskommas med FMV.

Kommentar: Definieras utifrån dokumentlistan i *kapitel 9*. Leveransplan för dokumentation ska finnas.

8.2 Verksamhetskrav för utveckling av tekniska system

Under utvecklingen måste viktiga konstruktionsbeslut överenskommas med FMV. Tidigare säkerhetsanalyser kan behöva omprövas allt eftersom utvecklingen fortskrider. Det är därför viktigt att utvecklande industri använder ett system där avvikelser registreras löpande och att FMV har tillgång till sådan information för att kunna följa utvecklingsarbetet. Detta ska överenskommas i kontraktet mellan FMV och utvecklande industri.

I de fall där avvikelserapporter indikerar att en omkonstruktion eller ändring i arbetssätt är nödvändig så ska detta överenskommas och protokollföras med FMV vid formella granskningsmöten där alla relevanta parter är representerade.

8.2.1 Krav på systemsäkerhetsanalys

Systemsäkerhetsanalyser inleds på övergripande systemnivå och bryts ned successivt för varje delsystem enligt kravställd systemsäkerhetsmetodik.

Under det inledande systemsäkerhetsarbetet ska en säkerhetsarkitektur tas fram för att om möjligt kunna sänka kritikalitetsnivån för de ingående delsystemen.

De fördjupade analyserna av konstruktionen ska ge underlag till förbättringar i säkerhetsarkitekturen så att kritikalitetsnivån kan hållas på ansatt nivå eller till lägre nivå.

Detaljering leder till ökad insikt om vilka delar, samt i vilken grad dessa delar påverkar de identifierade vådahändelserna. Olika metoder för systemsäkerhetsanalys kan tillämpas för att utreda var och hur programvaran kan vara involverad och som i förlängningen kan leda till olyckor. Exempelvis kan felträdsanalys (FTA) användas.

Oberoende granskning ska genomföras av identifierade dimensionerade vådahändelserna, samt hur dessa också har styrt valet av säkerhetsarkitekturen. Den person som utför denna oberoende granskning ska inte ha medverkat i utvecklingsarbetet, se även H SystSäk.

Data som hanteras av det säkerhetskritiska datorsystemet ska också analyseras och ha den kritikalitetsklassificering som krävs i det aktuella tekniska systemet. Data ska klassas på samma grunder som för övrig programvara, det vill säga klassas utifrån den effekt den kan ha på felhändelserna. Se *avsnitt 4.5*.

2.802.01-A	Spårbarhet skall finnas mellan datorsystem och dess inverkan på det tekniska systemets identifierade vådahändelser. *Kommentar*: Kravspårning åt båda håll kan redovisas i säkerhetsarkitekturarbetet.
2.802.02-A	Systemsäkerhetsanalysen skall redovisa kritikalitetsnivån för ingående programvaror i det tekniska systemet. *Kommentar*: Avser analys av säkerhetsarkitektur.
2.802.03-A	Val av säkerhetsarkitektur skall motiveras utifrån analys av de dimensionerade vådahändelserna.
2.802.04-A	Oberoende granskningar och genomgångar skall utföras under utvecklingen enligt överenskommen systemsäkerhetsplan (SSPP). *Kommentar*: Med oberoende granskare avses person som inte medverkat i utvecklingsarbetet.

2.802.05-A Data skall ha den kritikalitetsklassificering som krävs i det aktuella tekniska systemet.
Kommentar: Med data avses både statisk och realtidsgenererad information.

8.2.2 Krav på konstruktion

Vid utveckling av programvara i ett säkerhetskritiskt datorsystem finns det många krav och aspekter att ta hänsyn till. Nedanstående utgör en minsta mängd av de detaljkrav som ska omhändertas av utvecklande industri för tekniska system med initial kritikalitetsklassificering ***LÅG***.

Val av säkerhetsfunktioner och funktionsövervakningar ska göras på sådant sätt att detta inte i onödan komplicerar programvarusystemet. En avvägning bör alltid göras så att de centrala säkerhetsprinciperna såsom enkelhet, oberoende och determinism uppnås. Detta underlättar förståelse av programvarusystemets uppbyggnad och verifiering för framtida uppdateringar.

Det ska finnas ett oberoende mellan kritiska och icke-kritiska delar. Ett fullständigt oberoende kan i praktiken dock vara svårt att uppnå och de beroenden som identifierats ska dokumenteras med en bedömning på dess inverkan.

De delar av det tekniska systemet som styr eller indirekt styr riskkällorna ska vid fel kunna kommenderas till ett säkert tillstånd eller med hjälp av externa säkerhetsfunktioner kunna inta ett säkert tillstånd. I det säkra tillståndet kan räddningssystem ha övertagit styrningen av riskkällorna. Notera att ett räddningssystem i sig inte har något säkert tillstånd.

Testtäckningsgrad för inbyggd test (BIT) anger hur stor del av hårdvarufunktionerna eller hårdvarans möjliga felfunktioner i datorsystemet som den inbyggda testen BIT klarar av att hitta. BIT är en programvara som körs samtidigt som övrig operationell programvara i datorsystemet.

En viktig funktion är loggning av systemets interna tillstånd och händelser för att kunna spåra felorsak till eventuella felyttringar i programvarusystemet. Loggning bör kunna utföras med olika detaljeringsgrad i syfte att kunna lokalisera felorsak.

I SDP ska de metoder som kan användas för att påvisa Proven in use vara definierade. Motiven och kriterierna för att kunna använda sig av Proven in use för de använda fallen ska finnas dokumenterat.

2.802.06-T Val av säkerhetsfunktioner och funktionsövervakningar skall göras på sådant sätt att detta inte i onödan komplicerar programvarusystemet.

Kommentar: En avvägning bör alltid göras så att de centrala säkerhetsprinciperna såsom enkelhet, oberoende och determinism uppnås.

2.802.07-T Etablerat programspråk skall användas vid utveckling av säkerhetskritisk programvara.

Kommentar: Valda programmeringsspråk ska redovisas för FMV tillsammans med konstruktionsprinciper och säkerhetsarkitektur.

2.802.08-T För varje operativt tillstånd skall det tekniska systemet kunna inta ett säkert tillstånd.

Kommentar: För initial kritikalitetsklassificering ***LÅG*** avses säkert tillstånd där styrningar av verkställande delar har kommenderats/avslutats på ett säkert sätt eller där ett räddningssystem har övertagit kontrollen.

2.802.09-T Alla feltillstånd som kan påverka systemets funktion skall loggas i ett utvärderingsbart format.

Kommentar: Spårbarhet ska finnas mellan utlösande felsituation/felkriterium och det tillstånd det tekniska systemet intagit så att fel kan hittas i datorsystemet. Loggning kan ske internt i datorsystemet eller loggas i externt system.

2.802.10-T Det tekniska systemet skall vara i ett säkert tillstånd under uppstart.

Kommentar: Detta omfattar även omstart av datorsystemet.

2.802.11-T Vid uppstart av det tekniska systemet skall programvaran kontrollera att definierat säkert tillstånd har intagits innan kritiska delar aktiveras.

Kommentar: Säkerhetsnivån kan kontrolleras med hjälp av återläsning på kritiska styr- eller givarsignaler.

2.802.12-T Orimliga indata, som enligt systemsäkerhetsanalysen kan påverka systemets funktion, skall detekteras och omhändertas så att vådahändelse inte inträffar.

Kommentar: Med orimliga data menas alla data utanför definierat värdeområde eller data vid fel tidpunkt.

2.802.13-T Operatörsåtgärder och presenterad information, som berör säkerhetskritiska funktioner, skall registreras.

Kommentar: Valet av sätt att registrera kan variera utifrån systemkonfiguration, komplexitet och situation.

2.802.14-T Built In Test (BIT) skall innehålla Säkerhetskontroll (SK/PBIT) under uppstart, Funktionsövervakning (FÖ/CBIT) under drift och Manuellt initierad test/Funktionskontroll (FK/IBIT) vid underhåll.

2.802.15-T BIT-funktioner för uppstart och underhåll skall inte kunna aktiveras oavsiktligt under operativ drift av systemet.

Kommentar: Säkerhetsfunktion, exempelvis blockering, ska finnas så att handhavandefel kan undvikas.

2.802.16-T Oberoende watchdog-funktion skall aktiveras innan datorsystemet kan genomföra kritiska styrningar.¨

Kommentar: Oberoende watchdog-funktion realiseras lämpligen i hårdvara.

2.802.17-T Watchdog (WD) skall ha ett definierat tidsfönster (det vill säga min/max tid för WD-trigg).

Kommentar: Återställning av watchdog utförs av programvaran.

2.802.18-T Watchdog (WD) skall omfattas av Säkerhetskontroll (SK/PBIT) vid uppstart och godkänt resultat skall utgöra kriterium för att kunna aktivera watchdog-funktionen.

2.802.19-T Spänningsövervakning skall ske kontinuerligt på datorsystemets matningsspänningar.

Kommentar: Kontrollsignal från spänningsövervakningen kan utgöra ett av kriterierna i watchdog-funktionen.

2.802.20-T Resursutnyttjande vid första serieleverans skall vara definierat.

Kommentar: Kravet avser CPU, minne och kommunikationslänkar och bör vara högst 50%.

8.2.3 Krav på programvaruutvecklingsmiljö

För utvecklingsprojekt som löper under lång tid kommer använda utvecklingsverktyg att uppdateras och förändras. En fungerande konfigurationsledning under hela det tekniska systemets livscykel är därför viktigt.

Då programvaruutvecklingsmiljön uppdateras ska omverifiering ske av den utvecklade programvaran så att inga oavsiktliga förändringar har skett i funktionen. Om det finns en regressionstestmiljö kan denna användas som delverifiering på att den uppdaterade utvecklingsmiljön inte har medfört oönskad påverkan.

Testverktyg som inför ändringar i källkoden och som är nödvändiga för att kunna genomföra en verifiering i målmiljön, ska ha sådana egenskaper att ändringarna bibehålls i källkoden efter genomförd verifiering. Analys av eventuell påverkan från testverktyget ska genomföras som underlag för att påvisa att verifieringen är giltig.

2.802.21-A Val av programvaruutvecklingsmiljö skall motiveras och dokumenteras för det tekniska systemet.

Kommentar: Branschstandarder och tidigare erfarenheter ska beaktas utifrån vald kritikalitetsnivå.

2.802.22-A Revisionshistorik skall redovisas för använd utvecklingsmiljö.

Kommentar: Utvecklingsmiljön ska konfigurationsstyras under programvarans hela livscykel.

2.802.23-A Vid uppdatering av utvecklingsmiljö under utvecklingen av programvaran skall omverifiering ske av både utvecklingsmiljön och utvecklad programvara.

Kommentar: Tillvägagångssätt och kriterier beskrivs i SSPP eller i annat överenskommet dokument.

2.802.24-A Testverktyg som inför ändringar i programvaran skall inte användas vid verifiering av en fastställd programvaruversion.

Kommentar: Om modifieringar är nödvändig för att testverktyget ska kunna användas så ska dessa ändringar ses som del av programversionen.

8.2.4 Krav på verifiering

Systemsäkerhetsprovning ska genomföras vid verifiering av programvara som ingår i ett säkerhetskritiskt datorsystem. Denna verksamhet utgör del av arbetet med att fullständigt verifiera och validera det tekniska systemet. Avsikten är att verifiera att införda säkerhetsfunktioner och övervakningar fungerar på avsett sätt i

målmiljön och att de kan detektera fel innan dessa kan förorsaka en felsituation som kan leda till en vådahändelse. Systemsäkerhetsprovning ska även omfatta felaktigt handhavande av det tekniska systemet och inkludera alla användningsfall såsom utbildning och underhåll.

Testtäckning av kod är att verifiera hur stor del av krav och kod som är implementerade och genomlöpta av testsekvenser.

De använda testfallen ska också genomgå en oberoende granskning av en person som inte har medverkat i utvecklingen av det tekniska systemet.

Systemsäkerhetsprovning ska utföras med den fastställda systemversionen. Med fastställd systemversion avses den version av det tekniska system som ska levereras till FMV. Detta innebär även att den tänkta målmiljön ska ha slutgiltig status. I de fall oklarheter råder kring målmiljön kan FMV behöva precisera denna. Alternativt kan industrin dokumentera vilka antagande som är gjorda avseende målmiljön.

Resultatet av systemsäkerhetsprovningen ska visa att åtkomst till funktioner endast avsedda för visst driftläge/systemtillstånd inte kan nås under annat driftläge/systemtillstånd. Alla tänkta driftlägen såsom normal användning, underhåll och utbildning ska provas. Av dokumentationen ska det framgå vilka funktioner som finns tillgängliga, respektive är spärrade, i alla olika driftlägen.

Anpassningar och speciella testanordningar (testboxar) kan vara nödvändiga för att felinjicering/felsimulering ska kunna göras i målsystemets ordinarie gränsytor. Framtagning av testanordning bör samordnas med utveckling av underhållsfunktioner för det tekniska systemet, se vidare *avsnitt 4.6*.

I samband med systemsäkerhetsprovningen ska även maximalt resursutnyttjande av datorsystemet verifieras och dokumenteras. Syftet med detta är dels att säkerställa att såväl CPU, minne och länkar har tillräcklig kapacitet vid normal drift, dels för att möjliggöra framtida funktionstillväxt. En tumregel kan vara att marginalen vid maximalt resursutnyttjande ligger runt 50% vid första systemleverans med full funktionalitet.

Säkerhetskritiska funktioner bör vara deterministiska, det vill säga att de exekveras i en bestämd förutsägbar ordning. Denna egenskap bör verifieras under systemsäkerhetsprovningen.

För säkerhetskritisk programvara så kommer med stor sannolikhet flera uppdateringar att ske under det tekniska systemets livslängd. Planerar man för detta under utvecklingen och även för hur omverifiering av säkerhetsfunktioner ska genomföras, så säkerställs kontrollen på både systemsäkerhetskraven och kostnaderna för omverifieringen. Om testsekvenser och utvärdering automatiseras (regressionstestning) så kan tiden för omverifiering reduceras.

Om tidigare utvecklad programvara PDS ska användas i det tekniska systemet och om verifiering av denna funktionalitet ska stödja sig på tidigare erfarenheter (*Proven in use*) så ska kriterierna för detta finnas dokumenterade i SDP (*Software Development Plan*).

2.802.25-A	Systemsäkerhetsprovning skall planeras, utföras och granskas samt brister som upptäcks skall åtgärdas samt godkännas. *Kommentar*: Resultatet presenteras och eventuella identifierade åtgärder överenskoms med FMV.
2.802.26-A	Testfall vid systemsäkerhetsprovning skall genomgå en oberoende granskning av person som inte har medverkat i utvecklingen.
2.802.27-A	Systemsäkerhetsprovning skall utföras på en fastställd systemversion i det tekniska systemet. *Kommentar*: Med fastställd systemversion avses den version av det tekniska system som ska levereras, det vill säga även målmiljön ska ha fastställd status.

2.802.28-A Systemsäkerhetsprovning skall omfatta felinjicering i samtliga gränsytor av de i systemsäkerhetsanalyserna identifierade säkerhetskritiska signalerna.

Kommentar: Systemsäkerhetsprovningens avsikt är att visa att funktionsövervakningen kan detektera kritiska fel.

2.802.29-A Systemsäkerhetsprovning skall visa att funktioner endast avsedda för visst driftläge/systemtillstånd inte kan nås under annat driftläge/systemtillstånd.

Kommentar: Beakta även felaktigt handhavande och drifttillstånd såsom utbildning och underhåll.

2.802.30-A Maximalt resursutnyttjande av datorsystemet skall verifieras och dokumenteras.

Kommentar: Kravet avser CPU, minne och kommunikationslänkar.

2.802.31-A Verifiering skall ske av att exekvering sker i rätt ordning och vid rätt tidpunkt i tidskritiska funktioner.

Kommentar: Verifiering av exekveringsordning kan även utföras med hjälp av utvecklingsmiljön.

2.802.32-A Testtäckning (BIT) av säkerhetsfunktioner i det tekniska systemet skall verifieras.

2.802.33-A Använda kriterier för *Proven in use* skall vara överenskomna med FMV.

Kommentar: Kriterierna dokumenteras i SDP (Software Development Plan).

8.3 Krav inför leverans till FMV

Inför varje leverans av ett nytt tekniskt system, eller en ny uppdaterad konfiguration av det tekniska systemet, ska den överenskomna leveransaktiviteten enligt systemsäkerhetsplanen (SSPP) följas. Detta innefattar att utvecklande industri ska utfärda ett systemsäkerhetsutlåtande med erforderliga bilagor enligt H Syst-Säk. För tekniska system som innehåller datorsystem kan dessa bilagor bland annat utgöras av dokumentlistan i *kapitel 9*.

Då utvecklande industri alltid ska uppfylla och bekräfta att *Grundkrav Programvarusäkerhet* (GKPS) enligt kontraktet är uppfyllda så ska detta framgå av systemsäkerhetsutlåtandet eller av någon bilaga. Om det finns överenskomna undantag från GKPS så ska även detta anges med hänvisning till protokoll från kontraktsgenomgången. Om utvecklande industri har valt att följa en etablerad programvarustandard redovisas lämpligen denna kravuppfyllnad genom att hänvisa till programvarustandardens olika element där det finns överensstämmelse mellan GKPS och krav i standarden.

Utvecklande industri ska alltid vid slutleverans styrka att alla tidigare kända fel i programvaran är omhändertagna och åtgärdade. Om det ändå skulle finnas kvarvarande kända fel i programvaran så ska industrin redovisa detta i *Software Version Description* (SVD). Vidare ska industrin i systemsäkerhetsutlåtandet (SCA) redovisa att kravställd tolerabel risknivå innehålls trots kvarvarande kända fel. FMV kan därvid godkänna eller avvisa leveransen utifrån detta.

Om utvecklande industri har använt en särskild programvara enbart för att kunna genomföra provning för verifiering av kravuppfyllnad och denna programvara inte krävs för operationell drift så får denna inte finnas med vid leverans.

2.803.01-A	Lista över kvarstående kända fel skall redovisas för levererad version av det tekniska systemet. *Kommentar*: Anges i Software Version Description (SVD) enligt dokumentlista

2.803.02-A	Industrin skall, trots kvarvarande kända fel, visa att det tekniska systemet ändå uppfyller Försvarsmaktens krav på tolerabel risknivå.

8.4 Krav vid systemuppdatering

Systemuppdateringar av programvara kan initieras dels av utvecklande industri utifrån ett produktansvar, dels av FMV på uppdrag av Försvarsmakten. FMV ska först rådgöra med Försvarsmakten angående införandet. Alla systemuppdateringar ut till förband initieras av FMV genom att en Teknisk Order (TO) ges ut. Utvecklande industri tar fram ett nytt systemsäkerhetsutlåtande samt rådgör med FMV om hur systemuppdateringar ska administreras. Om ändringen enbart avser ändringsbara parametrar erfordras inget nytt systemsäkerhetsutlåtande. Dessa parametrars inverkan på det tekniska systemet ska vara analyserade, verifierade och angivna i systemsäkerhetsutlåtandet för systemversionen.

2.804.01-A	Vid ny version av det tekniska systemet skall omverifiering genomföras. *Kommentar*: Omverifieringsbehovet avgörs efter analys av vilka delar som är påverkade av ändringen.
2.804.02-A	I samband med systemuppdatering skall ett nytt systemsäkerhetsutlåtande ges ut. *Kommentar*: Avser inte ändringsbara parametrar.

8.5 Krav vid avveckling av resurser hos utvecklande industri

Utvecklande industri har produktansvar enligt Produktansvarslagen för produkter och tekniska system i 10 år efter att den enskilda produkten sattes på marknaden. Industrin ansvarar själv för att bibehålla erforderlig teknisk kompetens om produkterna samt för att spara tillräcklig dokumentation om produkten om en säkerhetsbrist skulle upptäckas och behöva utredas. Längre tider för produktansvar kan avtalas mellan FMV och industrin.

Så länge som Försvarsmakten och FMV använder de av industrin levererade tekniska systemen bör programvarurelaterad verksamhet och materiel finnas kvar vid industrin. Detta inkluderar bland annat driftloggar, utvecklingsmiljöer, testmiljöer samt programvarubaserade stödsystem. FMV och utvecklande industri kan reglera detta genom avtal enligt *kapitel 7*.

9 Beskrivning av dokumentation

Utvecklande industri följer dokumentlistan som anges i vald standard med hänsyn till överenskommen kritikalitetsnivå. Om grundkraven (GKPS) i ***kapitel 8*** bedöms vara tillräckliga att uppfylla för det tekniska systemet används dokumentlistan nedan. Information som ska presenteras enligt dokumentlistan ska alltid kunna redovisas för FMV oberoende av vald programvarustandard. Programvarudokumentationen utgör en nödvändighet för att kunna vidmakthålla ett tekniskt system.

9.1 Dokumentlista för grundkraven (GKPS)

I de fall utvecklande industri tar fram ett tekniskt system med kritikalitetsklassificering *LÅG* och där grundkraven anses vara tillräckliga att uppfylla kan nedanstående dokumentlista följas. Om motsvarande dokument tas fram inom ramen för systemsäkerhetsarbetet kan informationen lämpligen presenteras samlat där. Om det tekniska systemet i huvudsak består av programvara kan dock information kring programvarusäkerhet i vissa dokument nedan vinna på att särredovisas från annat systemsäkerhetsarbete.

Dokumentlistan nedan utgör en minsta delmängd för att kunna redovisa underlag för analys och genomförande av systemsäkerhetsverksamhet avseende det säkerhetskritiska datorsystemet (SSHA Datorsystem).

Dokumentlistan ska ses som ett stöd vid val av dokumentstruktur och fördelning av innehåll. Kvaliteten i informationen som ska presenteras om programvarusäkerhet är viktigare än att alla olika dokument tas fram. Utvecklande industri kan därför föreslå sammanslagning av vissa dokument och överenskomma detta med FMV i Systemsäkerhetsplanen (SSPP).

Motsvarande benämning på olika dokument som förekommer i dokumentlistan finns i många olika standarder. Dokumenten i de olika standarderna har liknande syften och innehåller motsvarande rubrikstrukturer även om benämningar kan variera. Rubriksättning får dock ändras och överenskommas med FMV.

Tabell 9:1 Exempel på dokumentlista för Grundkrav (GKPS) i kronologisk ordning

Exempel på dokumentlista för Grundkrav (GKPS)		
Akronym	Benämning	Beskrivning
SSPP	Systemsäkerhetsplan/System Safety Program Plan	Särskilda aktiviteter för utveckling av datorsystem infogas i projektets SSPP, se H SystSäk.
PSAC	Certifieringsplan för programvara/Plan for Software Aspects of Certification	Om systemet ska certifieras av en myndighet ska en PSAC tas fram och redovisas för certifieringsmyndigheten innan projektet påbörjas.
PSAA	Acceptansplan för programvara/Plan for Software Aspects of Approval	En PSAA bör tas fram för att klargöra kriterierna för acceptans och leverans av systemet innan projektet påbörjas.
SDP	Utvecklingsplan programvara/Software Development Plan	Beskriver hur programvaruframtagningen ska genomföras.
SCMP	Konfigurationsledningsplan programvara/Software Configuration Management Plan	Beskriver hur konfigurationsledningen av programvaran ska utföras och till vilken detaljnivå.
SVP	Verifieringsplan programvara/Software Verification Plan	Beskriver teststrategin och hur verifieringen ska genomföras.
SVR	Verifieringsrapport för programvara/Software Verification Report	Sammanfattning av resultatet från genomförd verifiering enligt SVP.
SQA Plan	Kvalitetsplan för programvara/Plan Software Quality Assurance Plan	Kvalitetsorganisation och kvalitetsmålen.
SQA Record	Kvalitetssäkringsrapport programvara/Records Software Quality Assurance Records	Rapport kvalitetsmål.
SSS	Systemspecifikation/System, Subsystem Specification	Specificering av systemsäkerhetskraven för det tekniska systemet.
SRS	Specifikation Programvarukrav/Software Requirement Specification	Specificering av systemkrav som ska realiseras i programvaran.
IRS	Gränsytespecifikation/Interface Requirement Specification	Specificering av de elektriska gränsytorna och programvarugränsytorna.

Exempel på dokumentlista för Grundkrav (GKPS)		
Akronym	Benämning	Beskrivning
SDD	Detaljerad design programvara/ Software Design Document	Specifikation av programvarukomponent med koppling till överliggande krav.
STD	Provprogram för programvara/ Software Test Description	Specificerar på detaljnivå hur respektive test ska genomföras och i vilken miljö.
STR	Testrapport programvara/ Software Test Report	Testrapport från STD.
SVD	Versionsbeskrivning av levererad systemversion/Software Version Description Document	Beskriver den aktuella systemreleasens status avseende funktion och konfiguration (både avseende SW samt FW/HW och avvikelser).
SSTD	Provprogram för systemsäkerhetsprovning/System Safety Test Description	Provprogram för systemsäkerhetsprovning.
SSTR	Systemsäkerhets-provningsrapport/System Safety Test Record	Provrapport systemsäkerhetsprovning.
SSHA CS	Systemsäkerhetsanalys Datorsystem/Sub System Hazard Analysis Computer System	SHA för datorsystemet i det tekniska systemet Ingår som underliggande dokument i SHA för hela det tekniska systemet, se H SystSäk.

9.2 Beskrivning av särskilda dokument

Information om vad som bör redovisas och hur det kan dokumenteras i olika dokument finns att läsa om i olika standarder (en relativt heltäckande sammanställning av olika dokument finns i ISO 15289). Nedan beskrivs övergripande syftet med respektive dokumentet i dokumentlistan och krav på principiellt innehåll. Dokumenten utgör endast minimikrav på innehåll ur systemsäkerhetssynvinkel och kan samordnas med den ordinära dokumentstrukturen för utvecklingen.

9.2.1 Systemsäkerhetsplan/System Safety Program Plan, SSPP

Syftet med *Systemsäkerhetsplan* (SSPP) är att beskriva planerade systemsäkerhetsaktiviteter. Denna aktivitet genomförs projektsammanhållande oavsett systemnivå. För tekniska system, där utvecklande industri har systemansvaret, ska SSPP godkännas av FMV innan ingående aktiviteter genomförs av industrin. I de fall FMV är systemsammanhållande gäller denna aktivitet även för FMV.

Vid kontraktsgenomgång mellan FMV och utvecklande industri ska ett protokoll upprättas. Av detta ska det framgå vilken programvarustandard och vilken kritikalitetsnivå som utvecklande industri kommer att följa vid utveckling av datorsystemet. Av protokollet ska det också framgå att industrin kommer att uppfylla GKPS (kraven i *kapitel 8*).

Industrins systemsäkerhetsplan (SSPP) ska utformas enligt H SystSäk och den ska även ange på vilket sätt kraven i *kapitel 8* kommer att uppfyllas. SSPP ska omfatta aktiviteter under programvarans samtliga livscykelskeden. Planen ska även omfatta hur de olika aktiviteterna följs upp, redovisas och levereras.

SSPP används för att utvärdera en potentiell industris förståelse för och prioritering av den systemsäkerhetsverksamhet som erfordras vid utveckling av ett tekniskt system. I detta fall avses ett datorsystem i säkerhetskritiska tillämpningar. Mer information om SSPP finns i H SystSäk. Programvarurelaterade systemsäkerhetsaktiviteter ska hanteras i SSPP.

I SSPP bör man komma överens om viktiga principer, såsom hur överenskommen programvarustandard ska tillämpas avseende hantering av redundans och diversitet.

9.2.2 Certifieringsplan för programvara/Plan for Software Aspects of Certification

Om det tekniska systemet ska certifieras av en myndighet ska en *Plan for Software Aspects of Certification* (PSAC) tas fram och redovisas för och godkännas av certifieringsmyndigheten innan projektet påbörjas. Dokumentet PSAC ska visa att tänkt livscykel för programvaran i datorsystemet följer det regelverk som krävs för den kritikalitetsnivå som programvaran är avsedd att uppfylla. PSAC är ett dokument definierat i DO-178. Det finns andra motsvarande dokument som till exempel *Plan for Software Aspects of Approval* (PSAA), definierat enligt DO-278.

Dokumentet PSAC bör innehålla en beskrivning av det tekniska systemet i stort och en beskrivning av programvaran med dess respektive funktioner, vilka funktioner som ligger i hårdvara respektive programvara. Beskriv tänkt uppdelning av programvaran och hur man arbetat med och säkerställer att säkerhetskraven kommer att uppfyllas.

Dokumentet bör även beskriva hur man avser att genomföra arbetet under programvarans olika livscykelfaser. De data som behövs för programvaran ska beskrivas hur de tas fram och handhas.

Vidare ska dokumentet beskriva hur man tänker visa för certifieringsmyndigheten hur man tänkt arbeta så att certifieringsmyndigheten får den insyn som krävs för att det tekniska systemet ska kunna certifieras. Dessutom bör man redovisa olika aspekter som kan påverka certifieringen, hur man tänkt anpassa (tailoring) sitt arbete såsom mot valda processer, hur utvecklingsmiljön är kvalificerad, hur man tänker hantera *Tidigare utvecklad programvara* (PDS), deaktiverad kod, samt laddning av programvara och data i målsystemet.

9.2.3 Utvecklingsplan programvara/Software Development Plan, SDP

Syftet med *Software Development Plan* (SDP) är att beskriva hur utvecklingen av programvara ska genomföras i det aktuella projektet och planen ska överenskommas med FMV. Om vald säkerhetsarkitektur kräver kritikalitetsnivå ***HÖG***, ska SDP också redovisa en anpassning mot överenskommen programvarusäkerhetsstandard och vald kritikalitetsnivå.

SDP ska innehålla följande:

- Identifikation och systemöversikt
- Projektorganisation och resurser
- Dokumentöversikt och koppling till andra dokument
- Referenser
- Använda processer och metoder för programvaruutveckling
  Här anges vilka standarder och kodningsföreskrifter som utvecklingen ska följa samt vilka verktyg och programvaruprodukter och standardbibliotek som ska användas.
- Konfigurationsledning (CM)
  Översiktlig redovisning av CM-planen, hur grundversion (baseline) för programvara definieras, återanvändbara komponenter (PDS) samt hur avvikelsehantering och hur korrigerande åtgärder i programvaran hanteras.
  Här redovisas också beslutsprocess för ändringshanteringen. Detaljer i CM-planen redovisas i Software Configuration Management Plan (SCMP).
- Kravhantering
  Beskriver hur spårbarheten hanteras avseende krav och verifiering av krav. Översiktlig identifiering av hur kravtaggar är definierade samt hur systemsäkerhetskraven identifieras. Beskriv arbetssätt och kravhanteringsverktyg.

- Utvecklings- och testmiljö

  Redovisning av utvecklingsmiljön samt hur testmiljön är uppbyggd och i vilka steg test av programvaran sker, samt hur återkoppling till kravhanteringen sker. Beskriv hur utvecklings- och testmiljön godkänts för användning.

  Om regressionstestning används ska principen för hur godkännandekriterierna fastställs i regressionstesten samt hur dessa godkännandekriterier kan testas.
- Granskningar

  Principer för hur granskning sker av programvara, test och godkännandekriterier samt vilka granskningssteg som redovisas mot FMV.
- Systemintegration

  Redovisning av hur integration och test genomförs i den skarpa målmiljön samt vilka krav som ska verifieras på den skarpa målmiljön.
- Använda verktyg

  Specificering av använda kravspårningsverktyg samt hur kravtaggar är uppbyggda.
- Säkerhetsarkitektur

  Redovisning av säkerhetsarkitektur.
- Använd programvarustandard

  Hänvisning till vilken programvarustandard som används och anpassningar mot denna.
- Leveransprocess programvara

  Redovisning av processen inför leverans av en ny programvarurelease till FMV.
- Vidmakthållandeprocess programvara

  Redovisning av process för vidmakthållandet av levererad programvara och ändringshantering.

9.2.4 Konfigurationsledningsplan programvara/Software Configuration Management Plan, SCMP

Syftet med *Software Configuration Management Plan* (SCMP) är att redovisa hur konfigurationsledning av programvaran ska utföras och till vilken detaljnivå det görs. Detta för att man vid varje tillfälle under programvarans livscykel ska kunna identifiera ingående delar, samt vid behov kunna återskapa en specifik version av programvaran. *Software Configuration Management Plan* (SCMP) ska överenskommas med FMV.

SCMP ska innehålla följande:

- Identifikation och systemöversikt
- Konfigurationsledningsorganisation och resurser
- Dokumentöversikt och koppling till andra dokument
- Referenser
- Använda verktyg

  Specificering av använda verktyg för utvecklingsmiljö, samt versions- och ärendehantering och på vilka format informationstillgångarna kan exporteras.

  Beskriva hur konfigurationsledningen av verktygen ska genomföras.
- Konfigurationsstruktur

  Specificering av konfigurationsledningsobjekt.
- Konfigurationsstatus

  Definition av vilka mätetal för programframtagningen som ska redovisas för FMV och när. Exempel på mätetal kan vara:
  - Antalet åtgärdade problemrapporter
  - Problemtyp (specifikationsfel eller kodningsfel)
  - Tidsåtgång och tidpunkt för åtgärdande av problem
  - Antalet kvarstående problemrapporter.
- Konfigurationsrevision

  Definition av hur revision av konfigurationsledning ska genomföras.

- Problemrapporter

  Hur felrapporter hanteras, klassas och hur felorsak identifieras.
- Ändringshantering

  Process för hur åtgärder, utifrån identifierade felorsaker eller förändrade krav, ska införas i kommande systemversioner utifrån en definierad grundversion (*base-line*).
- Systemversion av programvara

  Ange stegen i processen inför en systemversion och hur denna skapas utifrån en definierad *base-line*:
  - Omfattning av Test Readiness Review (TRR) och i vilka processteg denna ska utföras.
  - Identitet och dokumentstruktur för aktuell systemversion.
  - Använda verktyg och standardkomponenter för systemversionen.
- Leveransprocess

  Ange på vilket format leverans sker till FMV samt hur installationsprocessen och hur aktiviteter som *Factory Acceptance Test* (FAT)/*Site Acceptance Test* (SAT) ska genomföras och säkerställas på målobjektet.

9.2.5 Verifieringsplan programvara/Software Verification Plan, SVP

Syftet med *Software Verification Plan* (SVP) är att redovisa teststrategin och hur verifieringen ska genomföras.

SVP ska innehålla följande:

- Testorganisation
- Krav på bedömt oberoende mellan utveckling och testning
- Identifiering av testmiljö med tillhörande versioner
- Testmetoder som avses användas
- Testmetoder för säkerhetskritiska funktioner
- Testmetoder för säkerhetsfunktioner
- Hur in- och utdata från testfallen ska registreras och göras spårbara

- Godkännandekriterier för testfallen
- Spårbarhet till krav.

Verifieringsplanen ska även redovisa i vilka delar som testutrustningen kan påverka testresultatet och en bedömning av konsekvenserna. Detaljer avseende testsekvenser och godkännandekriterier kan redovisas i *Software Test Description* (STD).

9.2.6 Verifieringsrapport för programvara/Software Verification Report, SVR

Syftet med *Software Verification Report*, SVR är att sammanfatta resultatet från genomförd verifiering enligt *Software Verification Plan* (SVP).

I SVR ska alla krav från SRS vara samlade och resultatet ska vara spårbart ned till utförd test. Detaljer från genomförda tester ska redovisas i *Software Test Record* (STR).

Även brister med tillhörande problemrapport ska redovisas i SVR.

9.2.7 Kvalitetsplan för programvara /Software Quality Assurance Plan (SQA)

Syftet med *Software Quality Assurance Plan* (SQA) är att redovisa organisation och mål för kvalitetsarbetet. SQA ska även beskriva hur programvarukvaliteten ska säkerställas, vilka mätetal som ska användas och hur uppföljning och verifiering av dessa ska redovisas, se AQAP 2110/2210.

9.2.8 Kvalitetssäkringsrapport programvara/Software Quality Assurance Records (SQAR)

Syftet med *Software Quality Assurance Records* (SQAR) är att redovisa resultat från genomförda kvalitetsaktiviteter enligt fastställd S*oftware Quality Assurance Plan* (SQA).

9.2.9 System-, delsystemspecifikation/System, Subsystem Specification (SSS)

Syftet med *System/Subsystem Specification* (SSS) är att specificera alla kraven för det tekniska systemet inklusive systemsäkerhetskrav.

9.2.10 Specifikation Programvarukrav/Software Requirement Specification (SRS)

Syftet med *Software Requirement Specification* (SRS) är att specificera de systemkrav som ska realiseras i programvara med möjlighet till kravspårning gentemot systemkrav. Även härledda krav ska specificeras.

9.2.11 Gränsytespecifikation/Interface Requirement Specification (IRS)

Syftet med *Interface Requirement Specification* (IRS) är att specificera gränsytorna till det tekniska systemet. Särskilt fokus är att säkerhetskritiska signaler är identifierade och vilka test- och avläsningspunkter det finns för dessa.

9.2.12 Detaljerad design programvara/Software Design Document (SDD)

Syftet med *Software Design Document* (SDD) är att för respektive programvarukomponent specificera de implementerade funktionerna med koppling till krav från SRS. *Software Design Document* (SDD) är den lägsta detaljnivå på specifikation för programvaran. Omfattning och djup ska vara överenskommet med FMV.

SDD ska innehålla följande:

- Säkerhetskritiska funktioner och säkerhetsfunktioner ska vara speciellt markerade med tillhörande felhantering samt vara spårbara till systemkrav.
- Alla interna och externa gränsytor ska vara specificerade.
- Koppling till återanvända komponenter (PDS) samt standardbibliotek ska anges.
- Säkerhetskritisk data ska beskrivas.
- Övergripande arkitektur, övergripande principer för exekvering och datautbyte samt andra konstruktionsregler bör redovisas.

9.2.13 Provprogram för programvara/Software Test Description (STD)

Syftet med Software Test Description (STD) är att på detaljnivå specificera hur respektive testfall ska genomföras. STD kan ingå i SVP för mindre system men det rekommenderas att hålla dessa dokument separerade.

Alla systemkrav och härledda systemkrav ska vara spårade till testfall med definierade godkännandekriterier.

Testning av säkerhetskritiska funktioner och säkerhetsfunktion i programvaran ska vara tydligt beskriven.

Redovisning av vilka krav som omfattas av automatiska och/eller manuella tester.

9.2.14 Testrapport programvara/Software Test Report (STR)

Syftet med *Software Test Report* (STR) är att redovisa genomförda tester enligt *Software Test Description* (STD) ovan.

STR ska innehålla följande:

- Sammanfattning

  Kortfattad sammanfattning av resultatet från det genomförd provverksamhet.

  Om avvikelser har identifierats under provningen ska dessa anges i sammanfattningen med tillhörande referens till problemrapport.
- Identifikation och systemöversikt
- Dokumentöversikt och koppling till andra dokument
- Referenser
- Provobjekt och systemversion

  Kortfattat beskrivning av provobjektets konfiguration och status med tillhörande referenser och eventuella förändringar mot fastställt provprogram.
- Provningsresurser

  Ange var och när provningen utfördes och med vilken personal provningen genomfördes.
- Testutrustning

  Redovisa använd testutrustning. All testutrustning ska vara registrerad med tillhörande konstruktionsunderlag för ingående hårdvara och programvara.
- Sammanfattning av testresultat

  Detaljerad sammanfattning av antalet godkända respektive icke godkända tester.

  De icke godkända testerna ska vara identifierade med kravbenämning samt tillhörande problemrapport för att kunna analysera avvikelsen.

- Testomfattning
  Redovisning av testtäckning avseende systemkrav inklusive härledda systemkrav samt vilka krav som omfattas av automatiska och/eller manuella tester.
- Kravspårning
  Sammanställning av kravspårningsmatris med krav och koppling till vilken deltest som verifierar att kravet är uppfyllt.
- Loggdata och testresultat
  Spårbarhet till loggdata och testresultat ska finnas. Alla testresultat ska vara sparade i ett sådant format att en granskning av enskild testning, inklusive resultat, i efterhand ska vara möjlig att på nytt genomföra med en rimlig insats.

9.2.15 Versionsbeskrivning av levererad systemversion/Software Version Description Document, SVD

Syftet med *Software Version Description Document* (SVD) är att beskriva den aktuella programvaruversionens status avseende funktion mot systemkraven och konfiguration. Vidare anges vilka nya funktioner som tillkommit samt vilka avvikelser i programvaran som blivit åtgärdade sedan föregående programvaruversion.

Kvarvarande kända avvikelser ska redovisas med referens till tillhörande problemrapport samt eventuella restriktioner på grund av dessa.

SVD ska innehålla följande:

- Identifikation
- Dokumentöversikt och koppling till andra dokument
- Referenser
- Programvaruversion
  Beskrivning i tabellform av ingående komponenter med versionsnummer för den aktuella programvaruversionen. Ingående PDS ska specificeras med ingående version och checksumma.

- Införda ändringar

  Beskrivning av ny och förändrad funktionalitet samt åtgärdade fel sedan föregående levererad programvaruversion. Alla ändringar ska vara spårade mot krav och/eller problemrapport tillsammans med dokumentation som visar genomförd analys av ändring samt test av ändringen.
- Kvarvarande kända fel

  Kvarvarande kända fel ska vara spårade mot angiven problemrapport och redovisas i minst tre huvudgrupper:
  - Fel med systemsäkerhetspåverkan och eventuellt tillkommande restriktioner i användning eller underhåll.
  - Fel med funktionspåverkan och eventuellt tillkommande instruktioner vid användning och underhåll.
  - Övriga kända fel eller funktionsstörningar.

9.2.16 Provprogram för systemsäkerhetsprovning/System Safety Test Description (SSTD)

Syftet med *System Safety Test Description* (SSTD) är att beskriva hur verifiering av säkerhetsfunktioner i det tekniska systemet ska genomföras och vara spårbara mot systemkraven. Fel i systemet ska kunna detekteras innan dessa leder till en situation som inte kan kontrolleras av programvarusystemet.

Systemsäkerhetsprovning ska genomföras på komplett målsystem efter det att systemversionen är låst. Sker någon förändring på systemversionen, det vill säga hårdvara och/eller programvara efter genomfört systemsäkerhetsprov, ska omprov ske. Provet genomförs lämpligen tillsammans med FMV.

SSTD ska innehålla följande:

- Sammanfattning
- Identifikation

- Dokumentöversikt och koppling till andra dokument
- Referenser
- Personal

  Krav på oberoende från utvecklings- och testteam.
- Provobjekt och programvaruversion

  Kortfattad beskrivning av provobjektets konfiguration och status med tillhörande referenser. Det viktigaste är att tydligt klargöra tillåtna avvikelser på det använda provobjektet och ett ställningstagande på betydelsen av dessa. Den aktuella programvaruversionen ska vara dokumenterad enligt SVD.
- Testutrustning

  Beskrivning av testutrustning som krävs vid systemsäkerhetsprovningen för att kunna felinjicera i systemets ordinarie gränsytor. Det viktiga är att testutrustning inte påverkar systemfunktionen på annat sätt än den avsedda felinjiceringen. Framtagningen av testutrustningen bör samordnas med framtagningen av det tekniska systemets underhållsfunktioner.
- Provgenomförande

  Provgenomförandet ska vara överenskommet med FMV och redovisas tillsammans med förutsättningar för provet och förväntat resultat.

  Provgenomförande består av två delar, en ordinarie provomgång samt ett tilläggsprov. Den ordinarie provomfattningen ska alltid genomföras vid varje ny systemversion. Detta prov omfattar alla testbara säkerhetsfunktioner och funktionsövervakningar av dessa säkerhetsfunktioner (det vill säga både hård- och programvarufunktioner). Syftet är att säkerställa att ingen oavsiktlig påverkan har skett på säkerhetsfunktionerna i den nya systemversionen.

  Tilläggsprovet är specifikt utformat för införd funktionalitet, eller åtgärdade avvikelser sedan föregående systemversion vilka har bedömts ha inverkan på systemsäkerheten.

9.2.17 Systemsäkerhetsprovningsrapport/System Safety Test Report, SSTR

Syftet med *System Safety Test Report* (SSTR) är att redovisa resultat från genomförd systemsäkerhetsprovning. Provrapporten ska godkännas av FMV.

SSTR ska innehålla följande:

- Sammanfattning

  En kortfattad sammanfattning av resultatet från det genomförda systemsäkerhetsprovet ska finnas. Om eventuella avvikelser har identifierats under provningen ska dessa anges med tillhörande referens till problemrapport.
- Identifikation
- Dokumentöversikt och koppling till andra dokument
- Refererande dokument
- Provobjekt och programvaruversion

  En kortfattad beskrivning av provobjektets konfiguration och status med tillhörande referenser och eventuella förändringar mot fastställt provprogram.
- Provningsresurser

  Ange var och när provningen utfördes, vilken personal från FMV och industrin som deltog och ett ställningstagande till om krav på oberoende uppfyllts.
- Testutrustning

  En sammanställning över använd testutrustning med individnummer samt eventuella avvikelser eller gjorda anpassningar som är införda före eller under provningen.
- Resultat, ordinarie prov

  En redovisning av resultatet från genomfört ordinarie prov. Eventuella avvikelser och kriterier för godkännande ska redovisas tillsammans med tillhörande problemrapport. Mätresultat och eventuella loggfiler ska vara spårbara till respektive genomfört prov.

- Resultat, tilläggsprov

  En redovisning av resultat från tillkommande provpunkter. Eventuella avvikelser och kriterier för godkännande ska redovisas tillsammans med tillhörande problemrapport. Mätresultat och eventuella loggfiler ska vara spårbara till respektive prov.
- Slutsats

  En utökad sammanfattning av resultatet från systemsäkerhetsprovningen med rekommendationer på eventuella tillkommande restriktioner eller begränsningar vid användning eller underhåll.

  Alla systemsäkerhetskrav ska redovisas i tabellform med referens till motsvarande provpunkt.

9.2.18 Systemsäkerhetsanalys Datorsystem/Sub System Hazard Analysis Computer System (SSHA CS)

Syftet med *Sub System Hazard Analysis Computer System* (SSHA CS) är att identifiera eventuellt tillkommande olycksrisker efter den initiala riskidentifieringen samt att verifiera överensstämmelse med systemsäkerhetskraven för de tekniska delsystemen, i detta fall datorsystemet. Olycksrisker som kan förknippas med felmoder i datorsystemet och operativ hantering av datorsystemet analyseras. Vidare identifieras riskreducerande åtgärder. En SSHA CS kan dokumenteras enligt DI-SAFT-80101B, System Safety Hazard Analysis Report. Exempel på analysmetoder för genomförande av SSHA framgår av H SystSäk.

SSHA för datorsystem ska innehålla följande:

- Sammanfattning
- Inledning
- Externa och interna krav

  (Lagkrav, handböcker såsom H SystSäk, H ProgSäk, H VAS, vald standard)
- Systemsäkerhetsverksamhet

  Beskrivning över hur systemsäkerhetsarbetet har genomförts och vilka metoder som har använts.

- Systembeskrivning

  Kortfattad funktionsbeskrivning och översiktsbild av datorsystemet med alla gränsytor och definierade benämningar, referenser till konstruktionsunderlag samt vilka PDS som använts med version, vilket operativsystem och vilken hårdvara. Beskrivningen ska vara direkt spårbar till respektive konstruktionsunderlag.
- Produktidentitet

  Produktnamn med artikelnummer och dokumentnummer, aktuell programvaruversion med referens till SVD för analyserad utgåva av systemversionen.
- Historik

  En förteckning över åtgärdade fel från föregående systemversion med klassning avseende systemsäkerhetspåverkan ska finnas. Referens till genomförd verifiering och validering, kvarstående kända fel med klassning samt eventuella tillkommande restriktioner i användning och underhåll.
- Tillvägagångssätt

  En redovisning av använd metodik vid analysen samt hur klassificering av olycksrisker har genomförts ska finnas.
- Vådahändelser

  En redovisning av identifierade vådahändelser som kan initieras av eller påverkas från datorsystemet ska finnas. Kravnedbrytning och identifiering av säkerhetsfunktioner ska redovisas.
- Säkerhetsarkitektur

  Redovisning av vald säkerhetsarkitektur, inklusive motivering, och val av standard ska finnas.
- Verifiering av systemsäkerhetskrav

  Redovisning av olycksrisker före och efter åtgärd, alla åtgärder i riskmatris som medför riskreducering i mer än ett steg ska vara separat analyserade och överenskomna med FMV. Redovisning av genomförda systemsäkerhetsprov.

- Analysresultat

  En utökad sammanfattning av resultatet från den genomförda systemsäkerhetsanalysen med rekommendationer på eventuella tillkommande restriktioner eller begränsningar vid användning eller underhåll. Alla systemsäkerhetskrav ska redovisas i tabellform med referens till motsvarande verifiering.
- Förkortningar och definitioner
- Referenser

  Referenser till allt granskningsunderlag för analyserna ska finnas spårbart med datum och versionsnummer. Det är inte tillräckligt att endast hänvisa till en överliggande dokumentstruktur för det tekniska systemet. Alla dokument ska vara tillgängliga i digitalt format, exempelvis i PDF-format.
- Bilagor

  Exempelvis Hazard Log Datorsystem och FTA Datorsystem.

10 CE-MÄRKTA PRODUKTER SAMT PRODUKTER GODKÄNDA AV ANNAN AKTÖR

Detta kapitel behandlar produkter och tekniska system som är godkända av annan tillförlitlig aktör eller av annat land, oavsett om produkten försetts med konsumentmärkning eller inte. FMV måste alltid kontrollera att produkten eller det tekniska systemet uppfyller Försvarsmaktens behov avseende användningsmiljö och operationsbetingelser, samt skaffa sig visshet om att produkten uppfyller lagar och förordningar.

10.1 ALLMÄNT OM CE-MÄRKTA PRODUKTER

CE-märkning är obligatorisk i EU-lagstiftningen för vissa specificerade produktkategorier. Produkter som är särskilt framtagna för militär verksamhet och som syftar till att användas i strid kan dock inte CE-märkas. Maskiner tillverkade för militära ändamål omfattas således inte av maskindirektivet. Exempelvis kan inte ett vapen CE-märkas, eftersom dess huvudfunktion är att skada tredje person, men däremot kan en ammunitionsröjningsmaskin med tillhörande vapen CE-märkas.

Genom CE-märkningen intygar tillverkaren/distributören att produkten överensstämmer med de lagstadgade kraven på bland annat säkerhet, hälsa och miljö. För vissa produkter räcker det att tillverkaren/distributören själv intygar att produkten uppfyller alla krav. För andra produkter, som anses särskilt riskfyllda, krävs att tillverkaren/distributören låter ett oberoende tredjepartsorgan kontrollera produkten.

Som del i CE-märkning ska tillverkaren/distributören upprätta teknisk dokumentation för produkten, samt utfärda en EU-försäkran om överensstämmelse. Den CE-märkta produkten ska åtföljas av en bruksanvisning som innehåller all väsentlig information för att produkten ska kunna användas på ett säkert sätt för avsett ändamål. Vid leverans till slutkund ska produkten åtföljas av en bruksanvisning på mottagarlandets språk.

Vid anskaffning av kommersiella produkter föreligger ofta svårigheter att få tillräcklig information om tidigare genomförda systemsäkerhetsanalyser. Beroende på hur man avser att använda dessa produkter kan de bli säkerhetskritiska.

Detta kapitel avser att definiera rimlig omfattning av systemsäkerhetsverksamhet vid upphandling av CE-märkta produkter som innehåller programvara och som man avser att använda fristående från militära system. Ska den CE-märkta produkten integreras i det militära tekniska systemet ska denna integration omfattas av systemsäkerhetsanalys. Med fristående användning menas att produkten får strömförsörjas, utbyta information eller integreras med annan teknisk produkt i enlighet med tillverkarens/distributörens anvisningar. Denna kategori produkter kan i sin tur delas in i undergrupper för vilka behovet av systemsäkerhetsverksamhet varierar avseende omfattning och innehåll. Uppdelningen av fristående CE-märkta produkter är enligt följande och behovet av systemsäkerhetsverksamhet beskrivs för var och en enligt underavsnitten.

- Produkter som redan finns på marknaden och som innehåller säkerhetskritisk programvara men man planerar inte att genomföra egna uppdateringar eller andra ändringar (*avsnitt 10.2.*)
- Nyutvecklade produkter eller tekniska system som inte finns på marknaden men som CE-märks före leverans till FMV. Uppdateringar kan bli aktuella (*avsnitt 10.3*).
- Produkter eller tekniska system som godkänts av annan tillförlitlig aktör såsom annat land eller som erbjuds genom Nato:s försorg (*avsnitt 10.4*).

10.2 CE-märkta produkter som redan finns på marknaden

Detta avsnitt beskriver CE-märkta produkter som överensstämmer med samtliga av nedanstående påstående:

- Produkten var CE-märkt då den sattes på marknaden.
- Produkten avses enbart användas fristående och inte integreras i ett tekniskt system.
- Eventuella uppdateringar av programvaran genomförs uteslutande av leverantören.

För enkla CE-märkta produkter som innehåller programvara som inte är säkerhetskritisk krävs oftast ingen systemsäkerhetsverksamhet utöver att just göra bedömningen och klassningen att den aktuella produkten är av denna enkla karaktär. Exempel på sådana produkter kan vara persondatorer, bildskärmar, hushållsmaskiner (till exempel ugnar, tvättmaskiner), verktyg för hantverksarbete (till exempel borrmaskiner, lasermätare). För sådana produkter finns ofta harmoniserade standarder som bör vara uppfyllda vilket stärker säljarens grund för CE-märkning och förenklar kundens leveranskontroll. Vid överlämning ska CE-deklarationen (*Declaration of Conformity*, DoC) samt handhavande- och underhållsdokumentation bifogas. Produkten får endast användas/hanteras fristående från annan materiel i enlighet med tillverkaren/distributören instruktioner.

Om det av CE-märkningen framgår att harmoniserad standard är uppfylld kan produkten bedömas som tolerabelt säker förutsatt att kunden inte genomför egen programvaruuppdatering. Uppdateringar kan dock vid behov genomföras av tillverkaren/distributören inom ramen för dennes CE-märkning. Exempel på sådana produkter kan till exempel vara medicinskteknisk utrustning. Vid överlämning ska CE-deklaration (*Declaration of Conformity*, DoC) samt handhavande- och underhållsdokumentation bifogas. Produkten får endast användas/hanteras fristående från annan materiel i enlighet med tillverkarens/distributörens instruktioner.

10.3 CE-märkta produkter som inte finns på marknaden

Till denna kategori hör CE-märkta produkter samt produkter av teknisk karaktär som inte redan finns på marknaden (till exempel båtar och funktionscontainrar). FMV kan i sådana upphandlingar ställa särskilda krav på användningsmiljö, operationsbetingelser, systemsäkerhetsarbete och dess dokumentation inför tillverkaren/distributörens CE-märkning.

Precis som för produkter i *avsnitt 10.2* ovan gäller det att avgöra om programvaran i aktuell produkt är säkerhetskritisk i någon mening och i det fall den bedöms vara säkerhetskritisk, försäkra sig om att tillämpligt systemsäkerhetsarbete på dessa programvarustyrda funktioner är genomfört. När detta är gjort och så länge produkten används inom ramen för CE-märkningen och inga ändringar implementeras i handhavandeinstruktionen eller i konstruktionen, kan systemet betraktas som tolerabelt säkert.

Skillnaden mot de typer av produkter som beskrivs i *avsnitt 10.2* är att typerna i *avsnitt 10.3* oftast har lång livslängd. Den långa planerade livslängden kan i större grad innebära att behov av egna uppdateringar uppstår. Om FMV avser att genomföra egna uppdateringar av programvaran i en CE-märkt produkt behöver FMV ha tillgång till utvecklingsmiljö, dokumentation och källkod för produkten. Detta är inte alltid möjligt att få av tillverkaren/distributören och kan medföra att uppdateringen inte kan genomföras. Om det har gått mer än tio år efter att den sista produkten levererades är tillverkaren/distributören inte längre skyldig att tillhandahålla teknisk dokumentation för produkten vilket kan försvåra dokumentationen av den egna uppdateringen. Om det bedöms kunna bli aktuellt att göra egna uppdateringar av programvaran efter att denna tidsfrist löpt ut bör detta klargöras innan Försvarsmakten lägger anskaffningsuppdrag till FMV.

Uppdateringar av programvara som inte genomförs av tillverkaren/distributören, eller på inrådan av tillverkaren/distributören, kan göra att tillverkarens/distributörens CE-deklaration blir inaktuell eller inte tillämplig. I de fall den ursprungliga CE-märkning inte kan tillämpas, eller då det tänkta användandet av produkten inte täcks av deklarationen, kan en ny CE-märkning genomföras av auktoriserat certifieringsorgan på uppdrag av

FMV. Alternativt genomförs en liknande process där de krav enligt EU-direktiv som inte kan uppfyllas, ska dokumenteras. För dessa ouppfyllda krav genomförs systemsäkerhetsanalys enligt H SystSäk. FMV måste innan anbudsinfordran går ut och efter dialog med Försvarsmakten avgöra vilken omfattning av dokumentation som ska beställas av industrin för att säkerställa möjligheten till framtida egna uppdateringar av programvaran.

När nyutvecklad produkt som inte finns på marknaden anskaffas för fristående användning inom Försvarsmakten kan produkten med anledning av det ovanstående CE-märkas. Vissa undantag finns för produkter som är framtagna för särskild militär verksamhet och som inte omfattas av kraven på CE-märkning.

Det är viktigt att tidigt döma av om det under vidmakthållandefasen kan bli aktuellt för FMV att genomföra egna uppdateringar av programvaran i den produkt som ska CE-märkas. För detta krävs att FMV har full kontroll över bland annat programvaruutvecklingsmiljön. Detta är inte alltid möjligt att få av tillverkaren/distributören och det kan innebära att uppdateringen inte kan genomföras utan en reducering av säkerheten.

FMV som anskaffar en nyutvecklad produkt som ska CE-märkas bör särskilt säkerställa att tillverkaren/distributören följer och redovisar vilka EU-direktiv och harmoniserade standarder som tillverkaren/distributören gör sin försäkran om överensstämmelse mot. Därför måste FMV definiera och beskriva produktens tänkta driftprofil, användningsområde och operationsbetingelser. FMV reglerar detta i kravställningen till tillverkaren/distributören så att tillverkaren/distributören är införstådd i användandet. Tillverkaren/distributören ska i en systemsäkerhetsplan (SSPP) med beaktande av detta redogöra för vilka EU-direktiv och harmoniserade standarder produkten kommer att CE-märkas mot, samt hur kontroll av uppfyllnad av dessa standarder och krav ska genomföras.

I samband med leverans ska tillverkaren/distributören lämna en deklaration om överensstämmelse (*Declaration of Conformity*, DoC), samt den tekniska dokumentation som är relevant för produkten.

10.4 Produkter certifierade eller godkända av annan part

I linje med EU:s säkerhets- och försvarspolitik genomförs arbete med att stegvis upprätta en europeisk marknad för försvarsmateriel och för att möta behovet av militär kapacitet. I arbetet med att stärka en europeisk försvarsindustriell och militärteknologisk bas, ger direktivet 2009/81/EG vägledning till upphandlande myndigheter att standardisera tekniska specifikationer och utvärdera anbud baserade på likvärdiga lösningar avseende prestanda- och funktionskrav, samt refererande till internationella, europeiska eller nationella standarder.

EU-kommissionen uttryckte följande den 20 december 2013: *The European Defence Agency (EDA) and the Commission will prepare a roadmap for the development of defence industrial standards by mid-2014, without duplicating existing standards, in particular NATO standards*.

EDA utgör ett serviceorgan för sina medlemsstater med uppgift att stödja, effektivisera och samordna utveckling och inköp av försvarsmateriel. EDA utvecklar och tillhandahåller verktyg för upphandlande myndigheter, exempelvis *Collaborative Database* (CODABA), *Third Party Logistic Support* (TPLS) samt *Platform and Procurement Experts Network* (PEN). EDA arbetar för att harmonisera krav på försvarsmateriel och för samverkansupphandlingar med flera deltagande medlemmar. Ett mål är att konsolidera och standardisera krav för en kostnadseffektiv försvarsmaterielförsörjning. En lösning är användarklubbar, så kallade *User clubs*, med flera medlemsstater där krav, utvecklingsmetoder, standarder, godkännandeprocesser med mera kan samordnas på ett effektivt sätt.

Inom Nato finns N*ATO Support and Procurement Agency* (NSPA) med uppgift att bistå medlemsstater med anskaffning av försvarsmateriel, främst vid köp av materiel ”rakt över disk”. NSPA har bland annat i uppgift att ta fram och anpassa teknisk dokumentation i samband med försäljning av försvarsmateriel.

För flygtrafikledningssystem, som också omfattar mycket av de militära systemen, finns styrande direktiv. Exempelvis EG 552/2004 *Driftkompatibiliteten hos det europeiska nätverket för flyg-*

ledningstjänst ställer krav på interoperabilitet mellan nationella system. Krav ställs på att underlag från genomförda driftgodkännandeprocesser, inklusive underlag för systemsäkerhetsgodkännanden och programvarusäkerhetsgodkännanden ska levereras till begärande myndighet inom EU. Krav ställs både på systemnivå, exempelvis ett radiokommunikationssystem, som på komponentnivå, exempelvis en flygradio. Leverantörer av sådana system och produkter bör sammanställa underlag för driftgodkännande med hänsyn till att underlaget kan spridas till myndigheter inom hela EU.

För ammunition, som alltmer innehåller komplex programvara, finns inom EDA *The European Network of National Safety Authorities on Ammunition* (ENNSA). ENNSA syftar till ”*Better communication among national safety authorities on ammunition and to improve harmonization of national practices on ammunition safety standardization and test procedures where feasible*”.

Inom Nato finns *Munitions Safety Information Analysis Center* (MSIAC), en projektorganisation som består av vissa Natoländer samt ytterligare några länder, däribland Sverige. Syftet är att stödja medlemsländerna i ammunitionssäkerhetsfrågor. Denna organisation har ett bredare syfte än ENNSA. Tekniska frågor kring ammunitionens konstruktion och erfarenheter under användning kan ställas till MSIAC genom Sveriges kontaktperson som är placerad vid FMV.

Bild 10:1 ger ett exempel på hur en Nato-standard med krav för ammunition kan mötas med hjälp av en internationell civil standard för programvara i säkerhetskritiska tillämpningar.

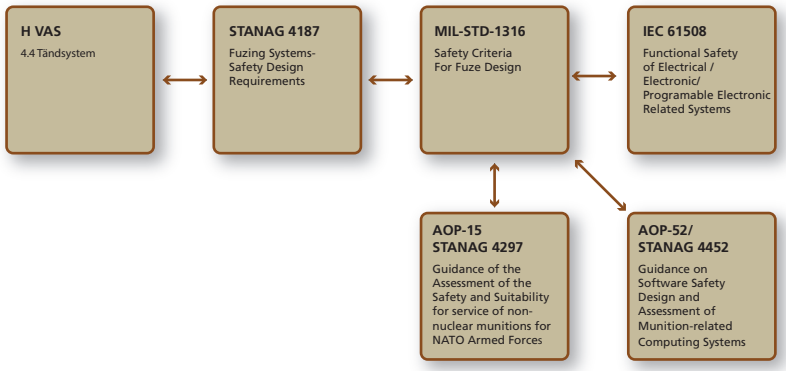


Bild 10:1 Koppling mellan Nato-standard/krav för ammunition och civila standarder

Utrustning som regleras av direktivet för marin utrustning ska rattmärkas (typgodkännas) och inte CE-märkas. Rattmärket visar att produkten uppfyller kraven enligt direktivet. Rattmärkning av utrustning regleras genom ett EU-direktiv. I direktivet fastställs gemensamma regler i syfte att undanröja skillnader vid genomförandet av internationella standarder genom att ha en tydligt identifierad uppsättning krav och enhetliga certifieringsförfaranden.

Europaparlamentets och rådets direktiv 2014/90/EU om marin utrustning är satt i kraft genom lagen (2016:768) om marin utrustning och förordningen (2016:770) om marin utrustning tillsammans med Transportstyrelsens föreskrifter (TSFS 2016:81) om marin utrustning. En produkt som är typgodkänd av ett anmält organ i en medlemsstat inom EU får placeras på ett gemenskapsfartyg, oavsett vilken flagg det för, vilket främjar fri rörlighet för marin utrustning på den inre marknaden.

Utöver ovanstående finns ytterligare EU-direktiv som har undantag för militär materiel, till exempel RoHS (avser elektrisk och elektronisk utrustning) och Reach (kemiska ämnen).

11 Hantering av tidigare utvecklad programvara (PDS)

Tidigare utvecklad programvara, ***Previously Developed Software*** (PDS), är programvara som redan finns färdigutvecklad då det nya tekniska systemet ska utvecklas. PDS kan antingen vara utvecklad internt hos industrin eller införskaffat från annan aktör. Varianter av tidigare utvecklad programvara kan även vara ***Commercial Off The Shelf*** (COTS), ***Government Off The Shelf*** (GOTS), ***Military Off The Shelf*** (MOTS), ***NATO Off The Shelf*** (NOTS) eller ***Open Source Software*** (OSS).

Vid utveckling av ett nytt tekniskt system med programvara ställs man inför valet att utveckla helt ny programvara, vidareutveckla en befintlig eller att använda en tidigare färdigutvecklad programvara.

11.1 Att beakta vid vägval för användning av PDS

Att införskaffa PDS kan initialt upplevas vara mer ekonomiskt förmånligt än att utveckla en ny programvara för en viss funktion. PDS måste dock uppfylla samma krav som för övrig nyutvecklad programvara avseende olika aspekter såsom systemsäkerhet, IT-säkerhet, testning, dokumentation, kvalitet, konfigurationsledning.

Nedan belyses några aspekter som kan förväntas uppkomma då PDS återanvänds i nya tillämpningar.

- PDS har oftast tagits fram för en bred användning och för att passa många användare, eller för användning i en tidigare version av aktuellt tekniskt system. Detta kan innebära att de specifika kraven för det nya tekniska systemet inte helt uppfylls.

- Uppfylls inte de specifika kraven kan det krävas en anpassning av PDS vilket kan leda till omfattande utvecklingsarbete och provning, speciellt om det saknas tillgång till fullständig dokumentation. Ibland måste ett omfattande arbete läggas på testning för att kunna kvalificera programvaran till erforderlig kritikalitetsnivå.
- Om dokumentation saknas kan detta försvåra framtida systemuppdateringar och testning. Om PDS ska användas i ett sammanhang där det krävs en högre kritikalitetsnivå än den är testad och dokumenterad för kan detta omöjliggöra användning.
- Brist på dokumentation kan leda till att designansvarig inte förstår funktionaliteten för PDS. Detta kan leda till att programvaran används på ett felaktigt sätt och att den inte kan testas tillräckligt noggrant.
- Om PDS är tänkt att vara generell och passa för många olika tillämpningar kan det finnas funktioner som inte behövs för det aktuella tekniska systemet. Således kan PDS innehålla död eller deaktiverad kod som kan orsaka felfunktion.
- Att kritikalitetsklassificera PDS till en högre nivå än den är certifierad för kan medföra höga kostnader eller vara omöjligt.
- Vid underhåll av tekniskt system med lång livslängd som innehåller PDS, gäller det att kontrollera att eventuella licenser fortfarande är giltiga och därmed kan underhållas och uppdateras genom underhållsavtal med leverantör. Om behov av ny funktionalitet i programvaran behövs, för att ge det tekniska systemet bättre prestanda eller nya funktioner, så ska dess utvecklingshistorik vara känd och spårbar.
- Det ska säkerställas att Försvarsmakten får nyttjanderätten till programvaran genom licenser i de fall utvecklande industri köper in PDS av annan aktör för användning i ett tekniskt system.
- Om leverantören gör en särskild version av PDS för det nya tekniska systemet, ska den hanteras på samma sätt som för annan nyutvecklad programvara. Denna nya PDS ska då kritikalitetsklassificeras och uppfyllas kraven i vald programvarustandard.

11.2 Förutsättningar för användning av PDS

Att återanvända PDS kan vara fördelaktigt i de fall programvarans funktionalitet motsvarar det som eftersöks och att den har använts i liknande tillämpningar. När PDS kan vara ett alternativ är:

- I stora tekniska system, som exempelvis ett fartyg, där man kan ha relativt låg kritikalitetsnivå på vissa oberoende och okritiska funktioner.
- När leverantören av PDS har certifierat programvarukomponenten eller funktionskomponent enligt viss programvarustandard, exempelvis certifierade komponenter enligt IEC 61508 med angiven SIL-nivå.
- För drivrutiner i gränsytor mot standardiserade hårdvarukomponenter.
- När leverantören av PDS kan leverera komplett dokumentation för erforderlig kritikalitetsnivå och det finns dokumenterad driftserfarenhet.

11.3 Utvärdering av leverantör för PDS

Vid utvärdering av olika PDS-lösningar, som alternativ till nyutveckling, behöver nedanstående aspekter utvärderas:

- Har leverantören funnits länge på marknaden och levererat liknande produkter?
- Bedöm leverantörens intresse och möjligheter att genomföra framtida systemuppdateringar och ge support på aktuell PDS, exempelvis genom särskilt avtal.
- Bedöm möjligheter att få tillgång till utvecklingsmiljöer och dokumentation såsom källkod, specifikationer, testdokument, användarhandledning, designdokument, beskrivningar och genomförda felrättningar.
- Kommer leverantören att överlämna kvalitetsdokument som kan redovisas för aktuell PDS som bevis för att dokumenterad utvecklingsprocess följts såsom granskningsprotokoll, testrapporter och kvalitetsgranskningsrapporter.
- Har leverantören använt aktuell PDS i andra liknande tekniska system och finns bevis för att PDS fungerar korrekt (*Proven in use*), med referenser till ett avvikelserapporteringssystem (*Service history*).
- Har leverantören definierade metoder som kan användas för att påvisa *Proven in use* av aktuell PDS och finns dessa dokumenterade?
- Kan leverantören uppvisa certifikat från oberoende tredjepartsgranskning?

12 Angränsande metodik- och teknikområden

I detta kapitel diskuteras ett antal angränsande metodik- och teknikområden. Syftet är att ge vägledning till hur närliggande områden kan behandlas och att därmed indirekt också beskriva tillämpligheten av den metodik som beskrivs i denna handbok.

12.1 Systemsäkerhetsverksamhet

Systemsäkerhetsverksamhet är det totala arbete som bedrivs under ett tekniskt systems hela livscykel i syfte att identifiera, analysera, värdera och åtgärda olycksrisker. Se H SystSäk.

I denna handbok beskrivs hur programvara och utveckling av programvara ska kravställas utifrån den inverkan den har på systemsäkerheten. Systemsäkerhet definieras nedan.

Systemsäkerhet

”Egenskapen hos ett tekniskt system att inte oavsiktligt orsaka skada på person, egendom eller yttre miljö”

Källa: H SystSäk

Det finns även andra krav på det tekniska systemet och den programvara som ingår samt på hur systemet med programvaran utvecklas. Några utvalda områden beskrivs nedan.

12.2 Verksamhetssäkerhet

Verksamhetssäkerhet syftar på Försvarsmaktens förmåga att hantera risker vid all verksamhet, inte bara de risker som är förknippade med tekniska system.

12.3 Informationssäkerhet

Informationssäkerhet syftar på skydd av informationstillgångar. I samband med programvara syftar man oftast på att skydda information som finns i IT-system, eller deras funktion, mot angrepp utifrån. Detta hanteras både genom tekniska och administrativa åtgärder i enlighet med *Handbok Försvarsmaktens säkerhetstjänst, Informationssäkerhet* (H Säk Infosäk 2013), särskilt avsnitten om IT-säkerhet, kommunikation och ackreditering, samt den senaste versionen av *Krav på säkerhetsfunktioner* (KSF).

All verksamhet är beroende av skyddsvärda tillgångar i form av personal, information, materiel och anläggningar. Utifrån verksamhetens karaktär kan en del av dessa tillgångar vara mer skyddsvärda än andra. Informationssäkerhetsområdets syfte är att skydda dessa tillgångar mot oönskade händelser.

Ur ett informationssäkerhetsperspektiv ska informationen vara:

- tillgänglig för alla som behöver den
- riktig
- skyddad mot obehörig åtkomst
- spårbar.

Om skyddet av informationen brister så att den blir otillgänglig, felaktig eller oönskat spridd till obehöriga, så får det konsekvenser för verksamhet, materiel, personal och anläggningar.

12.3.1 Informationssäkerhetsdeklaration (ISD)

Processen med Informationssäkerhetsdeklaration (ISD) är en del av Försvarsmaktens ackrediteringsprocess inom ramen för Försvarsmaktens materielförsörjning. Huvudsyftet med ISD-processen är att man genom en viss struktur och ett arbetssätt ska skapa förtroende för ett systems utformning och funktion ur perspektivet informationssäkerhet. Försvarsmakten kan då handha materielen med tolerabel risk med avseende på informationssäkerhet. ISD skapar enhetlighet, tydlighet, spårbarhet och effektivitet i arbetet med egenskaper som berör informationssäkerheten i verksamheter och tekniska system.

ISD följer standarderna ISO/IEC 15288 för *Systems Engineering* och ISO/IEC 27000 för *Ledningssystem inom informationssäkerhet*. Underlag angående ISD finns tillgängligt på FMV:s hemsida.

ISD är både en stödprocess för informationssäkerhetsarbete på FMV samt en deklaration avseende informationssäkerhet.

Syftet med ISD-processen och dess metodstöd är att skapa enhetlighet, tydlighet, spårbarhet och effektivitet i arbetet med att ta fram ackrediteringsunderlag och ISD-deklaration. Processen startar med kravdefinitionen och utvecklas under kravnedbrytning, produktion och överlämning. Den efterföljande deklarationen fastställer sedan att FMV:

- tar ett designansvar för IT-säkerhetslösningen
- uppfyller Försvarsmaktens krav på informationssäkerheten
- har utformat dokumentationen enligt den norm som gäller inom FMV
- har dimensionerat IT-säkerhetslösningen utifrån en definierad tolererbar risk utformad av Försvarsmakten
- har följt fastställd ISD-plan rörande IT-säkerhetsarbetet.

Finns särskilda villkor för denna deklaration ska dessa aspekter också redovisas i deklarationen.

Hur FMV ska bedriva informationssäkerhetsarbetet framgår av ISD-handböckerna:

- ISD IT-säkerhet Management
- ISD IT-säkerhet Oberoende granskning
- ISD IT-säkerhet Användningsfall och arkitektur.

12.3.2 Signalskydd

När känslig information ska skickas mellan olika enheter inom Försvarsmakten, eller mellan olika myndigheter, krypteras den. Specialfallet programvara med huvudsaklig uppgift att kryptera information hanteras utanför denna handbok.

Där signalskydd ska ingå kan kunskap inhämtas från *Handbok totalförsvarets signalskyddstjänst grundläggande regler för signalskydds signalskyddstjäns*t, H TST Grunder (M7746-734002).

12.4 Funktionella egenskaper

Funktionen hos en programvara kan även ha inverkan på förmågan att genomföra ett uppdrag, till exempel om ett vapen går att rikta mot ett identifierat mål eller inte. Det kan även finnas bristande funktion som innebär att man utsätter sig för fara genom att man utsätter sig för fientlig eller egen bekämpning genom att motmedelssystem eller identifieringssystem inte har avsedd funktion.

Det finns erfarenhet från utländska utvecklingsprojekt för militära system där metodik för kritikalitetsklassificering tillämpats som innebär att kritikalitetsklassificeringen av programvara (och därmed kraven på utvecklingen av programvaran) enligt aktuell programvarustandard tillämpats även för dessa aspekter.

12.5 Användbarhet

De allra flesta tekniska och icke-tekniska system kommer på ett eller annat sätt att kommunicera med operatörer och underhållspersonal. Om människor kan öka eller minska sin förmåga i denna interaktion med tekniska system har det betydelse för att system ska upplevas säkra, effektiva och användbara.

Insikt i hur psykologiska, fysiologiska, organisatoriska och tekniska aspekter interagerar i komplexa, påfrestande miljöer, skapar förutsättningar för att åstadkomma säkra, effektiva och

användbara system. Gentemot Försvarsmakten tar FMV ansvar för helheten och denna förmåga är nödvändig för att kunna omhänderta Försvarsmaktens behov i de tekniska systemen.

Grunden till att ett tekniskt system eller produkt blir användbar och till nytta för användarna och verksamheten läggs tidigt i utvecklingsarbetet. Att i efterhand kompensera för ett tekniskt systems eller en produkts brister med utbildning, eller kräva specialistkunskap hos användare, är kostnadsdrivande och verksamhetsbegränsande. Ett system med brister i säkerheten, effektiviteten och användbarheten kan vara förenat med livsfara för användarna och även ge skador på miljön eller ge ekonomiska förluster.

Vid utformning av användargränssnitt krävs kunskap om människans förutsättningar och förmågor, men även kunskap om människans begränsningar som användare och som en del av ett system. Kunskap behövs om hur människan via sina sinnen tar in information och tolkar omgivningen samt om människans minne, tänkande, processer för beslutsfattande med mera.

Som stöd för utformning av användargränssnitt finns flera olika generella principer, tumregler och riktlinjer, som ofta är baserade på erfarenhet och forskning. Exempel på detta är *Jakob Nielsens tumregler* eller *Ben Shneidermans åtta gyllene designregler*. Dessutom finns plattformspecifika designprinciper för exempelvis Microsoft Windows samt internationella standarder.

För tekniska system med lång livslängd sker ofta en vidareutveckling av användargränssnittet. Detta berör i synnerhet militära system där operatörer har utbildats och övats på en viss version av användargränssnitt. Konsekvenser av förändrade användargränssnitt kan både resultera i nya olycksrisker och minskad effektivitet då inövade handgrepp hos operatören eller visuell återkoppling från systemet förändras.

För bildskärmsarbete tillämpas AFS 1998:05, *Arbete vid bildskärm*. Utöver föreskrifter och standarder finns en FMV Handbok i användbarhet (H HFI) samt MIL-STD 1472G.

12.6 Programmerbar logik

Logik kan, förutom som programvara realiseras, i form av programmerade och i vissa fall programmerbara kretsar som *Application Specific Integrated Circuits* (ASIC), *Field-Programmable Gate Array* (FPGA) och *Programmable Logic Devices* (PLD). Detta innebär, förutom rena hårdvarufrågor, samma problematik med systematiska fel som finns för programvara. Därför kan det vara nödvändigt att för dessa komponenter, och de funktioner de stödjer, tillämpa en metodik som motsvarar den som tillämpas för programvara.

I *avsnitt 2.10* beskrivs standarden RTCA DO-254 som tillämpas för programmerbar logik inom flygområdet. Inom flygområdet finns även omfattande tillämpningsföreskrifter för standarden från amerikanska (FAA) och europeiska luftfartsmyndigheter (EASA).

Den beskrivning som finns i *Handbok Vapen- och ammunitionssäkerhet* (H VAS) för tillämpning av programmerbar logik i tändsystem är även användbar för andra typer av styrfunktioner i tekniska system. Se även STANAG 4187. Området behandlas inte i övrigt i denna handbok, men kraven i H VAS kan tillämpas.

12.7 Metoder för snabb systemutveckling

Det finns ett antal metoder för att forcera systemutvecklings- och programmeringsarbete i syfte att säkra resultat som kan omsättas i försäljning snabbare än i traditionell systemutveckling. I mer traditionellt arbetssätt riskeras att det tar lång tid att omhänderta kundbehov eftersom det kan ta lång tid att hantera många krav i relativt stora utvecklingssteg, istället för att som i de alternativa metoderna ta mindre, men desto fler, utvecklingssteg som vart och ett kan ge ett resultat som kan levereras eller sättas på marknaden. Bland exempel på metoder märks SCRUM och Agil systemutveckling.

Ibland kan beskrivningar av de här metoderna tolkas som att man inte behöver ha full spårbarhet på krav, ha styrda processer, ha komplett dokumentation eller göra fullständig verifiering av funktioner. Samma krav gäller dock vid den här typen av arbetssätt som vid användning av mer konventionella metoder.

13 Sammanställning av krav

Kapitel 6 Förutsättningar från Försvarsmakten

Avsnitt 6.1 Förutsättningar inför utveckling av tekniska system

Krav nr	Innehåll
2.601.01-A	FMV skall begära att Försvarsmakten preciserar sammanhang, användnings- och omgivningsmiljö samt operationsbetingelser för det tekniska systemet. *Kommentar*: Detta gäller både för militär användning och i förekommande fall för stöd till samhället i fredstid.
2.601.02-A	FMV skall begära att Försvarsmakten definierar övergripande funktionsinriktade prestandakrav för tekniskt system.
2.601.03-A	FMV skall begära att Försvarsmakten definierar tolerabel risknivå för det tekniska systemet under hela dess livslängd.
2.601.04-A	FMV skall begära att Försvarsmakten tillgängliggör drifterfarenheter från tidigare motsvarande tekniska system.

Avsnitt 6.2 Förutsättningar under utveckling av tekniska system

Krav nr	Innehåll
2.602.01-A	FMV skall av Försvarsmakten efterfråga vilken aktör som utses till att vara tekniskt designansvarig organisation. *Kommentar*: Om annan aktör än FMV ska vara tekniskt designansvarig behöver detta framgå av FMV:s Systemsäkerhetsgodkännande (SSG).

Avsnitt 6.3 Förutsättningar inför överlämning och användning

Krav nr	Innehåll
2.603.01-A	FMV skall begära att Försvarsmakten har ett avvikelserapporteringssystem för tekniska system där avvikelser kan rapporteras. *Kommentar*: Om annat avvikelserapporteringssystem än Försvarsmaktens ordinarie ska användas, behöver FMV ha kännedom om detta.
2.603.02-A	FMV skall begära att Försvarsmakten följer de anvisningar som överlämnas avseende handhavande, vidmakthållande/underhåll samt rutiner för att genomföra systemuppdateringar på överlämnad materiel. *Kommentar*: Om annan aktör än FMV ska vara Tekniskt designansvarig behöver Försvarsmakten delge FMV detta.
2.603.03-A	FMV skall utifrån Försvarsmaktens ställda krav specificera vilka inskränkningar och krav som gäller för personal som hanterar, vidmakthåller/underhåller eller genomför systemuppdateringar på överlämnad materiel. *Kommentar*: Detta gäller särskilt för insats då systemuppdatering kan behöva genomföras av Försvarsmaktens egen personal.

Krav nr	Innehåll
2.604.01-A	FMV skall hos Försvarsmakten begära avvikelserapporter för det tekniska systemet. *Kommentar*: Informationen kan överlämnas till Arbetsgrupp för systemsäkerhet (SSWG).
2.604.02-A	FMV skall begära att Försvarsmakten deltar i Arbetsgrupp för systemsäkerhet (SSWG).

Kapitel 7 Verksamhetskrav på FMV

Avsnitt 7.1 FMV:s arbete under livscykeln

Krav nr	Innehåll
2.701.01-A	FMV:s systemsäkerhetsledningsplan (SSMP) skall omhänderta krav på programvarusäkerhet. *Kommentar*: FMV:s SSMP ska omhänderta Försvarsmaktens krav på tolerabel risknivå för det tekniska systemets alla systemnivåer. I de fall FMV utarbetar en intern SSPP för ett projekt ska den även omfatta programvarusäkerhet.
2.701.02-A	Programvarusäkerhetsfrågor skall omhändertas av Arbetsgrupp systemsäkerhet (SSWG).

Avsnitt 7.3 Utveckling, produktion och anskaffning

Krav nr	Innehåll
2.703.01-A	FMV skall säkerställa att SSPP omfattar programvarurelaterade systemsäkerhetsaktiviteter innan kontrakt tecknas. *Kommentar*: SSPP ska vara utformad i enlighet med H SystSäk och innehålla alla nödvändiga aktiviteter och metoder för att genomföra programvarusäkerhetsarbetet och i förekommande fall i enlighet med överenskommen programvarustandard.
2.703.02-A	FMV skall för programvara med initial kritikalitetsklassificering ***HÖG*** överenskomma med industrin om etablerad programvarustandard inklusive kritikalitetsnivå, tillämplig inom teknikområdet, med vilken industrin ska visa överensstämmelse. *Kommentar*: Vid kritikalitetsnivå ***LÅG*** räcker grundkraven (GKPS).
2.703.03-A	FMV skall säkerställa att det av protokollet från kontraktsgenomgången framgår vilka eventuella avsteg från GKPS som överenskommits. *Kommentar*: Av protokollet ska det framgå att industrin kommer att uppfylla övriga krav enligt GKPS. Flera kontraktsgenomgångar kan genomföras under tiden för genomförande av projektet.

Krav nr	Innehåll
2.703.04-A	FMV skall säkerställa att industrin redovisar avvikelser med betydelse för systemsäkerhet som identifierats under utveckling och drift samt den totala mängden avvikelser. *Kommentar*: Redovisningen ska vid leverans åtminstone omfatta vilka avvikelser som är öppna eller stängda från och med verifierande provning. FMV ska ha omhändertagit eventuella öppna anmärkningar i sitt systemsäkerhetsarbete åtminstone genom att ta ställning till att de inte föranleder åtgärder med avseende på systemsäkerhet.
2.703.05-A	FMV skall kravställa att industrin visar överensstämmelse med grundkraven (GKPS) i denna handbok för alla programvaror oavsett kritikalitetsnivå.
2.703.06-A	FMV skall säkerställa att industrin kan ge stöd för analys och åtgärder för uppkomna systemsäkerhetsproblem under systemets livslängd. *Kommentar*: FMV ska beställa stöd från tillverkaren i enlighet med den omfattning och den tid FMV och Försvarsmakten har behov av. Hänsyn ska tas till det tekniska systemets egenskaper och förväntade livslängd.
2.703.07-A	FMV:s och industrins systemsäkerhetsarbete inklusive programvarusäkerhet skall vara slutfört och systemsäkerhetsgodkännande (SSG) skall vara fastställt före överlämning till Försvarsmakten.

Avsnitt 7.4 Användning och systemuppdateringar

Krav nr	Innehåll
2.704.01-A	FMV skall säkerställa att industrin har tillgång till programvarans utvecklingsmiljö under produktens hela livscykel i den omfattning som behövs. *Kommentar*: Omfattning regleras av FMV beställning.
2.704.02-A	Uppdatering av systemsäkerhetsgodkännande (SSG) skall alltid göras vid förändring eller modifiering av ett tekniskt system. *Kommentar*: Se H SystSäk avseende Systemsäkerhetsgodkännande (SSG).

Avsnitt 7.5 Avveckling av tekniskt system

Krav nr	Innehåll
2.705.01-A	Förnödenheter som programvaror och datorer skall registreras i relevanta stödsystem i samband med leverans från industrin. *Kommentar*: Detta gäller även förnödenheter som överförts i statens ägo, men som finns kvar hos industrin.
2.705.02-A	Avveckling av tekniskt system (eller del av tekniskt system) skall omfatta de resurser som används för stöd till utveckling och vidmakthållande av systemen. *Kommentar*: I det som ska hanteras ingår utvecklingsmiljöer för programvaran, avtal för verksamhet inklusive personal och dataförsörjning med mera

Kapitel 8 Grundkrav (GKPS) på utvecklande industrin

Avsnitt 8.1.1 Krav på kompetens hos personal

Krav nr	Innehåll
2.801.01-A	Roller inklusive erforderlig kompetensnivå skall överenskommas med FMV. *Kommentar*: Kompetensprofil för personal som medverkar i utvecklingen av det tekniska systemet såsom projektledare, tekniskt ansvarig för systemarkitektur, verifieringsansvarig samt oberoende granskare dokumenteras.
2.801.02-A	Minst två personer skall ha kännedom om motiv för vald systemarkitektur. *Kommentar*: Valet av systemarkitektur utifrån genomförd kravnedbrytning av dimensionerade vådahändelser ska vara känd av minst två personer.
2.801.03-A	Utvecklande industri skall utse en kontaktperson för programvarusäkerhet. *Kommentar*: Denna person säkerställer att för projektet överenskommet arbetssätt och metodik för systemsäkerhetsarbetet följs samt ansvarar för verifieringen av grundkraven (GKPS) och redovisar att dessa krav är uppfyllda.

Avsnitt 8.1.2 Krav på verksamhets- och systemsäkerhetsledning

Krav nr	Innehåll
2.801.04-A	Industrin skall följa AQAP 2110. *Kommentar*: Detta gäller främst insynsrätten.
2.801.05-A	Industrin skall följa AQAP 2210.

Krav nr	Innehåll
2.801.06-A	Industrin skall ta fram en Systemsäkerhetsplan (SSPP). *Kommentar*: Systemsäkerhetsplanen (SSPP) ska även omfatta erforderliga aktiviteter såsom kravdokument, testplaner och testprocedurer för programvaruutveckling, samt en beskrivning av överenskomna utvecklingsverktyg.
2.801.07-A	Utvecklande industri skall i Systemsäkerhetsplanen (SSPP) redovisa hur GKPS kommer att uppfyllas.
2.801.08-A	Systemsäkerhetsanalys skall omfatta datorsystemets påverkan på det tekniska systemets hela livscykel. *Kommentar*: Analysen ska utföras iterativt under utvecklingen, från kravnedbrytning till avslutad verifiering.

Avsnitt 8.1.3 Krav på utformning av säkerhetsarkitektur

Krav nr	Innehåll
2.801.09-A	För datorsystemet skall säkerhetsarkitektur och konstruktionsprinciper dokumenteras och redovisas. *Kommentar*: Industrin ska presentera en säkerhetsarkitektur enligt avsnitt 4.3 vilken redovisas i Systemspecifikation /System, Subsystem Specification (SSS).
2.801.10-A	Konstruktionsprinciperna skall fastlägga vilka strategier för felupptäckt, feltolerans och felsäkerhet som tillämpas. *Kommentar*: Redogörelsen ska ange valda konstruktionsprinciper med motiveringar för gjorda val.

Krav nr	Innehåll
2.801.11-A	Konstruktionsbeslut avseende vald säkerhetsarkitektur skall dokumenteras och inkludera förutsättningar, antaganden samt motiveringar för valda konstruktionsalternativ.

Avsnitt 8.1.4 Krav på utvecklingsverktyg

Krav nr	Innehåll
2.801.12-A	Verktyg för kravspårning skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för kravspårning enligt IEC 12207.
2.801.13-A	Verktyg för konfigurationshantering skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för konfigurationshantering enligt IEC 12207.
2.801.14-A	Verktyg för avvikelserapportering skall användas och vara överenskommet med FMV. *Kommentar*: Verktyget bör uppfylla processkraven för felrapportering enligt IEC 12207.
2.801.15-A	FMV skall ges tillgång till information för kravspårning, konfigurationsledning, avvikelserapportering och testdata. *Kommentar*: FMV behöver se till att förutsättningar finns för att kunna hantera och läsa informationen.

Avsnitt 8.1.5 Krav på dokumentation

Krav nr	Innehåll
2.801.16-A	En dokumentlista skall överenskommas med FMV. *Kommentar*: Definieras utifrån dokumentlistan i *kapitel 9*. Leveransplan för dokumentation ska finnas.

Avsnitt 8.2.1 Krav på systemsäkerhetsanalys

Krav nr	Innehåll
2.802.01-A	Spårbarhet skall finnas mellan datorsystem och dess inverkan på det tekniska systemets identifierade vådahändelser. *Kommentar*: Kravspårning åt båda håll kan redovisas i säkerhetsarkitekturarbetet.
2.802.02-A	Systemsäkerhetsanalysen skall redovisa kritikalitetsnivån för ingående programvaror i det tekniska systemet. *Kommentar*: Avser analys av säkerhetsarkitektur.
2.802.03-A	Val av säkerhetsarkitektur skall motiveras utifrån analys av de dimensionerade vådahändelserna.
2.802.04-A	Oberoende granskningar och genomgångar skall utföras under utvecklingen enligt överenskommen systemsäkerhetsplan (SSPP). *Kommentar*: Med oberoende granskare avses person som inte medverkat i utvecklingsarbetet.
2.802.05-A	Data skall ha den kritikalitetsklassificering som krävs i det aktuella tekniska systemet. *Kommentar*: Med data avses både statisk och realtidsgenererad information.

Krav nr	Innehåll
2.802.06-T	Val av säkerhetsfunktioner och funktionsövervakningar skall göras på sådant sätt att detta inte i onödan komplicerar programvarusystemet. *Kommentar*: En avvägning bör alltid göras så att de centrala säkerhetsprinciperna såsom enkelhet, oberoende och determinism uppnås.
2.802.07-T	Etablerat programspråk skall användas vid utveckling av säkerhetskritisk programvara. *Kommentar*: Valda programmeringsspråk ska redovisas för FMV tillsammans med konstruktionsprinciper och säkerhetsarkitektur.
2.802.08-T	För varje operativt tillstånd skall det tekniska systemet kunna inta ett säkert tillstånd. *Kommentar*: För initial kritikalitetsklassificering ***LÅG*** avses säkert tillstånd där styrningar av verkställande delar har kommenderats/avslutats på ett säkert sätt eller där ett räddningssystem har övertagit kontrollen.
2.802.09-T	Alla feltillstånd som kan påverka systemets funktion skall loggas i ett utvärderingsbart format. *Kommentar*: Spårbarhet ska finnas mellan utlösande felsituation/felkriterium och det tillstånd det tekniska systemet intagit så att fel kan hittas i datorsystemet. Loggning kan ske internt i datorsystemet eller loggas i externt system.
2.802.10-T	Det tekniska systemet skall vara i ett säkert tillstånd under uppstart. *Kommentar*: Detta omfattar även omstart av datorsystemet.

Krav nr	Innehåll
2.802.11-T	Vid uppstart av det tekniska systemet skall programvaran kontrollera att definierat säkert tillstånd har intagits innan kritiska delar aktiveras. *Kommentar*: Säkerhetsnivån kan kontrolleras med hjälp av återläsning på kritiska styr- eller givarsignaler.
2.802.12-T	Orimliga indata, som enligt systemsäkerhetsanalysen kan påverka systemets funktion, skall detekteras och omhändertas så att vådahändelse inte inträffar. *Kommentar*: Med orimliga data menas alla data utanför definierat värdeområde eller data vid fel tidpunkt.
2.802.13-T	Operatörsåtgärder och presenterad information, som berör säkerhetskritiska funktioner, skall registreras. *Kommentar*: Valet av sätt att registrera kan variera utifrån systemkonfiguration, komplexitet och situation.
2.802.14-T	Built In Test (BIT) skall innehålla Säkerhetskontroll (SK/PBIT) under uppstart, Funktionsövervakning (FÖ/CBIT) under drift och Manuellt initierad test/Funktionskontroll (FK/IBIT) vid underhåll.
2.802.15-T	BIT-funktioner för uppstart och underhåll skall inte kunna aktiveras oavsiktligt under operativ drift av systemet. *Kommentar*: Säkerhetsfunktion, exempelvis blockering, ska finnas så att handhavandefel kan undvikas.
2.802.16-T	Oberoende watchdog-funktion skall aktiveras innan datorsystemet kan genomföra kritiska styrningar.¨ *Kommentar*: Oberoende watchdog-funktion realiseras lämpligen i hårdvara.

Krav nr	Innehåll
2.802.17-T	Watchdog (WD) skall ha ett definierat tidsfönster (det vill säga min/max tid för WD-trigg). *Kommentar*: Återställning av watchdog utförs av programvaran.
2.802.18-T	Watchdog (WD) skall omfattas av Säkerhetskontroll (SK/PBIT) vid uppstart och godkänt resultat skall utgöra kriterium för att kunna aktivera watchdog-funktionen.
2.802.19-T	Spänningsövervakning skall ske kontinuerligt på datorsystemets matningsspänningar. *Kommentar*: Kontrollsignal från spänningsövervakningen kan utgöra ett av kriterierna i watchdog-funktionen.
2.802.20-T	Resursutnyttjande vid första serieleverans skall vara definierat. *Kommentar*: Kravet avser CPU, minne och kommunikationslänkar och bör vara högst 50%.

Avsnitt 8.2.3 Krav på programvaruutvecklingsmiljö

Krav nr	Innehåll
2.802.21-A	Val av programvaruutvecklingsmiljö skall motiveras och dokumenteras för det tekniska systemet. *Kommentar*: Branschstandarder och tidigare erfarenheter ska beaktas utifrån vald kritikalitetsnivå.
2.802.22-A	Revisionshistorik skall redovisas för använd utvecklingsmiljö. *Kommentar*: Utvecklingsmiljön ska konfigurationsstyras under programvarans hela livscykel.

Krav nr	Innehåll
2.802.23-A	Vid uppdatering av utvecklingsmiljö under utvecklingen av programvaran skall omverifiering ske av både utvecklingsmiljön och utvecklad programvara. *Kommentar*: Tillvägagångssätt och kriterier beskrivs i SSPP eller i annat överenskommet dokument.
2.802.24-A	Testverktyg som inför ändringar i programvaran skall inte användas vid verifiering av en fastställd programvaruversion. *Kommentar*: Om modifieringar är nödvändig för att testverktyget ska kunna användas så ska dessa ändringar ses som del av programversionen.

Avsnitt 8.2.4 Krav på verifiering

Krav nr	Innehåll
2.802.25-A	Systemsäkerhetsprovning skall planeras, utföras och granskas samt brister som upptäcks skall åtgärdas samt godkännas. *Kommentar*: Resultatet presenteras och eventuella identifierade åtgärder överenskoms med FMV.
2.802.26-A	Testfall vid systemsäkerhetsprovning skall genomgå en oberoende granskning av person som inte har medverkat i utvecklingen.
2.802.27-A	Systemsäkerhetsprovning skall utföras på en fastställd systemversion i det tekniska systemet. *Kommentar*: Med fastställd systemversion avses den version av det tekniska system som ska levereras, det vill säga även målmiljön ska ha fastställd status.

Krav nr	Innehåll
2.802.28-A	Systemsäkerhetsprovning skall omfatta felinjicering i samtliga gränsytor av de i systemsäkerhetsanalyserna identifierade säkerhetskritiska signalerna. *Kommentar*: Systemsäkerhetsprovningens avsikt är att visa att funktionsövervakningen kan detektera kritiska fel.
2.802.29-A	Systemsäkerhetsprovning skall visa att funktioner endast avsedda för visst driftläge/systemtillstånd inte kan nås under annat driftläge/systemtillstånd. *Kommentar*: Beakta även felaktigt handhavande och drifttillstånd såsom utbildning och underhåll.
2.802.30-A	Maximalt resursutnyttjande av datorsystemet skall verifieras och dokumenteras. *Kommentar*: Kravet avser CPU, minne och kommunikationslänkar.
2.802.31-A	Verifiering skall ske av att exekvering sker i rätt ordning och vid rätt tidpunkt i tidskritiska funktioner. *Kommentar*: Verifiering av exekveringsordning kan även utföras med hjälp av utvecklingsmiljön.
2.802.32-A	Testtäckning (BIT) av säkerhetsfunktioner i det tekniska systemet skall verifieras.
2.802.33-A	Använda kriterier för Proven in use skall vara överenskomna med FMV. *Kommentar*: Kriterierna dokumenteras i SDP (Software Development Plan).

Avsnitt 8.3 Krav inför leverans till FMV

Krav nr	Innehåll
2.803.01-A	Lista över kvarstående kända fel skall redovisas för levererad version av det tekniska systemet. *Kommentar*: Anges i Software Version Description (SVD) enligt dokumentlista
2.803.02-A	Industrin skall, trots kvarvarande kända fel, visa att det tekniska systemet ändå uppfyller Försvarsmaktens krav på tolerabel risknivå.

Avsnitt 8.4 Krav vid systemuppdatering

Krav nr	Innehåll
2.804.01-A	Vid ny version av det tekniska systemet skall omverifiering genomföras. *Kommentar*: Omverifieringsbehovet avgörs efter analys av vilka delar som är påverkade av ändringen.
2.804.02-A	I samband med systemuppdatering skall ett nytt systemsäkerhetsutlåtande ges ut. *Kommentar*: Avser inte ändringsbara parametrar.

Definitioner och ordförklaringar

Följande definitioner och ordförklaringar används i handboken. Ett antal definitioner är handbokens egna och dessa är särskilt markerade med ”H ProgSäk 2018” i kolumn ”Referens” (handbokens egen definition).

För andra använda termer hänvisas bland annat till Terminologicentrum TNC och ISO/IEC/IEEE 24765:2010 *Systems and software engineering - Vocabulary*.

Term	Referens	Definition/Förklaring
Avvikelsehantering	H ProgSäk 2018	Processen om hur fel ska rapporteras in från kund, klassificeras och som resulterar i en eller flera problemrapporter som redovisar hur avvikelsen omhändertas.
CE-märkning	Wikipedia	Produktmärkning inom EES. Bokstäverna CE är en förkortning för Conformité Européenne. En produkt med CE-märkning får säljas i EES-området utan ytterligare krav.
Continuous mode	IEC 61508	Where the safety function retains the EUC in a safe state as part of normal operation.
Data	H ProgSäk 2018	Avser information, ofta lagrad som filer eller databaser, som programvaran använder då den exekverar eller genererar annan information.
Datorsystem	H ProgSäk 2018	Innehåller hårdvara, programvara och data.
ECE-reglemente	Transportstyrelsen	ECE-reglementen är bilagor till 1958-års överenskommelse om att anta enhetliga tekniska föreskrifter för hjulförsedda fordon eller för utrustning och delar som kan monteras eller användas på sådana fordon.
EUC	IEC 61508	Equipment under control, EUC, equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities.

Term	Referens	Definition/Förklaring
Etablerad kodningsföreskrift	H ProgSäk 2018	Vedertagen normsamling som har en bred användning, och främst inom sin bransch. Kodningsföreskrift innehåller: • regler för tillåtna och otillåtna programvarukonstruktioner • regler för märkning, kommentering och namngivning av kritiska delar • anvisningar för minimering av komplexitet • restriktioner på grund av problem i kompilator eller målsystem med mera • detaljerade regler för säker konstruktion i använt lågnivåspråk.
Etablerad programvarustandard	H ProgSäk 2018	Internationellt vedertagen programvarustandard som har en bred användning inom sin bransch och som uppdateras vid behov.
Etablerad standard	H ProgSäk 2018	Internationellt vedertagen standard som har en bred användning inom sin bransch och som uppdateras vid behov.
Farligt tillstånd	H SystSäk 2011	En fysisk situation som kan leda till en olycka.
Felorsak	H SystSäk 2011	Omständighet som lett till uppkomst av fel.
Funktion	H ProgSäk 2018	Innehåller hårdvara såsom datorsystem, brytare, strömförsörjning, verkställande delar och sensorer.
Funktionalitet	Svenska dataterm-gruppen	Förmåga hos en produkt att kunna utföra de funktioner som den är konstruerad för.
Funktionssäkerhet	SS 441 05 05	Förmågan hos en enhet att utföra en krävd funktion under givna förhållanden under ett givet tidsintervall.
Granskning	H SystSäk 2011	Syftar till att på ett kvalitetssäkrat och spårbart sätt granska främst teknisk dokumentation.

Term	Referens	Definition/Förklaring
High demand mode	IEC 61508	Se IEC 61508.
Low demand mode	IEC 61508	Se IEC 61508.
Nödsystem	H ProgSäk 2018	System som säkerställer att de viktigaste funktionerna bibehålls vid farligt fel i det tekniska systemet.
Oberoende granskning	H ProgSäk 2018	Granskning utförd av person som inte deltagit i utvecklingen. Flera nivåer av oberoende finns. Oberoende kan kräva person från annan organisation.
Olycka	H SystSäk 2011	Inträffar då någon/något exponeras för vådahändelse eller farligt tillstånd och därvid skadas (skada på person, egendom eller yttre miljö). Olyckan är alltid oplanerad, inte resultatet av till exempel fientlig handling.
Problemrapport	H ProgSäk 2018	Intern informationsbärare för hantering av registrerade avvikelser.
Programvara	H ProgSäk 2018	Innehåller programvaruinstruktioner, data och dokumentation för datorsystem.
Programvaruutvecklingsmiljö	H ProgSäk 2018	En omgivning för programvaran som utgörs av ett datorprogram för att utveckla, producera, modifiera, analysera och testa ett annat program och dess dokumentation. Denna kan även inkluderar funktioner för konfigurationshantering och kravspårning av programvara och dess dokumentation.
Proven in use	H ProgSäk 2018	Programvara som har använts i liknande tekniska system med bevis på korrekt funktion och har spårbar användningshistorik (*service history*).

Term	Referens	Definition/Förklaring
Rattmärke	Transportstyrelsen	Utrustning som regleras av direktivet för marin utrustning ska rattmärkas och inte CE-märkas. Rattmärket visar att produkten uppfyller kraven enligt direktivet. Typgodkännande (rattmärkning) av utrustning regleras genom ett EU-direktiv. I direktivet fastställs gemensamma regler i syfte att undanröja skillnader vid genomförandet av internationella standarder genom att ha en tydligt identifierad uppsättning krav och enhetliga certifieringsförfaranden.
Regressionstestmiljö	H ProgSäk 2018	Testmiljö för programvara som möjliggör automatisk test och utvärdering av programfunktioner. Regressionstest används för testa hela eller delar av det tekniska systemet sedan ändringar har införts. Detta för att säkerställa att systemet fungerar som tidigare och att inte nya problem har uppstått som följd av införda ändringar.
Risk	H SystSäk 2011	Avser risk för skada på människa, egendom och/eller yttre miljö. Uttrycks som funktion av sannolikheten för att olycka inträffar och dess konsekvens (konsekvensen vanligen fördelad på de fyra skadeklasserna för människa respektive ekonomi).
Riskkälla	H SystSäk 2011	Något som kan leda till skada på person, egendom eller yttre miljö.
Räddningssystem	H ProgSäk 2018	System som säkerställer att säkert tillstånd för det skyddsvärda kan intas vid farligt fel i det tekniska systemet.
Sammanlagd drifttid	H ProgSäk 2018	Antal timmar som ett tekniskt system används.
SCRUM	Wikipedia	SCRUM är en metodik för systemutveckling skapad av Jeff Sutherland och Ken Schwaber. Ordet ”SCRUM” kommer från rugbyn och är ett moment när bollen sätts i spel.

Term	Referens	Definition/Förklaring
Skadeklass	H SystSäk 2011	För personskada: Dödsfall, allvarlig personskada, mindre allvarlig personskada och försumbar skada. För ekonomisk skada: Jämförbart med total systemförlust, betydande förlust, begränsad förlust, liten förlust. Detaljer framgår av *H SystSäk del 1, avsnitt 4.2.3*.
Skyddsfunktion	H ProgSäk 2018	Funktion för att nå eller upprätthålla ett säkert tillstånd i den styrda utrustningen.
System		Se *Tekniskt system*.
Systematiska fel	H SystSäk 2011	Ett fel som alltid inträffar vid viss användning av system och som ger samma felutfall varje gång. Orsaken kan till exempel vara logiskt fel i programvara som ger samma felutfall vid exekvering.
Systemsäkerhet	H SystSäk 2011	Egenskapen hos ett tekniskt system att inte oavsiktligt orsaka skada på person, egendom eller yttre miljö.
Systemsäkerhetsverksamhet	H SystSäk 2011	Det totala arbete som bedrivs under ett tekniskt systems hela livscykel i syfte att identifiera, analysera, värdera och åtgärda olycksrisker.
Systemuppdatering	H ProgSäk 2018	Installation av uppdaterad systemversion enligt leverantörens anvisningar, inklusive kontroll av att installationen blivit korrekt införd.
Säkerhetsarkitektur	H ProgSäk 2018	Metod för att minska datorsystemets kritikalitet i ett säkerhetskritiskt tekniskt system.
Säkerhetsfunktion	H ProgSäk 2018	Tillförd funktion vars syfte är att reducera sannolikheten för att vådahändelse ska inträffa vid fel i den säkerhetskritiska funktionen.
Säkerhetskritiskt datorsystem	H ProgSäk 2018	Datorsystem som direkt eller indirekt styr eller övervakar energier och som vid fel kan orsaka vådahändelse och i förlängningen olyckor.

Term	Referens	Definition/Förklaring
Säkerhetskritisk funktion	H ProgSäk 2018	Funktion som styr eller övervakar energier och som vid fel kan leda till vådahändelse och i förlängningen olyckor. Kommentar: Funktion kan innehålla både hård- och programvara.
Säkert tillstånd	H ProgSäk 2018	Tillstånd där styrningar av verkställande delar har kommenderats/avslutats på ett säkert sätt eller där ett nöd- eller räddningssystem har övertagit kontrollen.
Tailorisering	H ProgSäk 2018	Val och anpassning av verksamhet och/eller dokumentation.
Tekniskt system	H ProgSäk 2018	Utgörs av komponenter, förbrukningsmateriel och programvaror samt instruktioner och övrig produktinformation, organiserade för att uppnå ett eller flera uttalade syften i en given omgivningsmiljö.
Testverktyg	H ProgSäk 2018	Del av utvecklingsverktyg som används vid funktionstest/verifiering av programvara.
Testtäckningsgrad för kod	H ProgSäk 2018	Anger hur väl har man har testat programvarukoden, att alla krav har testats och att alla delar i koden har testats. Mäts efter att koden har utvecklats.
Testtäckningsgrad för inbyggd test	H ProgSäk 2018	Anger hur stor del av hårdvarufunktionerna eller hårdvarans möjliga felfunktioner i datorsystemet som den inbyggda testen BIT klarar av att hitta. BIT är en programvara som körs samtidigt som övrig operationell programvara i datorsystemet.
Tidskritiska data	H ProgSäk 2018	Data där informationens ålder måste vara känd då den är av vital betydelse.
Tillbud	H SystSäk 2011	Vådahändelse som inte leder till olycksfall eftersom ingenting exponeras vid vådahändelsen.

Term	Referens	Definition/Förklaring
Tillförlitlighet	H ProgSäk 2018	Innebär att ett tekniskt system tillhandahåller en viss funktion med en viss sannolikhet över tiden eller då funktionen efterfrågas.
Tolerabel risknivå	H ProgSäk 2018	Ett av Försvarsmakten/FMV angivet läge som lägst uppfyller lagens krav på betryggande säkerhet utifrån givna förutsättningar.
Utvecklingsmiljö	H ProgSäk 2018	Utrustningar som krävs för framtagning (såsom kompilatorer och länkare), verifiering av programvaran, riggar, simulatorer, utrustning för dataförsörjning och konfigurationsledning.
Verksamhetssäkerhet	H SystSäk 2011	Försvarsmaktens verksamhetssäkerhet avser Försvarsmaktens förmåga att hantera risker vid all verksamhet så att författningsenliga krav på arbetsmiljö och säkerhet för Försvarsmaktens personal samt kraven på säkerhet för tredje man, yttre miljö och egendom uppfylls.
Vådahändelse	H SystSäk 2011	Händelse som inträffat av våda, det vill säga utan uppsåt, oplanerat och som kan resultera i olycka eller tillbud om någon eller något exponeras.

Akronymer/förkortningar

Följande akronymer och förkortningar används i handboken.

Akronym/förkortning	Förklaring
AFS	Arbetsmiljöverkets författningssamling
ANS	Air Navigation Services
AOP	Allied Ordnance Publication
AQAP	The Allied Quality Assurance Publications
ASA	Aircraft Safety Assessment
ASIC/PLD	Application Specific Integrated Circuits/Programmable Logic Devices
ASIL	Automotive Safety Integrity Level
ATEX	ATEX är en förkortning av det franska namnet på ett av direktiven, Appareils destinés à être utilisés en ATmosphères EXplosibles (Explosionsskydd)
BIT	Built-In-Test
BOA	Beslut om användning
CBIT	Continuous-Built-In-Test (Funktionsövervakning)
CCF	Common Cause Failures (Gemensam felorsak)
CE	Conformité Européenne (Europeisk konformitet)
CEN	European Committee for Standardization
CENELEC	Committee for Electrotechnical Standardization
CM	Configuration Management (Konfigurationsledning)
CODABA	Collaborative Database
COTS	Commercial off the Shelf
CSSB	Centralt Systemsäkerhetsbeslut
DAL	Design Assurance Level
DC	Diagnostic Coverage (Feldetekteringsförmåga)
DoC	Declaration of Conformity (Deklaration om överensstämmelse)
EASA	European Aviation Safety Agency (Europeiska byrån för luftfartssäkerhet)
ECE	Economic Commission for Europe (Ekonomiska kommissionen för Europa)

Akronym/förkortning	Förklaring
EDA	European Defence Agency
EMC	Elektromagnetisk kompatibilitet
ENNSA	European Network of National Safety Authorities on Ammunition
ETSI	European Telecommunications Standards Institute
EUC	Equipment Under Control
EUROCAE	European Organization for Civil Aviation Equipment
FAA	Federal Aviation Administration (Amerikanska luftfartsmyndigheten)
FAT	Factory Acceptance Test
FC	Functional Check, se även Funktionskontroll (FK)
FDAL	Function Development Assurance Level
FHA	Functional Hazard Analysis
FHA	Functional Hazard Assessment
FK	Funktionskontroll, se även Functional Check (FC)
FM	Functional Monitoring (Funktionsövervakning)
FORTV	Fortifikationsverket
FPGA	Field-Programmable Gate Array
FRA	Försvarets radioanstalt
FSA	Functional Safety Assessment
FTA	Felträdsanalys
FVL	Full Variability Language (Programspråk som inte har begränsat språkomfång)
FÖ	Funktionsövervakning, se även FM
GKPS	Grundkrav programvarusäkerhet
GOTS	Government Off The Shelf
H SystSäk	Handbok Systemsäkerhet
H VAS	Handbok Vapen- och ammunitionssäkerhet
HFI	Human Factors Integration
HFT	Hardware Fault Tolerance
HKV	Försvarsmakten Högkvarteret
HW	Hardware (Hårdvara)

Akronym/förkortning	Förklaring
IBIT	Initiated BIT
IDAL	Item Development Assurance Level
IEC	International Electrotechnical Commission
IMA	Integrated Modular Avionics
IRS	Interface Requirement Specification (Gränsytespecifikation)
ISD	Informationssäkerhetsdeklaration
ISO	International Organization for Standardization
ISO/TC	ISO Technical Committee
LOR	Level Of Rigor
LRU	Line Replaceable Units
LVD	Lågspänningsdirektivet
M	Mandatory (Obligatorisk)
MIL-STD	Military Standard
MOTS	Military Off The Shelf
MoU	Memorandum of Understanding
MCS	Minimal Cut Set (Minimala hindermängder)
MSIAC	Munitions Safety Information Analysis Center
$MTTF_d$	Mean Time To dangerous Failure
NOTS	Nato Off The Shelf
NR	Not Recommended (Inte rekommenderat)
NSPA	NATO Support and Procurement Agency
OM	Other Measures
OSS	Open Source Software (Programvara med öppen källkod)
PASA	Preliminary Aircraft Safety Assessment
PBIT	Power on BIT
PDS	Previously Developed Software
PEN	Platform och Procurement Experts Network
PL	Performance Level
PLC	Programmable Logic Controller (Programmerbara styrsystem)

Akronym/förkortning	Förklaring
PSAC	Plan for Software Aspects of Certification (Certifieringsplan för programvara)
PSSA	Preliminary System Safety Assessment
QM	Quality Management (Kvalitetsledning)
R	Recommended (Rekommenderat)
RAMS	Reliability, Availability, Maintainability and Safety (Tillförlitlighet, funktionssannolikhet, driftsäkerhet, tillgänglighet, underhållsmässighet och säkerhet)
RTCA	Radio Technical Commission for Aeronautics
SAE	Society of Automotive Engineers
SC	Severity Category
SC	Safety Check (Säkerhetskontroll)
SCA	Safety Compliance Assessment (Systemsäkerhetsutlåtande)
SCC	Software Control Category
SCMP	Software Configuration Management Plan (Konfigurationsledningsplan programvara)
SDD	Software Design Document (Detaljerad design programvara)
SDP	Software Development Plan (Utvecklingsplan programvara)
SEK	Svenska Elektriska Kommittén (Svensk nationalkommitté av IEC)
SFF	Safe Failure Fraction (felsäkerhetskvot)
SFS	Svensk författningssamling
SHA	Safety Hazard Analysis
SIL	Safety Integrity Level
SIRT	Systems Integration Requirements Task
SIS	Swedish Standards Institute
SL	Software Level
SOP	Start Of Production
SQA Plan	SQA Plan Software Quality Assurance Plan (Kvalitetsplan för programvara)
SQA Record	SQA Records Software Quality Assurance Records (Kvalitetssäkringsrapport programvara)

Akronym/förkortning	Förklaring
SRASW	Safety-Related Application Software
SRCF	Safety-Related Control Functions
SRECS	Safety-Related Electrical Control Systems
SRESW	Safety-Related Embedded Software
SRP/CS	Safety-Related Parts/Control Systems
SRS	Software Requirement Specification (Specifikation Programvarukrav)
SSA	System Safety Assessment
SSG	Systemsäkerhetsgodkännande (Safety Statement)
SSHA CS	Sub System Hazard Analysis Computer System (Systemsäkerhetsanalys Datorsystem)
SSLR	Software Safety Lifecycle Requirements
SSMP	System Safety Management Plan (Systemsäkerhetsledningsplan)
SSPP	System Safety Program Plan (Systemsäkerhetsplan)
SSRS	Software Safety Requirements Specification
SSS	System, Subsystem Specification (Systemspecifikation)
SSTD	System Safety Test Description (Provprogram för systemsäkerhetsprovning)
SSTR	System Safety Test Record (Systemsäkerhetsprovningsrapport)
SSWG	System Safety Working Group (Arbetsgrupp för systemsäkerhet)
STANAG	Standard NATO Agreement
STD	Software Test Description (Provprogram för programvara)
STR	Software Test Record (Testrapport programvara)
SVD	Software Version Description Document (Versionsbeskrivning av levererad systemversion)
SVP	Software Verification Plan (Verifieringsplan programvara)
SVR	Software Verification Record (Verifieringsrapport för programvara)
SW	Software (Programvara)
SWAL	Software Assurance Level

Akronym/förkortning	Förklaring
SwCI	Software Criticality Index
TNC	Terminologicentrum
TPLS	Third Party Logistic Support
TRR	Test Readiness Review
TS	Teknisk specifikation (Technical Specification)
UTC	Universal Time Coordinated
VÅS	Verksamhetsåtagandespecifikation (Statement of Work, SoW)

Referenser

Följande dokument utgör källdokument till handboken. Angivna dokumentbeteckningar är de som var aktuella vid handbokens färdigställande. I de fall där en viss referens behöver tillämpas, rekommenderas att förekomsten av senare utgåva kontrolleras.

Titel, dokument
AFS 1998:05, Arbete vid bildskärm
AOP-15 edition 3 (2009), Guidance On The Assessment Of The Safety And Suitability For Service Of Non-nuclear Munitions For Nato Armed Forces
AOP-52 edition 1 (2009), Guidance On Software Safety Design And Assessment Of Munition-related Computing Systems
Def Aust 5679 (2006), The Procurement of Computer-Based Safety-Critical Systems
Def Stan 00-56 edition 4, Safety Management Requirements for Defence Systems
DoD, Joint Software Systems Safety Engineering Handbook Version 1.0 Published August 27, 2010
ED-153, Guidelines for ANS Software Safety Assurance
EG-direktiv 2014/90/EU om marin utrustning är satt i kraft genom lagen (2016:768) om marin utrustning och förordningen (2016:770) om marin utrustning tillsammans med Transportstyrelsens föreskrifter (TSFS 2016:81) om marin utrustning
EN 50126, Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)
EN 50128, Railway applications - Communication, signaling and processing systems - Software for railway control and protection systems
EN 50129, Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling
EN 62061, Maskinsäkerhet – Funktionssäkerhet hos elektriska, elektroniska och programmerbara elektroniska säkerhetskritiska styrsystem
EN ISO 13849-1 Maskinsäkerhet – Säkerhetsrelaterade delar av styrsystem – Del 1: Allmänna konstruktionsprinciper
FM Handbok Informationssäkerhet (H Säk Infosäk), M7739-352056
FM Handbok Systemsäkerhet (H SystSäk 2011, del 1), M7739-352022
FM Handbok Systemsäkerhet (H SystSäk 2011, del 2), M7739-352023
FMV Handbok i användbarhet (H HFI)

Titel, dokument
FMV Handbok ISD IT-säkerhet Användningsfall och arkitektur
FMV Handbok ISD IT-säkerhet Management
FMV Handbok ISD IT-säkerhet Oberoende granskning
FMV Handbok Vapen- och ammunitionssäkerhet 2012, M7762-000871
Handbok Försvarsmaktens säkerhetstjänst, Informationssäkerhet (H Säk Infosäk 2013)
IEC 60601, Elektrisk utrustning för medicinskt bruk
IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
IEC 61511, Functional safety – Safety Instrumented Systems for the Process Industry Sector
ISO 15289, Systems and Software Engineering – Content of Life-cycle Information Items
ISO 26262, Road vehicles - Functional safety
ISO 9001:2015 – Ledningssystem för kvalitet
ISO/IEC 12207, System- och programvarukvalitet – Livscykelprocesser för programvara
ISO/IEC 15288, System- och programvarukvalitet – Livscykelprocesser för system
ISO/IEC 15504, Information Technology – Process Assessment
ISO/IEC 27000 Ledningssystem inom informationssäkerhet
ISO/IEC/IEEE 24765:2010, Systems and software engineering – Vocabulary
MIL-STD 1472G (2012), Department OF Defense Design Criteria Standard: Human Engineering
MIL-STD 882E, System Safety
NASA Software Safety Guidebook (NASA-STD-8719.13)
RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification
RTCA DO-254, Design Assurance Guidance for Airborne Electronic Hardware
SAE ARP4754A, Aerospace Recommended Practice - Guidelines for Development of Civil Aircraft and Systems
STANAG 4187 Fuzing Systems - Safety Design Requirements
STANAG 4452, Safety assessment requirements for munition related computing systems

Bilaga 1 Jämförelser mellan programvarustandarder

På följande sidor finns jämförelsetabeller för utvalda programvarustandarder avseende vissa aspekter.

Tabell B1:1 Administrativa aspekter

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Sektor/tillämpningsområde	Programmerbara elektriska system	Vägfordon	Maskinstyrning	Maskinstyrning	Flyg (SW)	Programmerbar logik (HW)	Flyg (system)	Flyg	Järnväg	Processindustri
Aktuell utgåva (utgivningsår)	2010	2011	2015	2015	2011	2000	2010	2009	2011	2016
Antal delar i standarden	7	10	2	1	1	1	1	1	1	3

Tabell B1:2 Kritikalitetsklassificering

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Grund för klassning	Allvarlighet, sannolikhet	Allvarlighet, exponering, kontrollerbarhet	Allvarlighet, frekvens, möjlighet att undvika	Allvarlighet, sannolikhet	Allvarlighet	Allvarlighet	Allvarlighet	Allvarlighet, sannolikhet	Allvarlighet, frekvens (enligt exempel)	Allvarlighet, sannolikhet
Metodik för klassning	Riskgraf	Riskgraf	Riskgraf	Tabell	Bedömning allvarlighet	Bedömning allvarlighet	Bedömning allvarlighet	Riskgraf	Riskgraf	Flera metoder i IEC 61511-3
Nivåer för klassning	SIL 1–4	ASIL A–D	PL a–e	SIL 1–3	Level A–E	DAL A–E	Level A–E	SWAL 1–4	SIL 0–4	SIL 1–4
Högsta nivå	SIL 4	ASIL D	PL e	SIL 3	Level A	DAL A	Level A	SWAL 1	SIL 4	SIL 4

Tabell B1:3 Teknisk omfattning

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Syfte	Säkerhets-funktion	Produkt	Säkerhets-funktion	Säkerhets-funktion	Produkt	Produkt	Produkt	Produkt	Produkt	Produkt
Adresserar system	Ja	Ja	Ja	Ja	Nej	Nej	Ja	Till liten del	Nej	Ja
Adresserar SW	Ja	Ja	Ja	Ja	Ja	Nej	Nej	Ja	Ja	Ja
Adresserar HW	Ja	Ja	Ja	Ja	Nej	Ja	Nej	Nej	Nej	Ja
Adresserar informationssäkerhet	Nej	Nej	Nej	Nej	Nej	Nej	Nej	Ja, till viss del	Nej	Ja, till viss del
Standarden tillräcklig i sig själv	Ja	Ja	Ja	Ja	I princip om bara SW ingår	I princip om bara HW ingår	Nej	Ja		Ja
Adresserar underleverantör	Nej	Ja	Nej	Nej	Ja	Nej	Nej	Ja	Ja	Ja, IEC 61511-1, avsnitt 5.2.5
Harmoniserad mot EU-direktiv	Nej	Nej	Ja	Ja	Nej	Nej	Nej	Nej	Nej	Nej
Certifiering	Ja	Ja	Ja	Ja	Ja, om DO178C, DO-254, ARP4754A tas med	Ja, om DO178C, DO-254, ARP4754A tas med	Ja, om DO178C, DO-254, ARP4754A tas med	Ja	Nej	Nej

Tabell B1:4 Tekniker och metoder

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Metod för riskanalys	Ja	Ja	Ja	Ja	Nej	Nej	Nej	Ja	Ja, som exempel	Ja, IEC 61511-1, kapitel 8. Flera metoder i IEC 61511-3
Metod för att bestämma allvarlighet	Ja	Ja	Ja	Ja	Nej	Nej	Ja	Nej	Nej	Flera metoder i IEC 61511-3
Krav: *shall* eller *should*	*Shall*	*Shall*	*Shall*	*Shall*	*Should*	*Should*	*Should*	*Shall*	*Shall* och några *should*	*Shall*
Terminologi till viss del anpassad till standard	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
V-modell för utveckling	Ja	Ja	Ja	Ja	Nej, möjlig men anges inte	Ja	Ja	Nej, möjlig men anges inte	Ja, som exempel	Ja
Krav på personer och/eller organisation	Ja	Ja	Nej	Nej	Nej	Nej	Nej	Ja	Ja	Ja, IEC 61511-1, kapitel 5
Hel livscykel definierad (SW, och HW)	Ja	Ja	Ja	Ja	Nej, bara SW	Nej, bara HW	Ja	Stöd finns för SW men ingen definierad	Nej, finns i EN 50126	Ja

Tabell B1:5 Metodik

	IEC 61508	ISO 26262	EN ISO 13849-1
Ställer standarden särskilda krav på kompetens hos företaget?	Nej, bara krav på oberoende hos de som utför *Functional Safety Assessment*. Se del 1, tabell 4-5.	Ja, se del 2, avsnitt 5.4.2 som behandlar *Safety culture*.	´Nej
Ställer standarden särskilda krav på kompetens hos individen?	Ja, del 1, avsnitten 6.2.13–15, 8.2.9 samt 8.2.14	Ja, se del 2, avsnitt 5.4.3.	Nej
Finns det en rutin för oberoende granskning av programvara?	Nej, men omnämns som krav att den ska genomföras, se del 3, avsnitt 7.9.2.12	Ja, se del 2, avsnitt 6.4.7. Kraven på oberoende stiger med ökande ASIL.	Nej, men valideringen ska utföras av personer som är oberoende av konstruktionen, se del 2, avsnitt 4.1.
Finns det en särskild metodik för arkitekturarbete?	Ja, se avsnitt 3 7.4.3 och dessutom finns viss information i del 7, Annex F	Ja, standarden föreskriver systemkonstruktion som innebär krav på systemarkitektur. Se del 4, avsnitt 7.4. Arkitektur för programvara beskrivs i del 6, kapitel 7.	Standarden arbetar med systemarkitektur och definierar fem olika kategorier, men har inga krav på arkitektur för programvara.
Beskriver standarden principer för uppbyggnad av programvara eller ger den förslag på detaljerade konstruktionslösningar?	Ställer krav, men ger inga konstruktionslösningar.	Ja, tekniker och metoder föreskrivs beroende på ASIL.	Ja, beskriver uppbyggnad och krav på programvara i avsnitt 4.6.

	IEC 61508	ISO 26262	EN ISO 13849-1
Finns det metodik för hur integrering av PDS-programvara ska ske?	Nej, men trusted/verified software elements finns med som aspekt i del 3, tabell A.2 och A.4.	Ja, kvalificering av programvarukomponenter beskrivs i del 8, kapitel 12.	Nej
Finns krav på hur konfigurationsledning och kvalitetssäkring av programvara ska utföras?	Ja, se del 1, avsnitt 6.2 och del 3, avsnitt 6.2.	Ja, se del 8, kapitel 7.	Standarden har inga krav för detta.

	EN 62061	RTCA/DO 178C	RTCA/DO-254
Ställer standarden särskilda krav på kompetens hos företaget?	Nej	Nej	Nej
Ställer standarden särskilda krav på kompetens hos individen?	Nej	Nej, bara en aspekt beträffande testframtagning i avsnitt 6.2e.	Nej
Finns det en rutin för oberoende granskning av programvara?	Nej	Ja	Appendix A ger vägledning kring oberoende i verifieringsprocessen. All verifiering på Level A och B bör vara oberoende. Level C eller lägre nivåer kräver inte oberoende verifiering.
Finns det en särskild metodik för arkitekturarbete?	Standarden behandlar systemarkitektur i avsnitt 6.6.2.1 och programvaruarkitektur i avsnitt 6.11.3.3.	Nej, bara allmänna principer.	Ja, avsnitt 2.3 *Hardware Safety Assessment* behandlar aspekter som är viktiga för arkitekturval.

	EN 62061	RTCA/DO 178C	RTCA/DO-254
Beskriver standarden principer för uppbyggnad av programvara eller ger den förslag på detaljerade konstruktionslösningar?	Beskriver uppbyggnad och krav på programvara i avsnitt 6.11.3.	Nej	Standarden beskriver aktiviteter som bör utföras för att uppnå målen, snarare än att detaljera hur själva konstruktionen bör se ut.
Finns det metodik för hur integrering av PDS-programvara ska ske?	Nej	Ja, avsnitt 12.1 beskriver *Use of Previously Developed Software*.	Ja, se avsnitt 11.2.
Finns krav på hur konfigurationsledning och kvalitetssäkring av programvara ska utföras?	Ja, se avsnitt 6.11.3.2.	Konfigurationshantering finns angivet i chapter 7 *Software Configuration Management process*. Verifiering av källkod anges allmänt i chapter 6 *Software Verification Process*.	Ja, se kapitel 7.

	ARP 4754A	ED-153	EN 50128
Ställer standarden särskilda krav på kompetens hos företaget?	Nej	Ja, se tabell A.9.	Ja, i viss mån i *Training Process* och i *Improvement Process*.
Ställer standarden särskilda krav på kompetens hos individen?	Nej	Ja, baserade på roller, se avsnitt 5.2 och Annex B.	Ja, se *Training Process*

	ARP 4754A	ED-153	EN 50128
Finns det en rutin för oberoende granskning av programvara?	Nej	Nej, men krav finns i avsnitt 7.5 *Component implementation and testing*	Nej, nämns bara allmänt i *Verification Process*.
Finns det en särskild metodik för arkitekturarbete?	Nej	Nej	Nej
Beskriver standarden principer för uppbyggnad av programvara eller ger den förslag på detaljerade konstruktionslösningar?	Nej	Nej	Nej
Finns det metodik för hur integrering av PDS-programvara ska ske?	Nej	Nej, men krav finns (betecknas *pre-existing SW*, *COTS* eller *open source SW*).	Ja, både krav och rådgivning finns i avsnitt 7.2.
Finns krav på hur konfigurationsledning och kvalitetssäkring av programvara ska utföras?	Konfigurationshantering finns beskriven på många ställen och specifikt i avsnitt 5.6 Kvalitetssäkring för hela systemet med hjälp av verifiering och validering, vilka finns beskrivna i kapitel 5	*Software Configuration Management Plan* ska tas fram. Konfigurationskrav på vad som ska hanteras finns utspridda på många ställen. Det finns också en roll definierad, *Configuration Manager Role Specification*, med angivet ansvar och kompetens, se tabell B.10. Avsnitt 6.5 *Software Quality Assurance* gäller för kvalitetssäkring och dokument *Software Quality Assurance Plan* samt *Software Quality Assurance Verification Report* ska tas fram.	Se *Configuration Management Process* för konfigurationshantering. *Quality Assurance Process* finns som ger övergripande krav för kvalitetssäkring.

	IEC 61511
Ställer standarden särskilda krav på kompetens hos företaget?	Ja, se IEC 61511-1, kapitel 5
Ställer standarden särskilda krav på kompetens hos individen?	Ja, se IEC 61511-1, kapitel 5
Finns det en rutin för oberoende granskning av programvara?	Det finns ingen rutin, men oberoende förutsätts både för verifiering (IEC 61511-1, kapitel 7) och *SIS Safety Validation* (IEC 61511-1, kapitel 15).
Finns det en särskild metodik för arkitekturarbete?	Nej
Beskriver standarden principer för uppbyggnad av programvara eller ger den förslag på detaljerade konstruktionslösningar?	Nej
Finns det metodik för hur integrering av PDS-programvara ska ske?	Nej, men krav på val av komponenter finns i avsnitt 11.5 av IEC 61511-1
Finns krav på hur konfigurationsledning och kvalitetssäkring av programvara ska utföras?	Ja, konfigurationsledning av programvara ingår i *SIS Configuration Management* (IEC 61511-1, avsnitt 5.2.7)

Bilaga 2 Mall för FMV:s FHA (Functional Hazard Analysis)

Syftet med att genomföra en FHA är att identifiera systemsäkerhetsrelaterade konstruktionsinriktade krav att ställa i Teknisk specifikation (TS). Projektledaren vid FMV genomför FHA och redovisar resultatet för Försvarsmakten. Detta är en anpassad FHA för detta syfte.

Olycksrisk/Topphändelse	Användningsfas	**Orsakad av:** • **Utebliven funktion** • **Nedsatt funktion** • **Felaktig funktion** • **Oavsiktligt aktiverad**	Konsekvens givet att olycksrisken inträffar	Förslag på riskminskande åtgärd	Konstruktionsinriktade krav för att förhindra olycksrisken
Person-, egendoms- och miljöskada pga att egen luftfarkost oavsiktligt skjuts ner	Strid	Orsakad av utebliven funktion hos luftfarkostens identifikationsutrustning (IFF)	Dödsfall, egendomsförlust och begränsad miljöskada orsakat av att egen luftfarkost störtar	Dubbla sensorsystem hos luftvärnssystemet för att säkerställa identiteten hos luftfarkosten	Luftfarkosten skall kunna identifieras på två av varandra oberoende sätt Luftvärnssystemet skall ha minst 50% säker identifikation av luftfarkosten innan eld kan avges
…					

Bilaga 3 Exempel på FMV:s initiala kritikalitetsklassificering och kravställning

Grunder

Försvarsmakten ställer krav på systemsäkerhet, kompletterat med krav på tolerabel risknivå för det tekniska systemet, i aktuell Materielmålsättning. Krav på tolerabel risknivå används inte som grund för *FMV:s initiala kritikalitetsklassificering*, utan den används i systemsäkerhetsarbetet för att senare kunna återredovisa kravuppfyllnad inför överlämning av tekniskt system till Försvarsmakten. Beträffande Försvarsmaktens krav på systemsäkerhet, inklusive krav på tolerabel risknivå för det tekniska systemet, ofta uttryckt i en riskmatris, hänvisas till metodiken i H SystSäk.

Av Materielmålsättningen ska även den tänkta driftprofilen framgå. Driftprofilen används för att beräkna den sammanlagda drifttiden under hela livslängden för det tekniska systemet. Drifttiden är av betydelse för den utvecklande industrins krav på avsaknaden av farliga fel i säkerhetskritiska funktioner, se *bilaga 4*.

FMV behöver dels identifiera värsta tänkbara konsekvenser för person, egendom och yttre miljö, dels den sammanlagda drifttiden för det tekniska systemet. Nedan presenteras ett fiktivt exempel och den tänkta arbetsgången från mottagandet av Materielmålsättning till FMV:s förfrågningsunderlag till utvecklande industri.

Ingångsvärden

Försvarsmakten avser att beställa ett nytt tekniskt system och överlämnar därför en Materielmålsättning till FMV. FMV bedömer att det tekniska systemet kommer att innehålla säkerhetskritiska datorsystem. I detta exempel kommer datorsystemet att styra och övervaka funktioner som innehåller stora energimängder. Försvarsmakten ställer krav på att tolerabel risknivå för

enstaka dödsfall (skadeklass I, enligt H SystSäk) för viss olycka får vara maximalt 1×10^{-6} /systemindivid under livslängden. Se markering i *bild B3:1*.

Person		A	B	C	D	E	F
0	Flertalet dödsfall	ET	ET	ET	ET	ET	T
I	Enstaka dödsfall	ET	ET	ET	ET	T	T
II	Allvarlig skada	ET	ET	ET	T	T	T
III	Mindre allvarlig skada	ET	T	T	T	T	T
IV	Obetydlig skada	T	T	T	T	T	T

Bild B3:1 Exempel på Försvarsmaktens krav på tolerabel risknivå för personskada

Motsvarande krav på tolerabel risknivå ställs för egendom och yttre miljö. Notera dock att tolerabel risknivå även kan uttryckas på andra sätt än ovan, exempelvis att CE-märkning är tillräcklig.

FMV:s initiala kritikalitetsklassificering av tekniskt system

FMV genomför en förenklad *Functional Hazard Analysis* (FHA) för att identifiera dimensionerande olyckshändelser, så kallade topphändelser, se *bilaga 2*. Det tekniska systemets topphändelser blir tillika de vådahändelser som utvecklande industri ska arbeta med i sin säkerhetsarkitektur. FMV redovisar resultatet av FHA för Försvarsmakten. Detta ger underlag för Försvarsmaktens beslut om vilket alternativ som ska realiseras och beställas som ett utvecklingsuppdrag hos FMV. Resultatet av FHA och Försvarsmaktens kravdokument, ger underlag för de principiella konstruktionsinriktade systemsäkerhetskraven, som kan infogas i FMV:s Tekniska specifikation (TS).

Efter Försvarsmaktens beslut om vilket alternativ som ska realiseras beställs ett utvecklingsuppdrag hos FMV.

Om det tekniska systemets topphändelser kan leda till höga, allvarliga eller medelstora konsekvenser för person, egendom och/eller yttre miljö, enligt tillämpningsmatrisen i *bild B3:2*, blir resultatet av ***FMV:s initiala kritikalitetsklassificering "HÖG"***. Detta medför att det av FMV:s förfrågningsunderlag ska framgå att industrin ska uppfylla både kraven i en valfri etablerad programvarustandard för utveckling av systemsäkerhetskritisk programvara, och *Grundkrav för programvarusäkerhet* (GKPS) i *kapitel 8*.

Om de värsta konsekvenserna av det tekniska systemets topphändelser enbart kan resultera i låga eller inga konsekvenser blir resultatet av FMV:s initiala kritikalitetsklassificering "***LÅG***". Detta medför att det av FMV:s förfrågningsunderlag ska framgå att industrin enbart ska uppfylla *Grundkrav för programvarusäkerhet* (GKPS) i *kapitel 8*.

Tillämpningsmatris kopplad till MIL-STD 882E
för FMV:s initiala kritikalitetsklassificering av tekniska system

Nivå avseende konsekvens	Beskrivning	Genomförande	FMV:s initiala kritikalitets-klassificering
Hög	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***katastrofal*** konsekvens för person, ekonomi och/eller miljö *(flera eller enstaka dödsfall, total systemförlust och/eller bestående miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för högsta kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	***HÖG*** FMV ställer krav på industrin att etablerad programvarustandard ska följas.
Allvarlig	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***kritisk*** konsekvens för person, ekonomi och/eller miljö *(allvarliga och bestående personskador, omfattande ekonomisk och/eller miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för högre kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	
Medel	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***allvarlig*** konsekvens för person, ekonomi och/eller miljö *(allvarliga men inte bestående personskador, betydande ekonomisk och/eller miljöskada).*	Överenskommen programvarusäkerhetsstandard tillämpas och krav för medelhög kritikalitet tillämpas. FMV:s krav på dokumentation uppfylls.	
Låg	Tekniskt system innehållande säkerhetskritisk programvara där konsekvensen av olycka medför ***marginell*** konsekvens för person, ekonomi och/eller miljö *(mindre allvarlig personskada, mindre ekonomisk och/eller miljöskada).*	Grundkrav för utveckling av programvara för lägsta tolerabla kritikalitetsnivå tillämpas (GKPS).	***LÅG*** FMV ställer krav på industrin att lägst GKPS ska användas. *(Industrin kan dock välja att följa en etablerad programvarustandard)*
Ingen	Tekniskt system innehållande programvara där konsekvensen av olycka medför negligerbar konsekvens för person, ekonomi och/eller miljö.	Grundkrav för utveckling av programvara för lägsta tolerabla kritikalitetsnivå tillämpas (GKPS).	

Bild B3:2 Tillämpningsmatris för FMV:s initiala kritikalitets-klassificering av programvara

I detta exempel bedöms konsekvenserna för personskada vid en olycka bli höga. FMV kommer i förfrågningsunderlaget att ställa krav på att *Grundkrav Programvarusäkerhet* (GKPS) följs samt att industrin anger vilken etablerad programvarustandard som man avser följa vid utvecklingen av det tekniska systemet.

FMV:s initiala kritikalitetsklassificering dokumenteras i FMV:s systemsäkerhetsplan (SSPP).

Beräkning av den sammanlagda drifttiden

Av Materielmålsättningen framgår den tänkta driftprofilen för det tekniska systemet. Driftprofilen kan uttryckas på olika sätt och den kan dessutom vara villkorad för exempelvis internationella insatser. Nedan ges ett exempel på hur driftprofilen kan uttryckas i en Materielmålsättning.

> ”Det tekniska systemet ska ha en livslängd om 20 år (bör 30 år). Under 1 års drift bedöms driftprofilen fördelas till 50% insatser, 20% övning, 10% utbildning, 15% förrådsställning och 5% underhållsåtgärder. Det tekniska systemet ska kunna förvaras i förråd i 2 år (bör 4 år).
>
> Under fred bedöms ett utbildningsår motsvara cirka 9 månaders drift. Körsträcka cirka 4000 km/år. Skjutning cirka 600 skott/år. Under internationell insats bedöms körsträckan till cirka 8000 km/år. Skjutning cirka 1 000 skott/år.”

Utifrån ovanstående exempel på driftprofil, uppskattar FMV den sammanlagda drifttiden för det tekniska systemet till maximalt 10 000 timmar under 30 års livslängd. FMV ska dokumentera sina beräkningar som ett tillämpat dimensioneringsunderlag i avsett dokument, i syfte att till Försvarsmakten kunna återredovisa uppfyllande av tolerabel risknivå.

Kravställning i förfrågningsunderlag

Utifrån ovanstående exempel anger FMV följande i förfrågningsunderlaget:

- Krav på tolerabel risknivå för enstaka dödsfall (skadeklass I, H SystSäk) får vara maximalt 1×10^{-6} per systemindivid under livslängden 30 år.
- *Grundkrav för programvarusäkerhet* (GKPS) i H ProgSäk 2018 ska uppfyllas.
- Valfri etablerad programvarustandard, tillämplig inom teknikområdet, ska anges och uppfyllas (FMV ska inte ange kritikalitetsnivå i förfrågningsunderlaget).
- Driftprofilen motsvarar en sammanlagd drifttid om minst 10 000 timmar för det tekniska systemet.

Bilaga 4 Exempel på industrins arbetsgång inför kontrakt

Industrins mottagning av förfrågningsunderlag

Av FMV förfrågningsunderlag, enligt exempel i bilaga 3, framgår att:

- Krav på tolerabel risknivå för enstaka dödsfall (skadeklass I, H SystSäk) får vara maximalt 1×10^{-6} per systemindivid under livslängden 30 år.
- *Grundkrav för programvarusäkerhet* (GKPS) i H ProgSäk 2018 ska uppfyllas.
- Valfri etablerad programvarustandard, tillämplig inom teknikområdet, ska anges och uppfyllas.
- Driftprofilen motsvarar en sammanlagd drifttid om minst 10 000 timmar för det tekniska systemet.

Industrins anbud till FMV

Då FMV enligt Lagen om offentlig upphandling (LOU), är skyldiga att göra en formell utvärdering enligt likabehandlingsprincipen mellan olika anbud, måste industrin svara att man uppfyller samtliga ställda skall-krav enligt ovan.

I detta exempel väljer utvecklande industri programvarustandard IEC 61508. Motivet för valet grundar sig på att industrin redan arbetar enligt IEC 61508 och har goda erfarenheter av att tillämpa standarden. I industrins anbud till FMV anges att IEC 61508 kommer att tillämpas samt att kraven enligt *Grundkrav för programvarusäkerhet* (GKPS) kommer att uppfyllas. Dessutom kommer kravet på tolerabel risknivå och sammanlagd drifttid att uppfyllas. Om FMV har begärt ytterligare detaljering i förfrågningsunderlaget bifogas detta.

Kontraktskrivning mellan FMV och industrin

FMV lägger beställning på industrin utifrån ställda krav i förfrågningsunderlaget.

Kontraktsgenomgång mellan FMV och industrin

Inför den formella kontraktsgenomgången mellan FMV och industrin görs vissa förberedelser.

Industrin tar fram ett mer detaljerat koncept av det tekniska systemet och genomför en inledande systemsäkerhetsanalys i syfte att identifiera vådahändelser. I detta arbete utgår industrin från de mest kritiska vådahändelserna för skadeklass I (katastrofal konsekvens för person, ekonomi och/eller miljö) som kan inträffa i det tekniska systemet. Om skadeklass I inte kan inträffa används istället skadeklass II. En mer detaljerad konstruktion av det tekniska systemets säkerhetsarkitektur kan nu tas fram.

I kravnedbrytningen, utifrån kravställd tolerabel risknivå, identifieras de delar som kommer att styra kritikalitetsnivån på datorsystemet utifrån den tänkta säkerhetsarkitekturen.

Industrin analyserar tillämpningen av GKPS i det detaljerade konceptet. Industrin motiverar också val av kritikalitetsnivå enligt metodik i vald programvarustandard.

Industrin bedömer att ett av kraven i GKPS inte är tillämpbart.

”2.802.09-T Alla feltillstånd som kan påverka systemets funktion skall loggas i ett utvärderingsbart format.”

Motivet är att eftersom det säkerhetskritiska datorsystemet planeras som ett inbyggt system baserat på en 32-bitars microcontroller med begränsat internminne är möjligheterna till loggning begränsade.

Vid den formella kontraktsgenomgången överenskoms att krav 2.802.09-T inte är realiserbart i det föreslagna tekniska lösningen utan kommer istället överens om hur en alternativ metod för loggning med hjälp av extern loggutrustning som kan anslutas vid felsökning. Industrin motiverar också valet av SIL-nivån enligt metodiken i standarden IEC 61508.

För tekniska system som ska certifieras av myndighet är det lämpligt att industrin, inför kontraktsgenomgången, tar fram *Certifieringsplan* (PSAC) och *Acceptansplan* (PSAA).

Förtydliganden och överenskommelser vid kontraktsgenomgången dokumenteras i ett protokoll som signeras av båda parter. I de fall överenskommelserna bedöms påverka kontraktet så ska en kontraktsändring genomföras.

Nedan presenteras alternativa systemlösningar på säkerhetsarkitekturer vilka reducerar kritikalitetsnivån på datorsystemet.

Arbetsgång vid definition av säkerhetsfunktioner

Vid konstruktionen av säkerhetsfunktioner bör enkelhet samt kända och beprövade teknologier företrädesvis användas. Om säkerhetsfunktionen kan realiseras med delsystem där stor erfarenhet finns sedan tidigare och där felmoder och felfrekvenser är kända så underlättar detta även vid verifiering av ställda krav.

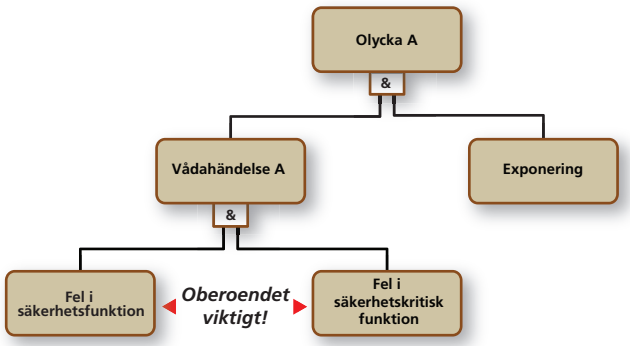


Bild B4:1 Generellt felträd för kravnedbrytning av vådahändelse

Målet med kravnedbrytning, av sannolikheten för Vådahändelse A, är att definiera en lämplig säkerhetsfunktion så att en så stor del av kravet som möjligt allokeras på säkerhetsfunktionen. Vid högt ansatt krav på den säkerhetskritiska funktionen leder detta till en högre kritikalitetsklassificering för denna del, vilket påverkar både stringensen i tillämpad utvecklingsmetodik och verifiering av det ansatta kravet. Ur verifieringssynpunkt är det i många fall en bättre strategi att ansätta så stor del som möjligt av det nedbrutna kravet till säkerhetsfunktionen.

Typexempel (a)

Här nedan ges ett exempel där en säkerhetsfunktion införs i det tekniska systemet så att kravet på fel i säkerhetskritisk funktion A kan reducerar till en nivå så att GKPS kan tillämpas på denna del i det tekniska systemet.

I de inledande analyserna har *Vådahändelse A* identifierats som det farliga fel i det tekniska systemet som vid exponering leder till *Olycka A*.

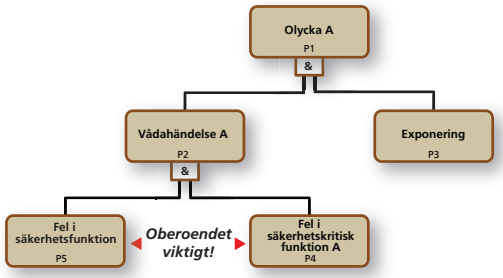


Bild B4:2 Generellt felträd för att beskriva relationer i olycksmodellen

Ingångskravet på tolerabel risknivå för enstaka dödsfall (skadeklass I) får maximalt vara 1×10^{-6} per systemindivid under livslängden 30 år.

Sannolikheten för exponeringen sätts konservativt = 1 (P3), det vill säga sannolikheten för *Olycka A* (P1) = sannolikheten för *Vådahändelse A* (P2).

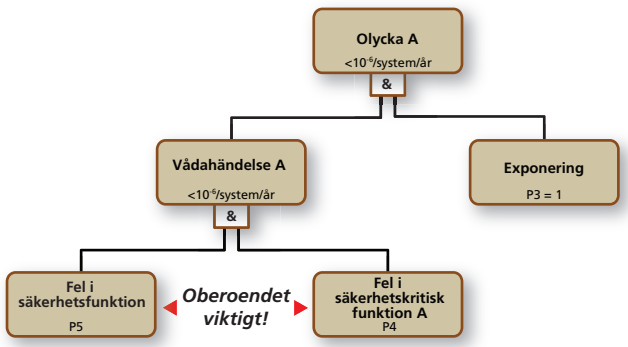


Bild B4:3 Kravnedbrytning av acceptabel sannolikhet för vådahändelse vid exponering =1

Baserat på FMV:s krav på en livslängd om 30 år beräknas den sammanlagda drifttiden av det tekniska systemet motsvara upp till 10 000 timmars drifttid (vilket motsvarande cirka 1 års kontinuerlig drift). Detta innebär att kravet på tolerabel risknivå för

enstaka dödsfall ska vara < 1×10^{-6} per systemindivid/år eller < 1×10^{-10} per systemindivid/timme (10 000 timmar motsvarar cirka 1 år).

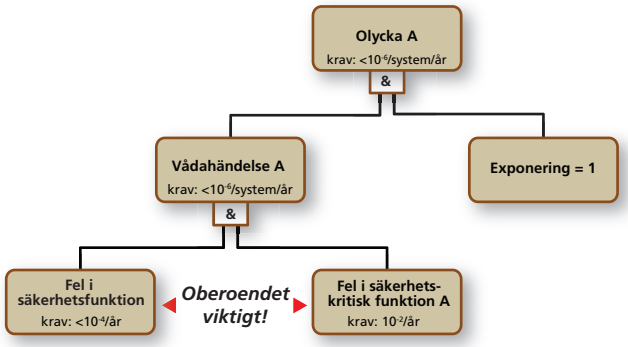


Bild B4:4 Exempel på kravnedbrytning

Om säkerhetsfunktionen vid kravnedbrytningen i *bild B4:4* ansätts en felsannolikhet < 10^{-4} systemindivid/år (< 10^{-8}/timme) då måste den säkerhetskritiska funktionen A klara resterande del, det vill säga en felsannolikhet < 10^{-2} systemindivid/år (< 10^{-6}/timme).

Observera att oberoendet mellan de två huvudgrenarna i felträdet måste beaktas så att felsannolikheterna för de både grenarna kan multipliceras, det vill säga sannolikheten för Vådahändelse A < 10^{-6} = ($10^{-4} \times 10^{-2}$).

Om den säkerhetskritiska funktionen A delas upp i två oberoende säkerhetskritiska funktioner (A1, A2) enligt *bild B4:5* så måste det samtidigt vara farligt fel i de båda kanalerna A1 och A2 för att farligt fel ska uppstå i den säkerhetskritiska funktionen A. På detta sätt kan det nedbrutna kravet på säkerhetskritisk funktion A i det ideala fallet brytas ned på två av varandra oberoende delfunktioner A1 och A2.

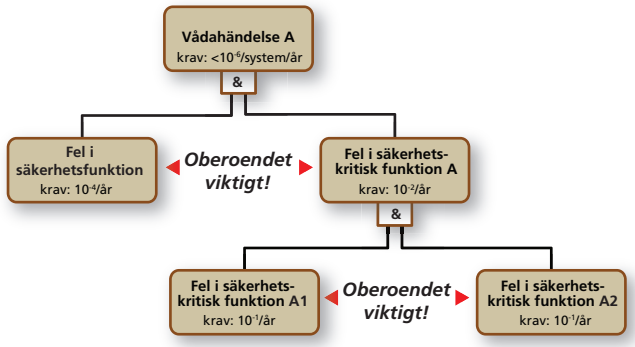


Bild B4:5 Kravnedbrytning i säkerhetskritiskt fler-kanaligt diversifierat system

Om istället en en-kanalig säkerhetsarkitektur används så måste kravnedbrytningen fördelas så att säkerhetsfunktionen tar en större del av det nedbrutna kravet.

Därför ansätts < 10^{-5}/år på säkerhetsfunktion och resterande < 10^{-1}/år på säkerhetskritisk funktion, se *bild B4:6* nedan.

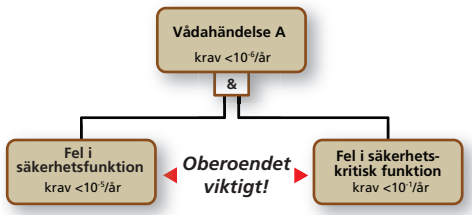


Bild B4:6 Exempel kravnedbrytning i säkerhetskritiskt system, en-kanaligt

Observera att ingångskravet på det tekniska systemet inte har förändrats utan enbart den del som kan orsaka farliga fel i säkerhetskritisk funktion.

I båda dessa exempel är det nedbrutna kravet på slumpmässigt farligt fel i säkerhetskritisk funktion < 10^{-1}/år (eller < 10^{-5}/timme). Enligt *tabell B4:1*så inryms denna kravnedbrytning så att initial kritikalitetsklassificering **LÅG** och GKPS bedöms vara tillräckligt att uppfylla.

Tillämpning av GKPS för kontinuerlig drift

Tabell B4:1 Omräkningstabell för lägsta ansatta felsannolikhet för slumpmässiga fel

System i kontinuerlig drift Sammanlagd drifttid under livslängden	Lägsta tillåtna ansatta sannolikhet för fel i säkerhetskritisk funktion för kritikalitetsnivå ***LÅG***
≤ 100 h	1×10^{-3} (p)
< 500 h	5×10^{-3} (p)
< 1 000 h	1×10^{-2} (p)
< 5 000 h	5×10^{-2} (p)
< 10 000 h	1×10^{-1} (p) (1 års kontinuerlig drift = 8 760 h) (1 år ~ 10 000 h)
< 50 000 h	5×10^{-1} (p)
≥ 100 000 h	= 1

Typexempel (b)

I detta exempel används en en-kanaligt säkerhetskritisk funktion, det vill säga utan redundans men med bashändelser enligt beskrivning i *avsnitt 4.3.4*. Detta är också tillämpbar i varje gren under *säkerhetskritisk funktion* i det fler-kanaliga exemplet enligt *bild B4:5*.

I *bild B4:7* har den fortsatta kravnedbrytningen gjorts utifrån det tidigare ansatta kravet för fel i *säkerhetskritisk funktion A* på $< 10^{-1}$/år (1 år ~ 10 000 h). Kravet har fördelats lika på respektive bashändelse (*fel i ställdon* eller *fel i givare* eller *fel i säkerhetskritiskt datorsystem*).

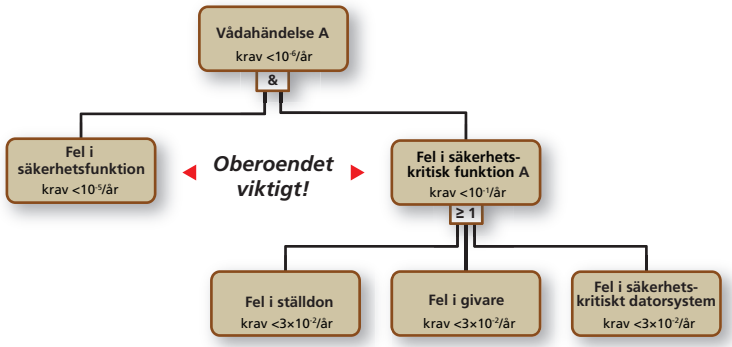


Bild B4:7 Reducerat generiskt felträd, en-kanaligt säkerhetskritiskt system

Felsannolikheten för säkerhetskritisk funktion A approximeras till 10^{-1} genom att summera de underliggande bashändelserna.

I nästa steg, för att ytterligare reducera sannolikheten för *vådahändelse A*, tillförs funktionsövervakning/diagnostik av *säkerhetsfunktion* respektive *säkerhetskritiska funktion A:s ställdon* och *givare*. Övervakningens syfte är att detektera slumpmässiga fel för att på så sätt ytterligare kunna reducera det nedbrutna kravet på det säkerhetskritiska datorsystemet A. Felträdet enligt *bild B4:7* utökas då enligt *bild B4:8* nedan.

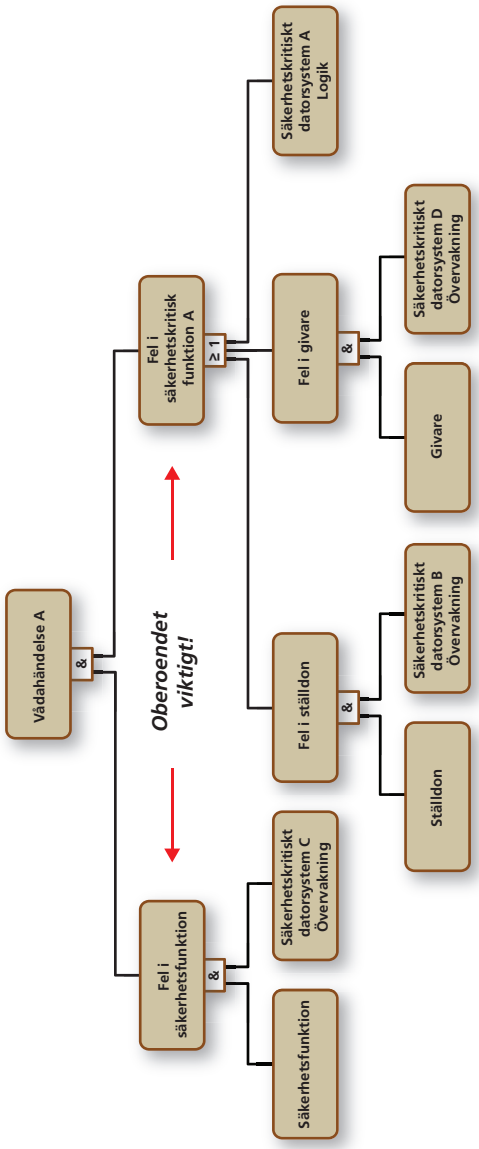


Bild B4:8 Reducerat generiskt felträd för ett en-kanaligt säkerhetskritiskt system med oberoende övervakning

En förutsättning är att den tillförda övervakningen i *Datorsystem C* kan anses vara oberoende av fel i det säkerhetskritiska *Datorsystemet A, B och D*. Om den tillförda övervakningen av säker-

hetsfunktionen (*Säkerhetskritiskt datorsystem C, Övervakning*) ansätts till 10^{-1}/år så kan kravfördelning på *Säkerhetsfunktion* enligt *bild B4:8* reduceras till < 10^{-4}/år.

Då *Datorsystem C* ligger inom initial kritikalitetsklassificering ***LÅG***, det vill säga 10^{-1}/år (1 år ~ 10 000 h) är krav enligt GKPS tillräckliga för utformningen av *Datorsystem C* övervakning.

Definieras sedan på samma sätt funktionsövervakning av ställdon och givare via *Datorsystem B och D* övervakning som också ansätts till initial kritikalitetsklassificering ***LÅG***, det vill säga med en felsannolikhet på 10^{-1}/år så kan bidraget från dessa två grenar i felträdet försummas i förhållande till farligt fel i *Säkerhetskritisk Datorsystem A*, det vill säga kravet på fel i *Säkerhetskritisk funktion A* kan allokeras helt på *Säkerhetskritisk Datorsystem A*, det vill säga 10^{-1}/år och därmed initial kritikalitetsklassificering ***LÅG*** och GKPS tillämpliga även för denna del.

Observera att kraven enligt GKPS omfattar både krav på hårdvara för att reducera sannolikheten för slumpmässiga fel samt krav på stringensen i utvecklingsprocessen av programvara för att reducera de systematiska felen. Kravnedbrytningen av sannolikhet för vådahändelse är endast giltig för de slumpmässiga felen. GKPS initial kritikalitetsklassificering ***LÅG*** definierar den minsta delmängd av de krav vilka ska motverka införandet av systematiska fel för denna nivå.

Bilaga 5 Exempel på FMV:s kravuppfyllnadsmall

Projektledaren vid FMV ska senast inför leverans till Försvarsmakten säkerställa att kraven i kontraktet med utvecklande industri är uppfyllda.

Kravnr	Krav	Uppfyllt	Ej uppfyllt	Hur är det uppfyllt?	Referens/underlag	Anmärkning/övrigt
2.801.03-A	Utvecklande industri skall utse en kontaktperson för programvarusäkerhet.	×		Titel/roll, förnamn och efternamn.	Protokoll xxx från kontraktsgenomgång.	
2.802.07-T	Etablerat programspråk skall användas vid utveckling av säkerhetskritisk programvara.	×		Utvecklande industri använder C++.	Protokoll xxx från kontraktsgenomgång.	
...						

Checklistan finns i digital utgåva på FMV hemsida. Huruvida ett krav är uppfyllt eller inte, eller om det inte är tillämpligt, kan anges i filens kravuppfyllnadskolumn (Ja/ Nej/ Inte tillämpligt).

Bild- och tabellförteckning

Bilder

Teckningar

Tabeller

Projektledare
Lars Lange, FMV

Områdesexperter
Björn Koberstein, FMV
Mikael Lindbergh, FMV
Peter Djervbrant, Peter Djervbrant AB

Teckningar
Stefan Gustafsson, UTBLICK MEDIA I HALLAND AB

Illustrationer och omslag
Mats Lundgren, Combitech AB

Original
Mats Lundgren, Combitech AB

Digital utgåva
Mats Lundgren, Combitech AB

Försvarets Materielverk
115 88 STOCKHOLM

M7762-001041