



Certified Professional for Requirements Engineering

Foundation Level

Kursplan

Stan Bühne

Martin Glinz

Hans van Loenhoud

Stefan Staal

Villkor

1. Individer och utbildningsanordnare får använda denna kursplan som bas för seminarier, under förutsättning att upphovsrätten erkänns och ingår i seminariematerialet. Alla som använder denna kursplan i reklamsyfte behöver skriftligt medgivande från IREB e.V. för detta ändamål.
2. Varje individ eller grupp av individer får använda denna kursplan som bas för artiklar, böcker eller andra härledda publikationer under förutsättning att upphovsrätten för författare och IREB e.V. som källa och ägare av detta dokument erkänns i sådana publikationer.

© IREB e.V.

Alla rättigheter förbehållna. Ingen del av denna publikation får återges, lagras i söksystem eller överföras i någon form eller på något sätt, elektroniskt, mekaniskt, genom fotokopiering, inspelning eller på annat sätt, utan antingen skriftligt tillstånd av författarna eller IREB e.V.

Erkännande

Denna kursplan skapades ursprungligen 2007 av: Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl and Chris Rupp. De har fått stöd av Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer och Joy Beatty.

Version 3.0 är en större revision skapad av Stan Bühne, Martin Glinz, Hans van Loenhoud, och Stefan Staal. De har fått stöd av Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp, and Camille Salinesi.

I arbetet med denna version har feedback getts av Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek, och Thorsten Weyer. Ytterligare feedback gavs av Wim Decoutere och Hans-Jörg Steffe.

Granskningar utfördes av Christoph Ebert, Barbara Paech och Chris Rupp.

Översatt av Elizabeth Bjarnason, Beata Karpinska och Daniel Säter. Granskat av Tobias Ahlgren.

Godkänd för utgivning den 22 juli 2020 av IREB-rådet på rekommendation av Xavier Franch och Frank Houdek.

Vi tackar alla för deras hedersamma engagemang.

Copyright © 2007 – 2022 för kursplanen med författarna som angivits ovan. Rättigheterna har överförts till International Requirements Engineering Board e.V. (IREB), Karlsruhe, Germany.

Inledning

Sommaren 2017 genomförde vi en studie för att undersöka relevansen av den befintliga Certified Professional for Requirements Engineering (CPRE) certifiering på grundnivå (version 2.2). Syftet med undersökningen var att få feedback om certifieringens praktiska

marknadsrelevans från utbildningsleverantörers perspektiv såväl som från CPRE-certifierade som arbetar som kravingenjörer [MFeA2019]. Undersökningen avspeglade att den befintliga CPRE-grundplanen v2.2 fortfarande i allmänhet tillgodoser marknadens viktigaste behov och förmedlar relevanta kravkunskaper till kandidaterna. Trots detta fick vi feedback om att flera tekniker inte längre används i praktiken, medan andra saknas i sökandet efter mer iterativ och adaptiv utveckling. Denna feedback var i linje med IREB:s egen uppfattning om förändringarna inom området kravhantering. Vi har därför beslutat att göra en större revision av CPRE-Foundation Level, ta bort föråldrat innehåll och lägga till nya element. Version 3 av kursplanen återspeglar läget för kravhantering och täcker både plandrivna och agila metoder för att specificera och hantera krav.

Kandidater som vill erhålla CPRE-certifikat enligt denna kursplan förväntas ha grundläggande kunskaper i systemutveckling med plan-driven och agilt tillvägagångssätt.

Syftet med dokumentet

Kursplanen definierar grundnivån av certifieringen Certified Professional for Requirements Engineering som fastställts av International Requirements Engineering Board (IREB). Kursplanen ger kursleverantörer grunden för att skapa sina kursmaterial. Eleverna kan använda kursplanen för att förbereda sig inför examineringen.

Kursplanens innehåll

Foundation Level möter behoven hos alla personer som deltar i ämnet kravhantering. Detta inkluderar personer i roller som kravingenjör, verksamhetsanalytiker, systemanalytiker, produktägare eller chefer, utvecklare, projekt- eller IT-chefer eller domänexperter.

I denna kursplan kan förkortningen "RE" (eng. RE = Requirement Engineering) för kravhantering förekomma.

Innehållets omfattning

CPRE Foundation Level kommunicerar grundläggande principer som kan appliceras inom alla områden och för olika typer av system (exempelvis mobila applikationer, inbyggda system, säkerhetskritiska system, och administrativa system). CPRE Foundation Level förutsätter inte någon specifik utvecklingsprocess och är inte heller inriktad på en specifik applikationsdomän. Utbildningsleverantörer kan erbjuda utbildning som fokuserar på specifika typer av system, processer eller tillämpningsdomäner, så länge utbildningsmålen för denna kursplan är fullständigt täckta.

Detaljnivå

Detaljnivån i denna kursplan tillåter internationellt överensstämmande undervisning och examination. För att nå detta mål innehåller kursplanen följande:

- Generella utbildningsmål
- Innehåll med en beskrivning av utbildningsmålen
- Referenser till ytterligare litteratur (vid behov)

Könsanknutna formuleringar

Vi har avsiktligt avstått från att använda könsanknutna formuleringar i detta dokument.

För oss på IREB är det självklart viktigt att använda korrekt könsanknutna formuleringar. Dock anser vi det nödvändigt att formulera komplicerade teman och frågor på ett sätt som enkelt kan förstås.

Texter som kräver en maskulin eller feminin form kan vara mindre läsbara och därmed svårare att förstå. Detta dokument syftar dock till att presentera och kommunicera innehållet på ett precist och tydligt sätt. Då vårt mål är att hjälpa läsare att fokusera på innehållet har vi enbart och avsiktligt använt den maskulina formen av individer i dokumentet. Detta skall dock inte ses som en brist på respekt.

Utbildningsmål/Kognitiva kunskapsnivåer

Alla delar och utbildningsmål i denna kursplan tilldelas en kognitiv nivå. Följande kognitiva nivåer används i denna kursplan:

- **L1: Kunna** (beskriva, räkna upp, känneteckna, känna igen, namnge, komma ihåg, ...) – kunna, komma ihåg, känna till, ett tidigare lärt material.
- **L2: Förstå** (förklara, tolka, slutföra, sammanfatta, motivera, klassificera, jämföra, ...) – greppa/konstruera förståelse från givet material eller situationer.
- **L3: Tillämpa** (specificera, skriva, designa, utveckla, implementera, ...) – tillämpa kunskap och skicklighet i givna situationer.



En högre nivå inkluderar de lägre nivåerna. Observera, att alla termer som definieras i ordlistan måste vara kända (L1), även om de inte uttryckligen nämns i utbildningsmålen. Ordlistan är tillgänglig för nedladdning på IREB: s hemsida <https://www.ireb.org/downloads/#cpre-glossary>

Strukturen i kursplanen

Kursplanen består av sju huvudkapitel. Ett kapitel täcker en utbildningsenhet (EU). Titeln hos varje huvudkapitel innehåller den kognitiva nivån för kapitlet, vilket är den högsta nivån av underkapitel. Den angivna tiden indikerar den tid som kursen bör använda för att gå igenom kapitlet. Kursleverantörer kan fritt tilldela mer tid men bör bibehålla proportionerna mellan EUs. Viktiga termer i kapitlet som definieras i ordlistan listas i kapitlets början.

Exempel

EU 4 praxis för att utarbeta krav (L3)

Längd: 4 timmar 30 minuter

Termer: Kravkälla, systemgräns, systemkontext, kavelicitering, kravvalidering, intressent, Kano-modellen, konflikt, konfliktlösning

Detta exempel visar att kapitel 4 innehåller utbildningsmål på nivå L3 och 4 timmar 30 minuter är avsedda för undervisning av materialet i detta kapitel.

Varje kapitel kan innehålla underkapitel. De innehåller också den kognitiva nivån på deras innehåll.

Utbildningsmål (EO) räknas före den egentliga texten. Numreringen visar vilket underkapitel de tillhör. Till exempel beskrivs utbildningsmål EO 4.2.1 i underkapitel 4.2.

Ordning av ämnen i kursplanen

Ordningen på kapitel i denna kursplan utgör en logisk ordning av ämnen. Ämnen behöver dock inte undervisas i exakt denna ordning. Utbildningsleverantörer kan fritt lära ut materialet i vilken ordning som helst (inklusive integrera ämnen från olika EU) som de anser lämpliga i samband med deras utbildning och som passar deras didaktiska koncept.

Examinering

Kursplanen är grunden till examen för certifikat på CPRE Foundation Level.



En fråga under ett examenstillfälle kan täcka information från flera kapitel i kursplanen. Alla kapitel (EU 1 till EU 7) i kursplanen kan ingå vid ett och samma examenstillfälle.

Formatet av examineringen är flervalsfrågor.

Examineringar kan hållas direkt efter en kurs, men också oberoende av kurser (t ex i ett examineringscentrum). En lista över IREB-licensierade certifieringsorgan kan hittas på webbplatsen <https://www.ireb.org>.

Versionshistorik

Version	Datum	Kommentarer
3.0.0	1 Juli 2021	Stor uppdatering för att återspegla nuläget av samtida kravhantering som täcker plandrivna och agila tillvägagångssätt för att specificera och hantera krav.
3.0.1	2 Juli 2021	Korrigerings av stavfel och grammatik.
3.1.0	1 September 2022	Rättade stavfel och referenser i hela dokumentet för att förbättra läsbarheten. EU 1: <ul style="list-style-type: none">▪ Utbildningsmål är flyttat från EO 1.3.2 till EO 1.1.2.▪ EO 1.2.2 och 1.3.1 är uppdaterade EU 3 <ul style="list-style-type: none">▪ Utbildningsmål EO 3.1.3 är uppdaterade▪ EU 3.1.2: Omarbetade paragrafer på abstraktionsnivåer▪ EU 3.4: Alla modelltyper som inte tillämpas på Foundation Level har flyttats till ett nytt underkapitel 3.4.6▪ EU 3.6: Uppdaterad rubrik till "Kravdokument och dokumentationsstrukturer" EU 4 <ul style="list-style-type: none">▪ EU 4.1: Beskrivningen av intressenter och intressentroller definierades mer exakt.▪ EU 4.2: Introduktionen och motiveringen för design- och idégenererande tekniker har uppdaterats för att vara mer exakt.▪ EU 4.3: Uppdaterad rubrik till "Konfliktlösning angående krav" EU 6 <ul style="list-style-type: none">▪ EO 6.3.1 och EO 6.5.2 har modifierats lite
3.1.1	1 Januari 2024	Ny grafisk profil

Innehåll

Innehåll	7
1 Introduktion och översikt över kravhantering (L2)	10
1.1 Kravhantering: Vad (L1)	10
1.2 Kravhantering: Varför (L2)	11
1.3 Kravhantering: Var (L2)	11
1.4 Kravhantering: Hur (L1)	11
1.5 Kravingenjörens roll och uppgifter (L1)	11
1.6 Vad man ska lära sig om kravhantering (L1)	12
2 Grundläggande principer för kravhantering (L2)	13
2.1 Översikt över principer (L1)	13
2.2 Förklaring av principerna (L2)	13
3 Arbetsprodukter och dokumentationsmetoder (L3)	18
3.1 Arbetsprodukter inom kravhantering (L2)	19
3.1.1 Egenskaper för arbetsprodukter (L1)	19
3.1.2 Abstraktionsnivåer (L2)	20
3.1.3 Detaljeringsnivå (L2)	20
3.1.4 Aspekter att beakta i arbetsprodukter (L1)	20
3.1.5 Allmänna riktlinjer för dokumentation (L1)	21
3.1.6 Planera de arbetsprodukter som ska användas (L1)	21
3.2 Arbetsprodukter baserade på naturligt språk (L2)	22
3.3 Mallbaserade arbetsprodukter (L3)	23
3.4 Modellbaserade arbetsprodukter (L3)	23
3.4.1 Modellers roll inom kravhantering (L2)	23
3.4.2 Kontextmodellering (L2)	25
3.4.3 Struktur- och datamodelivering (L3)	25
3.4.4 Funktions- och flödesmodellering (L3)	25

3.4.5	Modellering av tillstånd och beteende [L2]	26
3.4.6	Ytterligare modelltyper i kravhantering [L1]	26
3.5	Ordlistor [L2]	26
3.6	Kravdokument och dokumentstruktur för krav [L2]	27
3.7	Prototyper inom kravhantering [L1]	28
3.8	Kvalitetskriterier för arbetsprodukter och krav [L1]	28
4	Praxis för utarbetande av krav [L3]	30
4.1	Kravkällor [L3]	30
4.2	Kravelicitering [L2]	32
4.3	Lösa konflikter angående krav [L2]	33
4.4	Kravvalidering [L2]	34
5	Process- och arbetsstruktur [L3]	35
5.1	Faktorer som påverkar [L2]	35
5.2	Olika aspekter av kravhanteringsprocessen [L2]	36
5.3	Konfigurera en kravhanteringsprocess [L3]	38
6	Praxis för kravförvaltning [L2]	40
6.1	Vad är kravförvaltning? [L1]	40
6.2	Livscykelhantering [L2]	40
6.3	Versionshantering [L2]	41
6.4	Konfigurationer och baslinjer [L1]	41
6.5	Attribut och vyer [L2]	41
6.6	Spårbarhet [L1]	42
6.7	Ändringshantering [L1]	42
6.8	Prioritering [L1]	42

7	Verktøysstød (L2)	44
7.1	Verktøy for kravhantering (L1)	44
7.2	Introduksjon av verktøy (L2)	45
	Referenser	46

1 Introduktion och översikt över kravhantering (L2)

Mål: Att förstå vad kravhantering handlar om och förstå värdet av kravhantering

Längd: 1 timme

Termer: Krav, kravspecifikation, kravhantering, intressenter, kravingenjör

Utbildningsmål

- EO 1.1.1 Komma ihåg den grundläggande terminologin (L1)
- EO 1.1.2 Förstå de olika typerna av krav (L2)
- EO 1.2.1 Förklara värdet av kravhantering (L2)
- EO 1.2.2.2 Uppräkna symptom på otillräcklig RE (L1)
- EO 1.3.1 Känna till var RE kan tillämpas och var krav uppstår (L1)
- EO 1.4.1 Känna till de viktigaste uppgifterna för kravhantering och att en kravhanteringsprocess måste anpassas för att utföra dessa uppgifter (L1)
- EO 1.5.1 Beskriva en kravingenjörs roll och uppgifter (L1)
- EO 1.6.1 Komma ihåg vad en kravingenjör behöver lära sig (L1)

1.1 Kravhantering: Vad (L1)

Människor och organisationer har önskemål och behov för att bygga nya saker eller vidareutveckla befintliga saker. Vi kallar sådana behov för *krav*.

Saker som ska byggas eller utvecklas kan vara:

- *Produkter* som tillhandahålls till kunder
- *Tjänster* som görs tillgängliga för kunderna
- *Andra leverabler* som apparater, procedurer eller verktyg som hjälper människor och organisationer att uppnå ett specifikt mål
- *Uppsättningar* eller *komponenter* av produkter, tjänster eller andra leverabler

Alla dessa saker kan betraktas som *system*. I denna kursplan använder vi termen *system* för att beteckna alla slags saker som *intressenter* har krav på. *Intressenter* är personer eller organisationer som påverkar kraven för ett system eller som påverkas av det systemet.

Målet med kravhantering är att specificera och förvalta krav för system så att de system som implementeras och levereras uppfyller intressenternas önskemål och behov.

Inom kravhantering skiljer vi mellan tre typer av krav [Glin2020]:

- *Funktionella krav* som omfattar ett resultat eller ett beteende som ska tillhandahållas av en systemfunktion. Detta inkluderar krav för data eller interaktion mellan ett system och dess miljö.
- *Kvalitetskrav* avser kvalitetsaspekter som inte täcks av funktionella krav, som prestanda, tillgänglighet, säkerhet eller tillförlitlighet.
- *Begränsningar* är krav som begränsar lösningsutrymmet utöver vad som är nödvändigt för att uppfylla de givna funktionella kraven och kvalitetskraven.

1.2 Kravhantering: Varför (L2)

Tillräcklig (adekvat) kravhantering tillför *värde* i utvecklingsprocessen för ett system:

- Minskar risken att fel system utvecklas
- Bättre förståelse av problemet
- Grund för att uppskatta utvecklingsinsatser och kostnader
- Förutsättning för att testa systemet

Typiska symtom på otillräcklig kravhantering är, saknade, oklara eller felaktiga krav. Detta beror särskilt på:

- Att man kastar sig rakt in i att bygga systemet
- Kommunikationsproblem mellan inblandade parter
- Antagandet att kraven är självklara
- Otillräcklig utbildning och kunskap inom kravhantering

1.3 Kravhantering: Var (L2)

Kravhantering kan tillämpas på krav för alla typer av system. Dock tillämpas kravhantering idag till övervägande del på system där programvara spelar en viktig roll. Sådana system består vanligtvis av programvarukomponenter, fysiska och organisatoriska element.

Kraven kan vara följande:

- *Systemkrav* – vad ett system ska göra
- *Intressentkrav* – vad intressenter vill ha från deras perspektiv
- *Användarkrav* – vad användare vill ha från sitt perspektiv
- *Domänkrav* – nödvändiga domänegenskaper
- *Affärskrav* – affärsmål, avsikter och behov i en organisation

1.4 Kravhantering: Hur (L1)

De viktigaste uppgifterna inom kravhantering är elicitering (4.2), dokumentation (3), validering (4.4) och förvaltning (6) av krav. Verktygsstöd (7) kan hjälpa till att utföra dessa uppgifter. Kravanalys och upplösande av kravkonflikter (4.3) anses vara en del av elicitering. För att kunna utföra kravhanteringsaktiviteter på rätt sätt, måste en lämplig kravhanteringsprocess anpassas utifrån ett brett spektrum av möjligheter (5).

1.5 Kravingenjörens roll och uppgifter (L1)

Kravingenjör är vanligtvis inte någon yrkestitel utan en *roll* för de som:

- Eliciterar, dokumenterar, validerar och/eller förvaltar krav som en del av sina arbetsuppgifter.
- Har djupgående kunskaper inom kravhantering.
- Kan överbrygga klyftan mellan problemet och potentiella lösningar.

I praktiken har affärsanalytiker, applikationsspecialister, produktägare, systemingenjörer och till och med utvecklare rollen som kravingenjör.

1.6 Vad man ska lära sig om kravhantering (L1)

Denna kursplan täcker de grundläggande färdigheter som en kravingenjör måste lära sig. Det omfattar de grundläggande principerna för kravhantering (2), hur man dokumenterar krav i olika former (3), hur man utarbetar krav med olika metoder (4), hur man definierar och arbetar med lämpliga kravhanteringsprocesser (5), hur man förvaltar befintliga krav (6) och hur man använder verktygsstöd (7).

2 Grundläggande principer för kravhantering

(L2)

Mål: Känna till och förstå principerna inom kravhantering
Längd: 1 timme 30 minuter
Termer: Kontext, krav, kravhantering (Requirements Engineering, RE), intressent, samförstånd, validering

Utbildningsmål

EO 2.1.1 Räkna upp principerna för kravhantering (L1)
EO 2.2.1 Komma ihåg de termer som är associerade med principerna (L1)
EO 2.2.2 Förklara principerna och varför de är viktiga (L2)

2.1 Översikt över principer (L1)

Kravhantering styrs av grundläggande principer som kan tillämpas på alla uppgifter, aktiviteter och praxis inom kravhantering. Följande nio principer utgör grunden för de metoder som presenteras i de följande kapitlen i denna kursplan.

1. Värdeorientering: Krav är ett medel för ett mål, inte ett mål i sig
2. Intressenter: Kravhantering handlar om att tillgodose intressenternas önskemål och behov
3. Samförstånd: Framgångsrik systemutveckling är omöjlig utan en gemensam grund
4. Systemkontext: System kan inte förstås utan att beakta dess kontext
5. Problem – Krav – Lösning: En oundvikligen sammanflätad trippel
6. Validering: Ovaliderade krav är värdelösa
7. Förändring: Kravförändringar är inte olycksfall i arbetet, utan normalfallet
8. Innovation: Mer av samma sak är inte tillräckligt
9. Systematiskt och disciplinerat arbete: Vi klarar oss inte utan kravhantering

2.2 Förklaring av principerna (L2)

Princip 1 - Värdeorientering: Krav är ett medel för ett mål, inte ett mål i sig

Värdet av ett krav motsvarar dess fördelar minus kostnaderna för elicitering, dokumentering, validering och kravförvaltning. Fördelarna med ett krav är den grad till vilken det bidrar till:

- Att bygga system som tillgodoser intressenternas önskemål och behov.
- Minska risken för felsymptom och kostsam omarbetning när systemet utvecklas.

Princip 2 - Intressenter: Kravhantering handlar om att tillgodose intressenternas önskemål och behov

Eftersom kravhantering handlar om att förstå intressenternas önskemål och behov, är god hantering av intressenter en central uppgift inom kravhantering. Varje intressent har en roll i sammanhanget för det system som ska byggas – till exempel användare, klient, kund, operatör eller förvaltare. En intressent kan också ha mer än en roll samtidigt. För intressentroller med för många individer eller när individer är okända kan fiktiva arketypiska beskrivningar som kallas *personas* definieras för att ersätta dessa. Det räcker inte att enbart ta hänsyn till slutanvändarnas eller kundernas krav. Genom att göra detta skulle vi kunna missa kritiska krav från andra intressenter. Användare som ger återkoppling till ett system som används bör också betraktas som intressenter.

Intressenter kan ha olika behov och synpunkter, vilket kan leda till motstridiga krav. Det är kravhanteringens uppgift att identifiera och lösa sådana konflikter.

Att engagera rätt personer för de relevanta intressentrollerna är avgörande för en framgångsrik kravhantering. Metoder för att identifiera, prioritera och arbeta med intressenter beskrivs i 4.

Princip 3 - Samförstånd: Framgångsrik systemutveckling är omöjlig utan en gemensam grund

Kravhantering skapar, främjar och säkerställer samförstånd mellan och bland de berörda parterna: intressenter, kravingenjörer och utvecklare. Det finns två former av samförstånd:

- *Explicit samförstånd* (uppnås genom dokumenterade och överenskomna krav)
- *Implicit samförstånd* (baserad på delad kunskap om behov, visioner, sammanhang, etc.)

Domänkunskap, tidigare framgångsrikt samarbete, delad kultur och värderingar och ömsesidigt förtroende möjliggör samförstånd, medan geografiskt avstånd, outsourcing eller stora team med hög personalomsättning är hinder.

Beprövade metoder för att uppnå samförstånd innefattar skapande av ordlistor (3.5), skapande av prototyper (3.7) eller användning av ett befintligt system som referenspunkt. Metoder för att bedöma samförstånd innefattar exempel på förväntade utfall, utforskande prototyper eller uppskattning av kostnaderna för att implementera ett krav. Den viktigaste metoden för att minska effekterna av missförstånd är användningen av en process med korta återkopplingsloopar (5).

Princip 4 - Systemkontext: System kan inte förstås utan att beakta dess kontext

System befinner sig i en *kontext*. Utan förståelse för den kontexten så är det omöjligt att korrekt specificera ett system. Inom kravhantering är systemkontexten den del av systemets miljö som är relevant för att förstå systemet och dess krav. *Systemgränsen* är gränsen mellan ett system och dess omgivande kontext. Inledningsvis är systemgränsen ofta otydlig och den kan till och med förändras över tid.

Att förtydliga systemgränsen och definiera de externa gränssnitten mellan systemet och de element i kontexten som de interagerar med är genuina kravhanteringsuppgifter. Samtidigt behöver systemets *omfattning* – det vill säga de saker som kan formas och designas genom systemutvecklingen – bestämmas. Vi behöver också ta hänsyn till den så kallade *kontextgränsen* som avgränsar den delen av systemets kontext som är relevant för kravhantering från resten av världen.

När man specificerar ett system är det inte tillräckligt att enbart ta hänsyn till kraven inom systemgränsen. Inom kravhantering måste man också överväga:

- Förändringar i kontexten som kan påverka systemkraven.
- Krav i verkliga världen som är relevanta för systemet (och hur man kopplar dem till systemkrav).
- Antaganden om kontexten som måste uppfyllas för att systemet ska fungera och uppfylla krav i verkliga världen.

Princip 5 - Problem - Krav - Lösning: En oundvikligen sammanflätad trippel

Ett *problem* uppstår när intressenterna inte är nöjda med nuvarande situation. *Krav* fångar vad intressenter behöver för att bli av med problemet eller förenkla det. Ett sociotekniskt system som uppfyller dessa krav utgör en *lösning*.

Problem, krav och lösningar förekommer inte nödvändigtvis i denna ordning. Lösningssidéer kan skapa användarbehov som måste bearbetas som krav och implementeras i en faktisk lösning. Detta är vanligtvis fallet med nya uppfinningar av innovativa lösningar.

- Problem, krav och lösningar är nära sammanflätade: de kan inte hanteras var för sig.
- Ändå strävar kravingenjörer efter att separera problem, krav och lösningar från varandra så långt det är möjligt när man tänker, kommunicerar och dokumenterar. Denna uppdelning gör kravhanteringsuppgifterna enklare att hantera.

Princip 6 - Validering: Ovaliderade krav är värdelösa

Så småningom måste vi validera att det levererade systemet uppfyller intressenternas önskemål och behov. För att ha kontroll över risken med otillfredsställda intressenter från början, måste validering av krav sätta igång redan under kravhantering. Vi måste kontrollera om:

- Enighet om kraven har uppnåtts bland intressenterna,
- Intressenternas önskemål och behov täcks på ett adekvat sätt av kraven,
- Antagandena om systemkontexten (se Princip 4) är rimliga.

Metoder för validering av krav diskuteras i 4.4.

Princip 7 - Förändring: Kravförändringar är inte olycksfall i arbetet, utan normalfallet

System och deras krav är föremål för *förändring*. Detta innebär att de ständigt förändras. Begäran om att ändra ett krav eller en uppsättning krav för ett system kan exempelvis orsakas av:

- Förändrade affärsprocesser
- Lansering av nya konkurrerande produkter eller tjänster
- Kunder med förändrade prioriteringar eller åsikter
- Tekniska förändringar
- Ändringar av lagar eller förordningar
- Återkoppling från systemanvändare gällande nya eller ändrade systemegenskaper

Dessutom kan kraven ändras på grund av återkoppling från intressenter vid validering av krav, på grund av upptäckta fel i tidigare eliciterade krav eller förändrade intressenters behov.

Som en följd av detta måste kravingenjörer sträva efter två till synes motsägelsefulla mål:

- Tillåta krav att ändras
- Hålla kraven stabila

Detaljer om hur man uppnår detta diskuteras i 6.7.

Princip 8 - Innovation: Mer av samma sak är inte tillräckligt

Genom att ge intressenterna exakt vad de vill ha förbises möjligheten att bygga system som tillfredsställer intressenternas behov bättre än vad de förväntar sig. God kravhantering strävar inte bara efter att tillgodose intressenterna, utan även att göra dem glada, förväntansfulla eller känna sig trygga. Ytterst, så är detta vad innovation handlar om.

Kravhantering formar innovativa system:

- I liten skala genom att sträva efter spännande nya funktioner och enklare användning.
- I stor skala genom att sträva efter banbrytande nya idéer.

På 4.2 diskuteras flera tekniker för att främja innovation inom förnybar energi.

Princip 9 - Systematiskt och disciplinerat arbete: Vi klarar oss inte utan kravhantering

Vi måste använda lämpliga processer och metoder för att systematiskt elicitera, dokumentera, validera och förvalta krav, oavsett vilken utvecklingsprocess som aktuellt används. Även när ett system utvecklas på ett slumpartat sätt så förbättras kvaliteten i det utvecklade systemet genom att utföra kravhanteringen på ett systematiskt och disciplinerat sätt.

Det finns inte en specifik process eller metod inom kravhantering som fungerar bra i varje given situation eller ens i de flesta situationer. Systematiskt och disciplinerat arbete innebär att kravingenjörer:

- Anpassar sina processer och metoder till det givna problemet, kontexten och miljön.
- Inte alltid använder samma process och uppsättning av metoder.
- Inte använder processer och metoder från ett föregående framgångsrikt kravhanteringsarbete utan reflektion.

Vid varje kravhanteringsinsats måste processer, metoder och arbetsprodukter väljas för att bäst passa den specifika situationen. Detaljerna beskrivs i 3, 4, 5 och 6.

3 Arbetsprodukter och dokumentationsmetoder

(L3)

- Mål: Förstå arbetsprodukternas grundläggande roll inom kravhantering och hur arbetsprodukter kan utformas
- Längd: 7 timmar
- Termer: Arbetsprodukt, arbetsprodukter baserade på naturligt språk, mallbaserade arbetsprodukter, modellbaserade arbetsprodukter, ordlista, kvalitetskriterier, kravspecifikation

Utbildningsmål

- EO 3.1.1 Känna till utmärkande egenskaper för arbetsprodukter inom kravhantering och kunna beskriva de arbetsprodukter som ofta används (L1)
- EO 3.1.2 Veta vad varje arbetsprodukt kan användas till och känna till livslängden för arbetsprodukter (L1)
- EO 3.1.3 Förklara de olika kategorierna och abstraktionsnivåerna för krav, inklusive hur man väljer lämpliga abstraktionsnivåer och detaljeringsnivåer (L2)
- EO 3.1.4 Känna till de aspekter som beaktas i arbetsprodukter och de inbördes förhållanden mellan dessa aspekter (L1)
- EO 3.1.5 Namnge de allmänna riktlinjerna för dokumentation (L1)
- EO 3.1.6 Beskriv varför det är värt att planera vilka arbetsprodukter som ska användas (L1)
- EO 3.2.1 Känna till arbetsprodukter baserade på naturligt språk och dess fördelar och nackdelar (L1)
- EO 3.2.2 Förklara reglerna för att skriva bra krav på naturligt språk (L2)
- EO 3.3.1 Känna till kategorierna för mallbaserade arbetsprodukter och dess fördelar och nackdelar (L1)
- EO 3.3.2 Specificera ett individuellt krav och en användarberättelse (user story) genom att använda en frasmall (L3)
- EO 3.3.3 Specificera ett användningsfall (use case) med hjälp av en formulärsmall (L3)
- EO 3.4.1 Förstå funktion, syfte och användning av modeller inom kravhantering (L2)
- EO 3.4.2 Förstå fördelarna och begränsningarna av modellering inom kravhantering (L2)
- EO 3.4.3 Känna till termerna modell, modelleringspråk, aktivitetsmodell, aktivitetsdiagram, klassmodell, klassdiagram, kontextmodell, kontextdiagram, domänmodell, målmodell, interaktionsmodell, processmodell, sekvensdiagram, tillståndsdigram, tillståndsmaskin, tillståndsmaskindigram, användningsfall (use case), användningsfallsdiagram (L1)
- EO 3.4.4 Förstå hur man väljer en lämplig modelleringsmetod för att specificera krav i en given situation (L2)
- EO 3.4.5 Förstå och tolka enkla modeller, skrivna i UML där det är tillämpligt, av följande typer: kontextmodeller, användningsfall (use case) och användningsfallsdiagram, domänmodeller, klassmodeller, aktivitetsmodeller, processmodeller och tillståndsdigram (L2)
- EO 3.4.6 Specificera en enkel modell av ett systems data eller objekten i en domän med hjälp av ett UML-klassdiagram (L3)
- EO 3.4.7 Specificera en enkel systemfunktion eller affärsprocess med ett UML-aktivitetsdiagram (L3)
- EO 3.5.1 Förklara syftet med ordlistor och hur man skapar dessa (L2)
- EO 3.6.1 Känna till ofta använda kravspecifikationsdokument (L1)
- EO 3.6.2 Förklara vilka dokumentstrukturer som tjänar vilket syfte och förklara kriterierna för strukturering (L2)

EO 3.7.1 Känna till olika typer av prototyper och vad de används för (L1)

EO 3.8.1 Känna till kvalitetskriterier för enskilda krav (L1)

EO 3.8.2 Känna till kvalitetskriterier för arbetsprodukter (L1)

3.1 Arbetsprodukter inom kravhantering (L2)

En arbetsprodukt är ett dokumenterat mellan- eller slutresultat som genereras i en arbetsprocess. Det finns en mängd olika arbetsprodukter inom kravhantering, som sträcker sig från, t ex grafiska skisser med kort livslängd, över användarberättelser i olika grader av mognad, till formellt släppta och kontrakterade kravspecifikationer på hundratals sidor.

3.1.1 Egenskaper för arbetsprodukter (L1)

Arbetsprodukter kännetecknas av dess syfte, framställning, storlek och livslängd. Följande arbetsprodukter förekommer ofta i praktiken för de angivna syftena. Observera att en arbetsprodukt kan innehålla andra arbetsprodukter.

- Arbetsprodukter för ett enskilt krav inkluderar individuella krav och användarberättelser (user stories).
- Arbetsprodukter för en uppsättning av krav omfattar användningsfall (use cases), grafiska modeller av vissa typer (3.4), uppgiftsbeskrivningar, externa gränssnittsbeskrivningar och epics.
- Arbetsprodukter som utgör omfattande dokument eller dokumentationsstrukturer är bland annat systemkravspecifikationer, produkt- och sprintbacklogs och story maps.
- Andra arbetsprodukter innefattar ordlistor, anteckningar, grafiska skisser och prototyper.

Arbetsprodukter kan *representeras* i olika former:

- Baserade på naturligt språk (3.2)
- Mallbaserade (3.3)
- Modellbaserade (3.4)
- Andra representationsformer, såsom ritningar eller prototyper (3.7)

De flesta arbetsprodukter *lagras* elektroniskt som filer, i databaser eller i kravhanteringsverktyg. Informella, tillfälliga arbetsprodukter kan också lagras på andra media – till exempel på papper eller post-it lappar på en Kanban-tavla.

Gällande livslängden för arbetsprodukter, så urskiljer vi tre kategorier:

- *Tillfälliga arbetsprodukter*: skapade för att stödja kommunikation och för att skapa samförstånd.
- *Föränderliga arbetsprodukter*: som växer fram över tid i flera iterationer; behöver en del metadata (6.5); process för ändringskontroll kan vara applicerbar.
- *Varaktiga arbetsprodukter*: som har fastställts i en baslinje eller en release; behöver en full uppsättning metadata (6.5); process för ändringskontroll måste följas (6.3, 6.4).

En tillfällig arbetsprodukt kan omvandlas till en föränderlig arbetsprodukt (genom att behålla den och lägga till metadata). På samma sätt, kan en tillfällig eller föränderlig arbetsprodukt bli varaktig genom att skapa en baslinje eller release för den.

3.1.2 Abstraktionsnivåer (L2)

Krav finns vanligtvis på många olika *abstraktionsnivåer* – från, exempelvis högnivå-krav för en ny affärsprocess till krav på en mycket detaljerad nivå, som beteende hos en specifik programvarukomponent när en viss händelse inträffar.

Valet av lämplig abstraktionsnivå beror på det ämne som ska specificeras och på syftet med specifikationen. Det är dock viktigt att inte blanda krav på olika abstraktionsnivåer. Inom små och medelstora arbetsprodukter bör kraven ligga på mer eller mindre samma abstraktionsnivå.

I stora arbetsprodukter, såsom en systemkravspecifikation, bör kraven på olika abstraktionsnivåer hållas åtskilda genom att strukturera specifikationen på lämpligt sätt (3.6). Ett krav på en hög abstraktionsnivå kan förfinas till flera detaljerade krav på lägre, mer konkreta nivåer.

När högnivå krav eller intressentkrav uttrycks i varaktiga arbetsprodukter – såsom specifikationer för affärskrav eller intressentkrav, eller i visionsdokument – föregår de specifikationen av systemkraven. I andra projekt kan affärskrav, intressentkrav och systemkrav samutvecklas.

3.1.3 Detaljeringsnivå (L2)

Den *detaljeringsnivå* som kraven bör specificeras på beror på flera faktorer, särskilt:

- Problemet och utvecklingskontexten
- Graden av samförstånd av problemet
- Graden av frihet som lämnas till designers och programmerare
- Möjligheten för snabb återkoppling från intressenter under design- och implementeringsarbetet
- Kostnaden kontra värdet av en detaljering av specifikationen
- Tvingande standarder och juridiska begränsningar

Ju högre detaljeringsnivå är i de specificerade kraven, desto lägre är risken för att så småningom få ett oväntat eller ospecificerat utfall. Dock ökar kostnaden för specificeringen med ökande detaljeringsnivå.

3.1.4 Aspekter att beakta i arbetsprodukter (L1)

Olika aspekter måste beaktas när du specificerar krav i arbetsprodukter.

1. Kraven klassificeras efter följande typer (1.1):
 - a) Funktionella krav
 - b) Kvalitetskrav
 - c) Begränsningar

2. Funktionella krav fokuserar på olika aspekter av funktionaliteten i ett system:
 - a) Struktur och data
 - b) Funktion och flöde
3. Tillstånd och beteende
 - a) Slutligen kan krav endast förstås i sin kontext eller sammanhang (Princip 4 i 2):
 - b) *Systemkontext, inklusive externa aktörer*
 - c) *Systemgräns och externa gränssnitt*

Det finns många relationer och beroenden mellan de nämnda aspekterna. Till exempel, en användares begäran (kontext) kan utlösa en tillståndsövergång (tillstånd och beteende) som initierar en åtgärd följt av en annan åtgärd (funktion och flöde) som kräver data (struktur och data) för att förse användaren med ett resultat (kontext) inom ett givet tidsintervall (kvalitet).

Vissa arbetsprodukter fokuserar på en specifik aspekt och abstraherar från de övriga aspekterna. Detta är särskilt fallet för kravmodeller (3.4). Andra arbetsprodukter, såsom en systemkravspecifikation, täcker alla dessa aspekter. När olika aspekter dokumenteras i separata arbetsprodukter eller i olika delar inom samma arbetsprodukt måste dessa arbetsprodukter eller delar hållas konsekventa med varandra.

3.1.5 Allmänna riktlinjer för dokumentation (L1)

Oberoende av vilka tekniker som används så gäller följande riktlinjer när arbetsprodukter skapas:

- Välj en typ av arbetsprodukt som passar det *avsedda syftet*.
- *Undvik redundans* genom att referera till innehåll istället för att upprepa samma innehåll igen.
- *Se till att det inte finns några inkonsekvenser* mellan arbetsprodukter, särskilt när de täcker olika aspekter.
- *Använd termer konsekvent*, så som de är definierade i ordlistan.
- *Strukturera* arbetsprodukter på lämpligt sätt.

3.1.6 Planera de arbetsprodukter som ska användas (L1)

Varje projekt och varje domän skiljer sig åt, så uppsättningen av använda arbetsprodukter måste definieras för varje specifikt projekt/domän. Därför behövs en överenskommelse om följande frågor:

- I vilka arbetsprodukter ska kraven dokumenteras och för vilket syfte?
- Vilka abstraktionsnivåer behöver beaktas?
- Till vilken detaljeringsnivå behöver kraven dokumenteras för varje abstraktionsnivå?
- Hur ska kraven representeras i arbetsprodukterna?

Arbetsprodukterna som ska användas bör definieras i ett tidigt skede i ett projekt. Detta har flera fördelar då det:

- Hjälper till att planera arbetsinsatsen och resurser.
- Se till att lämpliga notationer används.
- Se till att alla resultat dokumenteras i rätt arbetsprodukter.
- Förebygger behovet av att större ommöblering av information och "slutredigering".
- Hjälper till att undvika redundans, vilket resulterar i mindre arbete och bättre underhållbarhet.

3.2 Arbetsprodukter baserade på naturligt språk (L2)

Sedan starten av systematiskt arbete med kravhantering, har krav skrivna på naturligt språk varit ett grundläggande sätt att specificera krav på i praktiken.

Arbetsprodukter som är baserade på det naturliga språket har stora fördelar:

- Obegränsat naturligt språk är extremt uttrycksfullt och flexibelt.
- Nästan alla tänkbara krav i alla aspekter kan uttryckas på naturligt språk.
- Naturligt språk används i vardagen och undervisas i skolan, så ingen specifik utbildning krävs för att läsa och förstå texter på naturligt språk.

Dessa fördelar kommer på bekostnad av att texter skrivna på naturligt språk ofta kan tolkas på flera olika sätt, vilket utgör ett problem när du specificerar krav. Dessutom är det svårt och dyrt att upptäcka tvetydigheter, utelämnanden och motsägelser i sådana texter.

Att skriva bra krav baserade på naturligt språk det stöds av att:

- Skriva korta och välstrukturerade meningar.
- Definiera och använda en enhetlig terminologi på ett konsekvent sätt (3.5).
- Undvika vaga eller tvetydiga termer och fraser.
- Att känna till fallgroparna i tekniska texter angivna nedan.

När du skriver tekniska dokument på naturligt språk finns det några välkända fallgropar som bör undvikas eller användas med försiktighet [GoRu2003].

Undvik:

- Ofullständiga beskrivningar
- Ospecifika substantiv
- Ofullständiga villkor
- Ofullständiga jämförelser

Använd med försiktighet:

- Passiv form
- Universella kvantifierare (såsom "alla" eller "aldrig")
- Nominalisering (dvs. att skapa substantiv från ett verb, t.ex. "autentisering")

3.3 Mallbaserade arbetsprodukter (L3)

Mallbaserade arbetsprodukter används för att övervinna vissa av bristerna i arbetsprodukter baserade på naturligt språk genom att tillhandahålla fördefinierade strukturer för kraven.

- *Frasmallar* ger en fördefinierad syntaktisk struktur för en fras som uttrycker ett krav, särskilt ett individuellt krav eller ett användarberättelse (user story).
- *Formulärsmallar* ger en uppsättning av fördefinierade fält i ett formulär som ska fyllas, till exempel för att skriva ett användningsfall (use case) eller ett mätbart kvalitetskrav.
- *Dokumentmallar* ger en fördefinierad struktur för ett kravdokument.

Olika mallar har beskrivits i litteraturen. [ISO29148], [MWHN2009] och [Rupp2014] tillhandahåller frasmallar för individuella krav. [Cohn2004] definierar en brett använd frasmall för användarberättelser (user stories) och [Cock2001] beskriver formulärsmallar för användningsfall (use cases). [Laue2002] har föreslagit en mall för uppgiftsbeskrivningar. [ISO29148] och [RoRo2012] tillhandahåller dokumentmallar för hela specifikationer. Dessutom kan en kund kräva att kundspecifika mallar används i ett projekt.

Fördelar med mallbaserade arbetsprodukter:

- Ger en tydlig, återanvändbar struktur
- Hjälper till att fånga den mest relevanta informationen
- Gör att krav och kravspecifikationer ser enhetliga ut
- Förbättrar den totala kvaliteten på krav och kravspecifikationer

Nackdelar och fallgropar med mallbaserade arbetsprodukter:

- Människor fokuserar ofta på formellt färdigställande av mallen snarare än på innehållet.
- Aspekter som inte ingår i mallen är mer benägna att utelämnas.

3.4 Modellbaserade arbetsprodukter (L3)

Krav representerade på naturligt språk har begränsningar [Davi1993], särskilt när det gäller att få en överblick över en uppsättning krav och att förstå förhållanden mellan olika krav. Modellering av krav adresserar dessa begränsningar.

3.4.1 Modellers roll inom kravhantering (L2)

En *modell* är en abstrakt representation av en befintlig del av verkligheten eller en del av verkligheten som ska skapas. Begreppet verklighet inkluderar alla tänkbara uppsättningar av element, fenomen eller koncept, inklusive andra modeller. Med avseende på en modell, kallas den modellerade delen av verkligheten för *originalen*. Exempel på modeller utanför programvaruteknikens område är byggnadsinformationsmodeller (BIM) [ISO19650], som modellerar de delar som krävs för att planera, bygga och hantera byggnader och andra byggnadsdelar.

Inom kravhantering hjälper modeller till att förstå förhållanden och kopplingar mellan olika krav och ger en överblick för en uppsättning krav. Detta uppnås främst genom att fokusera på vissa aspekter – till exempel beteende – samtidigt som man abstraherar alla andra aspekter. Att få en överblick stöds också genom att använda en grafisk notation för en modell. Modeller kan dock också representeras på ett icke-grafiskt sätt, till exempel med tabeller.

Kravmodeller har fördelar jämfört med krav som representeras med naturligt språk:

- Förhållanden och kopplingar mellan krav är lättare att förstå med grafiska modeller än när de specificeras med naturligt språk.
- Att fokusera på en enda aspekt minskar den kognitiva belastning som krävs för att förstå de modellerade kraven.
- Kravmodellering har en begränsad syntax vilket minskar eventuella otvetydigheter och utelämnanden.

Modeller har också begränsningar:

- Att underhålla modeller som fokuserar på olika aspekter så att de är överensstämmande med varandra är utmanande.
- Information från olika modeller måste integreras för kausal förståelse.
- Modeller fokuserar främst på funktionella krav; de flesta kvalitetskrav och begränsningar kan inte uttryckas genom modeller med rimlig ansträngning.
- Den begränsade syntaxen för ett grafiskt modelleringsspråk innebär att all relevant information inte kan uttryckas i en modell.

Därför kombineras ofta kravmodeller och krav uttryckta med naturligt språk.

Inom kravhantering kan modeller användas för:

- *Specificering* av (främst funktionella) krav, delvis eller helt, som ett sätt att ersätta krav uttryckta i naturligt språk.
- *Nerbrytning* av en komplex verklighet i väl definierade och kompletterande aspekter; varje aspekt representeras av en specifik modell.
- *Omskrivning* av krav uttryckta i naturligt språk för att förbättra dess begriplighet, särskilt med avseende på relationer mellan kraven.
- *Validering* av krav med målet att upptäcka om något utelämnats, tvetydigheter, och inkonsekvenser.

Modelleringsspråk används för att uttrycka modeller. Flera modelleringsspråk, till exempel UML [OMG2017] eller BPMN [OMG2013], har standardiserats. När krav anges i ett icke-standardiserat format krävs en förklaring av syntaxen och semantiken för det modelleringsspråk som används.

Det finns många typer av modeller som kan användas inom kravhantering. En kravingenjör måste förstå vilken modelltyp som är bäst lämpad att specificera krav i en given situation.

I en tidig fas börjar kravingenjören ofta med att modellera sammanhanget (3.4.2) eller målen för det tänkta systemet.

3.4.2 Kontextmodellering (L2)

Modeller som fokuserar på den kontextuella aspekten specificerar den strukturella kontexten av ett system i dess miljö och interaktionen mellan ett system och aktörerna i systemets kontext.

Kontextmodeller specificerar ett system och aktörerna i dess kontext som interagerar med systemet. En kontextmodell skissar också på gränssnitten mellan ett system och dess kontext (t.ex. i termer av vilken information som utbyts).

Kontextdiagram används som ett grafiskt modelleringspråk för att uttrycka kontextmodeller. Det finns ingen standardiserad notation för kontextdiagram. Kontextdiagram från strukturerad analys [DeMa1978] eller skräddarsydda boxar-och-linjediagram [Glin2019] kan användas för att uttrycka kontextmodeller.

På modelleringspråket UML [OMG2017] ger *användningsfallsdiagram* en möjlighet att modellera ett system och dess kontext i termer av systemets användningsfall och aktörerna i systemkontexten som interagerar med systemet genom dessa användningsfall.

Användningsfall modellerar den dynamiska interaktionen mellan en aktör i systemkontexten och ett system ur aktörens perspektiv. Användningsfall skrivs oftast med formulärsmallar på naturligt språk (3.3) eller med hjälp av UML-aktivitetsdiagram (3.4.4).

3.4.3 Struktur- och datamodellering (L3)

Modeller som fokuserar på struktur- och dataspekter specificerar krav för statiska strukturella egenskaper hos ett system eller en domän.

Statiska domänmodeller specificerar (affärs)mål och dess relationer inom en given domän. De kan uttryckas med UML-klassdiagram [OMG2017].

Klassmodeller specificerar främst klasserna i ett system och deras attribut och relationer. Klasser representerar konkreta och immateriella enheter i den verkliga världen som systemet måste känna till för att utföra sina uppgifter. UML-klassdiagram används vanligtvis som modelleringspråk för klassmodeller.

3.4.4 Funktions- och flödesmodellering (L3)

Modeller som fokuserar på funktions- och flödesaspekter specificerar krav för sekvensen av åtgärder som krävs för att producera de erforderliga resultaten från en given indata eller de åtgärder som krävs för att utföra en (affärs) process, inklusive kontroll av flödet och data mellan aktiviteterna och vem som ansvarar för vilken aktivitet.

Aktivitetsmodeller används för att specificera systemfunktioner. På modelleringspråket UML [OMG2017] används *aktivitetsdiagram* för att uttrycka aktivitetsmodeller. De tillhandahåller element för att modellera åtgärder och kontrollflödet mellan handlingar. Aktivitetsdiagram kan också uttrycka vem som är ansvarig för vilken handling. Avancerade modelleringselement (som inte täcks av CPRE Foundation) gör det möjligt att modellera dataflödet.

Processmodeller används för att beskriva affärsprocesser eller tekniska processer. De kan uttryckas med UML-aktivitetsdiagram eller med specifika processmodelleringspråk som BPMN [OMG2013]. På CPRE Foundation-nivån använder vi bara UML-aktivitetsdiagram för processmodellering.

3.4.5 Modellering av tillstånd och beteende (L2)

Modeller som fokuserar på tillstånd och beteende specificerar krav på ett beteende hos ett system eller en domänkomponent när det gäller tillståndsberoende reaktioner på händelser eller dynamiken i komponentinteraktionen.

Tillståndsmaskiner modellerar händelser som triggar övergången från ett tillstånd till ett annat och de åtgärder som måste utföras när en tillståndsövergång äger rum.

Tillståndsdigram [Hare1988] är tillståndsmaskiner med tillstånd som bryts ned hierarkiskt och/eller ortogonalt. Tillståndsmaskiner, inklusive tillståndsdigram, kan uttryckas i modelleringspråket UML [OMG2017] med *tillståndsmaskindigram* (även kallade *tillståndsdigram*).

3.4.6 Ytterligare modelltyper i kravhantering (L1)

På CPRE:s grundnivå begränsas förståelsen och tillämpningen av modeller till utvalda, viktiga modelltyper. Det finns ytterligare modelltyper som används inom kravhantering. På CPRE:s grundnivå räcker det med att känna till dem och vad de används till.

Målmodeller representerar en uppsättning mål, delmål och relationerna mellan dem. Målmodeller kan också innehålla uppgifter och resurser som behövs för att uppnå ett mål, aktörer som vill uppnå målet och hinder som försvårar uppnåendet av målet.

I SysML [OMG2019] kan *blockdefinitionsdiagram* anpassas för att uttrycka kontextdiagram genom att använda stereotypa block för systemet och aktörerna. Blockdefinitionsdiagram kan också användas för att modellera ett systems struktur i form av systemets konceptuella enheter och relationerna mellan dem.

Modeller för berättelser om domäner kan användas för att modellera funktion och flöde genom att specificera visuella berättelser om hur aktörer interagerar med enheter, artefakter och andra objekt i en domän, vanligtvis med hjälp av domänspecifika symboler [HoSch2020]. De används för att förstå applikationsdomänen inom vilket ett system kommer att köra.

Interaktionsmodeller modellerar de dynamiska interaktionerna mellan objekt eller aktörer. UML-*sekvensdiagram* är ett vanligt förekommande sätt att specificera interaktionen mellan objekt.

3.5 Ordlistor (L2)

I varje kravhanteringsarbete som involverar mer än en person finns det risk för brist på samförstånd gällande terminologin – det vill säga att olika människor tolkar samma termer

på olika sätt. För att förebygga detta problem registreras den gemensamma förståelsen av relevanta termer i en ordlista.

En *ordlista* är en central samling av definitioner för: kontextspecifika termer, vardagliga termer med en speciell betydelse i den givna kontexten, förkortningar och akronymer. Synonymer (olika termer som betyder samma sak) bör markeras som sådana. Homonymer (med samma benämning för olika saker) bör undvikas eller markeras som sådana.

Följande regler gäller för ordlistor:

- Ordlistan måste underhållas centralt.
- Ordlistan måste underhållas under hela systemutvecklingens gång.
- Definiera en person eller liten grupp som ansvarar för ordlistan.
- Använd en enhetlig stil och struktur för ordlistan.
- Engagera intressenterna och sträva efter överenskommelse om terminologin.
- Gör ordlistan tillgänglig för alla inblandade.
- Gör det obligatoriskt att använda ordlistan.
- Kontrollera arbetsprodukter för korrekt användning av ordlistan.

3.6 Kravdokument och dokumentstruktur för krav (L2)

Kravspecifikationsdokument (3.1.1) omfattar flera arbetsprodukter för kravhantering. Det är därför viktigt att organisera sådana dokument med en väldefinierad struktur för att skapa en konsekvent och förvaltningsbar samling av krav. Förutom krav kan ett kravdokument också innehålla ytterligare information och förklaringar – till exempel en ordlista, acceptansvillkor, projektinformation eller egenskaper för den tekniska implementeringen.

Kraven kan också organiseras i andra dokumentationsstrukturer än klassiska dokument.

Ofta förekommande dokument är:

- Intressenternas kravspecifikation
- Användarkravspecifikation (en delmängd av intressenternas kravspecifikation, som endast avser användarnas krav)
- Systemkravspecifikation
- Affärskravspecifikation
- Visionsdokument

Ofta använda alternativa dokumentationsstrukturer är:

- Produktbacklogg
- Sprintbacklogg
- Kartberättelser (story map)

Både valet av en dokumentationsstruktur och den interna organiseringen av den valda strukturen beror på:

- Vald utvecklingsprocess (5)
- Utvecklingstyp och domän
- Kontraktet (en kund kan föreskriva användning av en viss dokumentstruktur)

- Dokumentets storlek

Dokumentmallar kan hjälpa till att strukturera en kravspecifikation. Mallar finns tillgängliga i litteraturen [Vole2020], [RoRo2012] och i standarder [ISO29148]. Mallar kan också återanvändas från tidigare, liknande projekt eller begäras av en kund. En organisation kan också besluta att skapa en mall som en intern standard.

3.7 Prototyper inom kravhantering (L1)

Inom kravhantering används *prototyper* för att specificera krav genom exempel och för att validera krav. I synnerhet kan prototyper användas om de berörda intressenterna inte vill skriva och granska arbetsprodukter som är mallbaserade, modellbaserade eller skrivna på naturligt språk.

Utforskande prototyper [LiSZ1994] används för att skapa samförstånd, förtydliga krav eller validera krav med olika grad av exakthet. De kastas efter användning.

- *Skisser* av ett användargränssnitt är prototyper med låg exakthet byggda med enkla material eller skissverktyg som främst tjänar till att diskutera och validera designidéer och koncept för användargränssnitt.
- *Mock-ups* är prototyper med en någorlunda exakthet. När de specificerar digitala system används de verkliga skärmar och klickflöden, men utan verklig funktionalitet. De används främst för att specificera och validera användargränssnitt.
- *Ursprungsprototyper* är prototyper med hög exakthet som implementerar kritiska delar av ett system i den utsträckning att intressenter kan använda prototypen för att se om den delen av systemet fungerar och uppträder som förväntat.

Beroende på graden av exakthet kan utforskande prototyper vara en dyr arbetsprodukt, så en avvägning behöver göras mellan kostnaden och värdet som kan erhållas.

Evolutionära prototyper [LiSZ1994] är pilotsystem som utgör kärnan i ett system som ska utvecklas. Det slutliga systemet utvecklas genom att stegvis utvidga och förbättra pilotsystemet i flera iterationer.

3.8 Kvalitetskriterier för arbetsprodukter och krav (L1)

Ett krav måste uppfylla specifika kvalitetskriterier för att betraktas som ett bra krav. I modern kravhantering med värdeinriktat tillvägagångssätt (Princip 1 i 2) ska graden av uppfyllelse för ett kvalitetskriterium motsvara det värde som skapas av detta krav. Detta innebär att kraven inte behöver följa alla kvalitetskriterier helt – men ju högre värdet ett krav har, desto mer relevanta blir kvalitetskriterierna, för att minska risken för fel.

Lämplighet och begriplighet är de viktigaste kvalitetskriterierna för enskilda krav. Utan dessa är ett krav värdelöst eller till och med skadligt, oavsett om alla andra kriterier uppfylls.

Kvalitetskriterier för *enskilda krav*:

- Lämpligt (beskriver verkliga och överenskomna behov av intressenter)
- Nödvändigt

- Otvetydigt
- Fullständigt (fristående)
- Begripligt
- Verifierbart

Som beskrivs i 3.1.1 dokumenteras krav vanligtvis i olika arbetsprodukter som täcker enskilda eller multipla krav. Kvalitetskriterierna ovan ska användas för att skapa enkla krav av hög kvalitet i en arbetsprodukt. För arbetsprodukter som täcker mer än ett krav ska följande kvalitetskriterier beaktas.

Kvalitetskriterier för arbetsprodukter som innehåller *flera krav*:

- Enhetliga
- Icke-redundanta
- Kompletta (inga kända och relevanta krav saknas)
- Modifierbara
- Spårbara
- Överensstämmande

4 Praxis för utarbetande av krav (L3)

Mål: Förstå användningen av metoder för att identifiera kravkällor, för att elicitera krav, för att identifiera och lösa konflikter och för att validera krav

Längd: 4 timmar 30 minuter

Termer: Kravkälla, systemgräns, systemkontext, kravelicitering, kravvalidering, intressenter, Kano-modellen, konfliktlösning

Utbildningsmål

- EO 4.1.1 Bestäm systemets gränser för att sätta fokus på relevanta krav (L3)
- EO 4.1.2 Kom ihåg de relevanta källorna för systemet som ska tas fram (L1)
- EO 4.1.3 Identifiera intressenter och skriv en intressentlista (L3)
- EO 4.1.4 Förstå fördelarna med förvaltningen av intressenter (L2)
- EO 4.2.1 Förstå hur Kano-modellen kan hjälpa till att elicitera rätt krav (L2)
- EO 4.2.2 Förstå skillnaden mellan insamlingstekniker, och design- och idégenererande tekniker (L2)
- EO 4.2.3 Förstå hur man väljer en lämplig eliciteringsteknik för en given situation (L2)
- EO 4.3.1 Kom ihåg de olika typerna av konflikter (L1)
- EO 4.3.2 Förstå vilka aktiviteter som är nödvändiga för att lösa konflikter (L2)
- EO 4.3.3 Förstå hur man använder lämpliga konfliktlösningstekniker (L2)
- EO 4.4.1 Förstå varför kravdokument måste valideras (L2)
- EO 4.4.2 Kom ihåg de fyra viktiga aspekterna för kravvalidering (L1)
- EO 4.4.3 Förstå hur man använder lämpliga tekniker för kravvalidering (L2)

4.1 Kravkällor (L3)

Kravkvalitet och fullständighet beror i hög grad på de kravkällor som är inblandade. Att sakna en relevant källa leder till en ofullständig förståelse av kraven eller till ofullständiga krav. Identifiering av kravkällor är en iterativ och återkommande process som kräver ständig reflektion.

Ett samförstånd (Princip 3 i 2) av kontexten för det system som ska utvecklas är en förutsättning för att kunna identifiera vilka kravkällor som är relevanta. Området mellan systemgränsen och kontextgränsen kallas (system)kontext (Princip 4 i 2). (System)kontexten behövs för att förstå vilken sorts krav som ska utvecklas och därmed identifiera de ursprungliga kravkällorna.

Kravkällor klassificeras i tre typer:

- Intressenter
- Dokument
- System

Ett systems intressenter (se [Glin2020] för en definition; se även Princip 2 i 2) är den huvudsakliga kravkällan. Typiska roller för intressenter är [BiSp2003]:

- Användare (även kallade slutanvändare)
- Sponsorer
- Chefer

- Utvecklare
- Myndigheter
- Kunder

Dessutom bör personer eller organisationer som *påverkas* av ett system betraktas som (indirekta) intressenter.

Systematisk identifiering av intressenter bör ske i början av ett utvecklingsinitiativ och resultaten bör hanteras under hela utvecklingen som en kontinuerlig aktivitet. Detta inkluderar identifiering av både intressentroller och personer i dessa roller.

För alla system med ett användargränssnitt utgör systemets *slutanvändare* en intressentgrupp som är av särskilt intresse för kravingenjören. Slutanvändarna bör delas in i grupper (t.ex. efter liknande roller, uppgifter eller ansvarsområden).

När slutanvändarna kan identifieras individuellt bör representanter från varje grupp väljas ut. Annars kan personas definieras för att representera de relevanta slutanvändargrupperna [Coop2004].

Potentiella källor för att identifiera relevanta intressenter och intressentroller är:

- Checklistor över typiska intressentgrupper och roller
- Organisationsstrukturer
- Affärsprocessdokumentation
- Marknadsrapporter
- Initiala intressenter för att identifiera *ytterligare* intressenter

Intressenter bör dokumenteras i en aktuell intressentlista med (åtminstone) följande information:

- Namn
- Funktion (roll)
- Ytterligare person- och kontaktinformation
- Tids- och platsmässig tillgänglighet under projektets gång
- Relevans
- Område och expertkunskaper
- Mål och intressen beträffande projektet

Problem med intressenter kan uppstå om intressenters rättigheter och skyldigheter inte är tydliga eller om intressentens behov inte är tillräckligt tillgodosedda. Förvaltning av intressentrelationer [Bour2009] är ett effektivt sätt att motverka problem med intressenter.

I de flesta fall finns ytterligare källor tillgängliga. De måste också beaktas för ett framgångsrikt nytt system, eftersom de flesta intressenter inte talar om det uppenbara: deras "undermedvetna" krav (4.2).

Ytterligare kravkällor inkluderar:

- Befintliga och äldre system
- Processdokumentation
- Legala eller regulatoriska dokument

- Företags specifika regelverk
- (Marknadsförings)information om potentiella framtida användare

Ytterligare kravkällor kan finnas genom att titta på liknande situationer i helt andra domäner.

4.2 Kravelicitering (L2)

Inom elicitering är det kravingenjörernas uppgift att förstå intressenternas önskemål och behov samtidigt som man säkerställer att kraven har samlats in från alla relevanta kravkällor genom att använda lämpliga tekniker för att elicitera dem. En viktig del av kravelicitering är att omvandla implicita krav, önskemål och förväntningar till explicita krav.

För att elicitera krav är det avgörande att veta arten och vikten av varje krav. Dessa kan ändras från projekt till projekt och även över tid. Kano-modellen [KaeA1984] klassificerar kraven i tre relevanta kategorier:

- Omedvetna behov (synonymer: Delighters, krav som överträffar kundens förväntningar)
- Uttalade behov (synonymer: Satisfiers, medvetna krav på viktiga funktioner)
- Underförstådda behov (synonymer: Dissatisfiers, för kunden självklara baskrav)

Det finns många olika eliciteringstekniker för att få fram dessa kategorier av krav. Vi skiljer mellan:

- Insamlingstekniker
- Design- och idégenererande tekniker

Insamlingstekniker är etablerade tekniker för kravelicitering [BaCC2015] som hjälper till att elicitera uttalade behov och underförstådda behov genom att undersöka olika källor.

Fyra huvudkategorier kan urskiljas:

- Frågetekniker
- Samarbetstekniker
- Observationstekniker
- Artefaktbaserade tekniker

Design- och idégenererande tekniker är avsedda att stimulera kreativiteten under kravutredningen. De syftar till att skapa idéer för att lösa ett problem och utforska designidéer [Kuma2013]. Detta kan leda till nya eller innovativa krav som ofta är kopplade till omedvetna behov (delighters). Populära exempel på sådana tekniker är brainstorming [Osbo1979], analogier, prototyper (t.ex. mock-ups), scenarier och storyboards.

Ett bredare koncept relaterat till design och idégenerering är *designtänkande*. Det finns olika tillvägagångssätt, så som *d.school* [Sdsc2012] och *Designing for Growth* [LiOg2011], som erbjuder en stor uppsättning tekniker som kan användas för att elicitera krav.

Eliciteringstekniker ska kunna upptäcka alla typer av krav – såväl funktionella och kvalitetskrav (icke-funktionella krav) som begränsningar. I praktiken får kvalitetskrav och begränsningar ofta mindre uppmärksamhet.

För att elicitera *kvalitetskrav* bör en kvalitetsmodell såsom ISO 25010-standarden [ISO25010] användas som en checklista. Denna modell kan också vara till hjälp vid kvantifiering av krav.

Begränsningar kan identifieras genom att överväga möjliga avgränsningar av den möjliga lösningsrymden – till exempel tekniska, juridiska, organisatoriska, kulturella eller miljöaspekter.

Att välja rätt eliciteringstekniker är en kritisk nyckelkompetens som beror på många olika faktorer, såsom:

- Typ av system
- Livscykelmodell för programvaruutveckling
- Involverade personer
- Organisatorisk struktur

De bästa resultaten uppnås vanligtvis med en kombination av olika eliciteringstekniker. [CaDJ2014] representerar ett systematiskt tillvägagångssätt för val av tekniker.

4.3 Lösa konflikter angående krav (L2)

Eliciteringstekniker i sig själva garanterar inte att den resulterande uppsättningen av krav som helhet är konsekvent, komplett, överensstämmande, etc. (3.8). För den slutgiltiga uppsättningen krav, måste alla intressenter förstå och komma överens om alla krav som är relevanta för dem. Om vissa intressenter inte håller med, måste denna situation identifieras som en konflikt som bör lösas. Lämpliga konfliktlösningstekniker bör väljas baserat på typen av konflikt och kontextuell information. Detta kräver en fördjupad förståelse av den aktuella kravkonflikten och av de berörda intressenternas perspektiv.

Uppgifter för att identifiera och lösa konflikter är:

- Konfliktidentifiering
- Konfliktanalys
- Konfliktlösning
- Dokumentation av konfliktlösning (fattade beslut)

Det är användbart att skilja mellan olika konflikttyper [Moor2014]. Följande typer av konflikter behöver ofta beaktas av kravingenjören:

- Ämneskonflikt
- Datakonflikt
- Intressekonflikt
- Värdekonflikt
- Relationskonflikt
- Strukturell konflikt

För att lösa konflikter framgångsrikt kan vanligt förekommande tekniker användas:

- Överenskommelse
- Kompromiss
- Röstning
- Avslag
- Definition av varianter

Dessutom finns det flera ytterligare tekniker, till exempel:

- Beakta-alla-fakta
- Plus-minus-intresse
- Beslutsmatris

4.4 Kravvalidering (L2)

Validering av krav är ett viktigt steg mot ett framgångsrikt system (Princip 6 i2). Att säkerställa kvaliteten på kraven i förväg förebygger onödiga ansträngningar under utvecklingen. Validering av krav innebär att man kontrollerar kvaliteten på den aktuella arbetsprodukten och de enskilda kraven i den (se 3.8 för mer information).

Viktiga aspekter som ska beaktas vid validering av krav är:

- Engagering av rätt intressenter
- Särskiljning av identifiering av defekter, från korrigerig av dessa
- Validering från olika perspektiv
- Upprepad validering

Det finns flera tekniker för kravvalidering (t.ex. [GiGr1993], [OleA2018]). Dessa valideringstekniker klassificeras ofta enligt:

- *Granskningstekniker*, inklusive:
 - Genomgångar
 - Inspektioner
- *Utforskande tekniker*, till exempel:
 - Prototyputveckling
 - Alfatestning och betatestning
 - A/B testning [KoTh2017]
 - Bygga en minsta livskraftiga produkt (MVP, Minimum Viable Product)
- *Exempel/provutveckling*

Dessa tekniker skiljer sig åt i formalitet och ansträngning. Vilken teknik som ska väljas beror på faktorer som livscykelmodellen för programvaruutveckling, utvecklingsprocessens mognad, systemets komplexitet och risknivå, eventuella rättsliga eller regulatoriska krav och/eller behovet av spårbarhet och redovisningskedja.

5 Process- och arbetsstruktur (L3)

Mål: Förklara koncepten för kravhanteringsprocesser och tillämpa lämpliga processkonfigurationer

Längd: 1 timme 15 minuter

Termer: Process, kravhanteringsprocess

Utbildningsmål

EO 5.1.1 Känna till de viktiga faktorer som påverkar en kravhanteringsprocess (L1)

EO 5.1.2 Förstå hur och varför dessa faktorer påverkar (L2)

EO 5.2.1 Förstå de aspekter som ska beaktas vid konfigurering av en kravhanteringsprocess (L2)

EO 5.3.1 Känna till typiska konfigurationer av en kravhanteringsprocess (L1)

EO 5.3.2 Förstå stegen för att konfigurera en kravhanteringsprocess (L2)

EO 5.3.3 Välja och använda en lämplig kravhanteringsprocesskonfiguration för enkla system- och utvecklingsmiljöer (L3)

En process behövs för att forma och strukturera det kravhanteringsarbete som ska utföras i ett givet sammanhang. Eftersom det inte finns en kravhanteringsprocess som passar alla (1.4), måste en anpassad kravhanteringsprocess tas fram så att den passar den givna utvecklings- och systemkontexten.

Kravhanteringsprocessen formar informationsflödet och kommunikationsmodellen mellan olika deltagare (t.ex. kunder, användare, kravingenjörer, utvecklare, testare) och definierar också de arbetsprodukter som ska användas eller produceras. Således ger kravhanteringsprocessen ramarna för att elicitera, dokumentera, validera och förvalta krav.

5.1 Faktorer som påverkar (L2)

Många faktorer påverkar konfigurationen av en kravhanteringsprocess. De viktigaste faktorerna är:

- Övergripande processpassning: kravhanteringsprocessen måste passa den övergripande systemutvecklingsprocessen.
- Utvecklingsammanhang
- Intressenters förmåga och tillgänglighet
- Gemensam förståelse
- Komplexitet och kritikalitet hos det system som ska utvecklas
- Begränsningar
- Tillgänglig tid och budget
- Hur förändringsbenägna kraven är
- Erfarenheten hos kravingenjörerna

En analys av de påverkande faktorerna ger information om hur kravhanteringsprocessen bör konfigureras. De påverkande faktorerna begränsar också utrymmet för möjliga processkonfigurationer. Till exempel, när intressenterna enbart är tillgängliga i början av

projektet så kan inte en process som bygger på kontinuerlig feedback från intressenterna väljas.

5.2 Olika aspekter av kravhanteringsprocessen (L2)

Det finns tre avgörande aspekter som måste beaktas när man konfigurerar en kravhanteringsprocess [Glin2019].

Tidsaspekten: Linjär vs iterativ

I en linjär process specificeras kraven i förväg i en enda fas av processen. I en iterativ process specificeras kraven stegvis (inkrementellt) och börjar med allmänna mål och några initiala krav, för att därefter addera eller modifiera krav i varje iteration.

Kriterier för att välja en *linjär* kravhanteringsprocess:

- Utvecklingsprocessen för systemet är plandrivna och mestadels linjär.
- Intressenterna känner till sina krav och kan specificera dem i förväg.
- En omfattande kravspecifikation krävs som avtalsgrund för kontraktering (outsourcing) av designen och implementeringen av systemet.
- Tillsynsmyndigheter kräver en omfattande, formellt utgiven kravspecifikation i ett tidigt skede av utvecklingen.

Kriterier för att välja en *iterativ* kravhanteringsprocess:

- Utvecklingsprocessen för systemet är iterativ och agil.
- Många krav är inte kända i förväg men kommer att framträda och utvecklas under systemutvecklingens gång.
- Intressenter är tillgängliga så att korta återkopplingslingor kan upprättas för att motverka risken att utveckla fel system.
- Utvecklingstiden tillåter mer än bara en eller två iterationer.
- Förmågan att enkelt förändra kraven är viktig.

Aspekt utifrån syftet: Normativ kontra explorativ

I en normativ kravhanteringsprocess utgör kravspecifikationen ett kontrakt: alla krav är bindande och måste implementeras. I en explorativ kravhanteringsprocess är endast målen kända på förhand, medan de konkreta kraven måste utforskas.

Kriterier för att välja en *normativ* kravhanteringsprocess:

- Kunden kräver ett fast kontrakt för systemutvecklingen.
- Funktionalitet och omfattning har företräde framför kostnad och deadlines.
- Utvecklingen av det specificerade systemet kan upphandlas eller utkontrakteras (outsourcas).

Kriterier för att välja en *explorativ* kravhanteringsprocess:

- Intressenter har ursprungligen enbart en vag uppfattning om sina krav.
- Intressenter är starkt involverade och ger kontinuerlig återkoppling.
- Deadlines och kostnad har företräde framför funktionalitet och omfattning.

- Det är inte klart på förhand vilka krav som faktiskt ska implementeras och i vilken ordning de ska implementeras.

Konceptmål: Kundenspecifik kontra marknadsorienterad

I en kundspecifik kravhanteringsprocess beställs systemet av en kund och utvecklas av en leverantör. I en marknadsorienterad kravhanteringsprocess utvecklas systemet som en produkt eller tjänst för en marknad, inriktad på specifika användarsegment.

Kriterier för att välja en *kundspecifik* kravhanteringsprocess:

- Systemet kommer huvudsakligen att användas av den organisation som har beställt systemet och betalar för dess utveckling.
- De viktigaste intressenterna är främst associerade med kundens organisation.
- Enskilda personer kan identifieras för intressentroller.
- Kunden vill ha en kravspekifikation som kan fungera som ett kontrakt.

Kriterier för att välja en *marknadsorienterad* kravhanteringsprocess:

- Den utvecklande organisationen avser att sälja systemet på någon marknad som en produkt eller tjänst.
- Potentiella användare kan inte identifieras individuellt.
- Kravingenjörerna måste utforma kraven så att de överensstämmer med de tänkta behoven hos målgruppen.
- Produktägare, marknadsförare, digitala designers och systemarkitekter är de främsta intressenterna.

Tips och varningar

- Kriterierna ovan är *heuristiska* snarare än bestämda regler. Till exempel utkontrakteras systemutveckling företrädesvis med en normativ kravhanteringsprocess snarare än med en explorativ process, eftersom kontrakt mellan kunden och leverantören vanligtvis är baserade på en omfattande kravspekifikation. Dock är det också möjligt att förhandla om ett outsourcingkontrakt baserat på en explorativ kravhanteringsprocess.
- Linjära kravhanteringsprocesser fungerar bara om en genomtänkt förändringsprocess för krav finns på plats.
- Linjära kravhanteringsprocesser innebär långa återkopplingslingor: För att minska risken för att utveckla fel system måste kraven valideras intensivt.
- Kravhanteringsprocesser definieras ofta som en kombination av de *linjära* och *normativa* koncepten.
- Explorativa kravhanteringsprocesser kombineras vanligtvis med iterativa processer (och vice versa).
- I en marknadsorienterad process är återkoppling från användare det enda sättet att verifiera om produkten faktiskt kommer att tillgodose behoven hos det tilltänkta användarsegmentet.
- Det fungerar inte så bra att kombinera det marknadsorienterade konceptet med de linjära och normativa koncepten.

5.3 Konfigurera en kravhanteringsprocess (L3)

I ett konkret systemutvecklingssammanhang måste de personer som ansvarar för kravhanteringen konfigurera kravprocessen som ska tillämpas. Baserat på en analys av de påverkande faktorerna i 5.1, kan en lämplig kombination av processaspekterna beskrivna i 5.2 användas [Glin2019]. Nedan beskrivs tre typiska kombinationer.

Deltagande kravhanteringsprocess: Iterativ & utforskande & kundspecifik

Huvudscenario:	Leverantör och kund har ett tätt samarbete; intressenter är djupt engagerade både i kravhanterings- och utvecklingsprocesserna
Typiska arbetsprodukter:	Produktbacklogg med användarberättelser och/eller uppgiftsbeskrivningar, prototyper
Typiskt informationsflöde:	Kontinuerlig interaktion mellan intressenter, produktägare, kravingenjörer och utvecklare; kan innefatta återkoppling från användare

Avtalsenlig kravhanteringsprocess: Vanligtvis linjär (ibland iterativ) & normativ & kundspecifik

Huvudscenario:	Kravspecifikationen utgör avtalsgrunden för systemutvecklingen av personer som inte är inblandade i specifikationen och med liten interaktion mellan intressenterna efter kravfasen.
Typiska arbetsprodukter:	Klassiska systemkravspecifikationer bestående av krav baserade på det naturliga språket och modeller
Typiskt informationsflöde:	Primärt från intressenter till kravingenjörer

Produktorienterad kravhanteringsprocess: Iterativ & Explorativ & Marknadsorienterad

Huvudscenario:	En organisation specificerar och utvecklar programvara för att sälja eller distribuera den som en produkt eller tjänst
Typiska arbetsprodukter:	Produktbacklogg, prototyper
Typiskt informationsflöde:	Interaktion mellan produktägare, marknadsförare, kravingenjörer, digitala designers, utvecklare och (kanske) snabb återkoppling från kunder/användare

Observera att det kan finnas system- och utvecklingssammanhang där ingen av de ovan nämnda konfigurationerna passar. Till exempel kan regulatoriska begränsningar förelägga

att en process som överensstämmer med givna standarder så som ISO/IEC/IEEE 29148 måste användas [ISO29148].

När du konfigurerar en kravhanteringsprocess rekommenderar vi att du använder en femstegsprocedure:

1. Analysera de påverkande faktorerna (5.1)
2. Bedöm aspektkriterierna (5.2)
3. Konfigurera processen (5.3)
4. Bestäm arbetsprodukter (3)
5. Välj lämpliga metoder

6 Praxis för kravförvaltning (L2)

Mål: Förstå behovet och nyttan av kravförvaltning

Längd: 2 timmar

Termer: Kravförvaltning, förändringshantering, spårbarhet, kravattribut, kravlivscykel, prioritering

Utbildningsmål

- EO 6.1.1 Veta vad kravförvaltning handlar om och varför det behövs (L1)
- EO 6.2.1 Förklara varför arbetsprodukter inom kravhanteringsprocessen behöver en status/livscykelmodell (L2)
- EO 6.3.1 Förklara hur ett koncept för versionering av krav ser ut i en viss projektsituation (L2)
- EO 6.4.1 Veta hur kravkonfigurationer och baslinjer kan användas (L1)
- EO 6.5.1 Känna till syftet med kravattribut (L1)
- EO 6.5.2 Förklara hur en lämplig uppsättning attribut för krav ser ut i en given projektsituation (L2)
- EO 6.5.3 Förklara syftet med vyer och namnge de olika kravvyerna (L2)
- EO 6.6.1 Ange anledningar till kravspårbarhet (L1)
- EO 6.6.2 Sammanfatta skillnaderna mellan implicit och explicit spårbarhet (L1)
- EO 6.6.3 Veta hur explicit spårbarhet kan dokumenteras (L1)
- EO 6.7.1 Veta hur man hanterar förändringar inom linjära (planbaserade) och agila metoder (L1)
- EO 6.8.1 Veta varför man prioriterar och känna till meningsfulla bedömningskriterier (L1)
- EO 6.8.2 Ange stegen för kravprioritering (L1)
- EO 6.8.3 Ange olika kategorier av prioriteringstekniker (L1)

6.1 Vad är kravförvaltning? (L1)

Kravförvaltning är processen för att hantera befintliga krav som dokumenterats i olika arbetsprodukter. I synnerhet inkluderar detta lagring, förändring och spårbarhet av krav [Glin2020]. Kravförvaltning kan utföras på olika sätt och på olika nivåer beroende på vald utvecklingsprocess och sammanhang – se till exempel [Leff2011], [Rupp2014], [WiBe2013]. Oavsett omständigheterna är syftet med kravförvaltning att underhålla kraven på ett sådant sätt att alla roller i ett projekt kan arbeta effektivt och ändamålsenligt.

6.2 Livscykelhantering (L2)

Livscykelhantering avser processen för att hantera alla arbetsprodukter med avseende på dess livscykelstatus. Varje dokumenterat krav och varje arbetsprodukt har sin egen livscykel: det skapas, utvärderas och förfinas innan det granskas, omarbetas, konsolideras, avtalas och så vidare. För att möjliggöra identifiering av statusen för varje arbetsprodukt behövs en livscykelmodell som definierar varje tillåten livscykelstatus och statusövergången mellan dessa. Den faktiska statusen för en arbetsprodukt ska alltid vara tydlig och (vanligtvis) inkludera historik över dess övergångar.

6.3 Versionshantering (L2)

Versionshantering av krav avser processen att hantera alla arbetsprodukter under dess utveckling. Varje förändring i en arbetsprodukt bör återspeglas i en ny version.

Versionshantering gör det möjligt att spåra historiken för en arbetsprodukt till dess ursprung och återställa en arbetsprodukt till vilken tidigare version som helst. För detta ändamål kräver versionskontroll tre åtgärder:

- Ett versionsnummer för att unikt identifiera en arbetsprodukts version.
- En historik över vad som ändrades.
- Ett koncept för lagring av arbetsprodukter.

Versionshantering måste övervägas för alla arbetsprodukter [WiBe2013]. Ett versionsnummer består vanligtvis av minst två delar: versionen och inkrementet.

6.4 Konfigurationer och baslinjer (L1)

En *kravkonfiguration* är en konsistent uppsättning av arbetsprodukter som innehåller krav. Varje konfiguration definieras för ett specifikt syfte och innehåller högst en version av varje arbetsprodukt [Glin2020]. Syftet med konfigurationer är, till exempel, att granska en uppsättning arbetsprodukter eller att underlätta en uppskattning av utvecklingsinsatsen.

En *baslinje* är en stabil, ändringskontrollerad konfiguration av arbetsprodukter, som används för planering av utgåvor (releaser) eller andra leveranser i ett projekt [Glin2020].

Konfigurationer har följande egenskaper:

1. Logiskt relaterade
2. Enhetliga
3. Unika
4. Oföränderliga
5. Grund för återställning

6.5 Attribut och vyer (L2)

Attribut krävs för att dokumentera viktiga metadata för en arbetsprodukt och används vanligtvis för att besvara ett antal viktiga frågor under projektets eller produktens livscykel.

Syftet med att använda attribut för att känneteckna krav är att göra det möjligt för teammedlemmar och andra intressenter att få tag på den information om kraven som de behöver när som helst under projektet.

Den relevant uppsättning attribut som definieras beror på informationsbehoven hos projektets olika intressenter. Befintliga standarder, till exempel [ISO29148], ger en översikt över de mest relevanta attributen.

Vyer är ett utdrag från den totala uppsättningen krav som endast visar den information som för närvarande är av intresse. Ur ett tekniskt perspektiv är en vy en kombination av filter-

och sorteringsinställningar som kan göras tillgängliga eller återanvändas för andra deltagare genom att spara den valda kombinationen.

Vi skiljer mellan tre typer av vyer:

- *Selektiva vyer*
- *Projektiva vyer*
- *Aggregerade vyer*

I de flesta fall är kravvyer en kombination av selektiva, projektiva och aggregerade vyer för att skapa rapporter.

6.6 Spårbarhet (L1)

Spårbarhet [GoFi1994] är förmågan att spåra ett krav *tillbaka till dess ursprung* (dvs. intressenter, dokument, motiveringar etc.) och *framåt till efterföljande arbetsprodukter* (t.ex. testfall), liksom *till andra krav* som detta krav är beroende av.

Spårbarhet är en förutsättning för kravförvaltning och krävs ofta uttryckligen av standarder, lagar och förordningar. Att genomföra spårbarhet innebär i princip att upprätthålla beroenden mellan olika arbetsprodukter (3.1) på olika abstraktionsnivåer (3.1.2), detaljnivåer (3.1.3) och till alla relevanta föregångare och efterföljare av analys-, efterlevnads- och informationsskäl.

Spårbarhet kan dokumenteras *implicit* genom att strukturera och standardisera arbetsprodukter, eller *explicit* genom att relatera arbetsprodukter till varandra baserat på deras unika identifierare i olika former [HuJD2011]. Vanliga representationsformer är hyperlänkar, referenser, matriser, tabeller eller grafer.

6.7 Ändringshantering (L1)

Kraven är inte statiska. Förändringar i krav sker av många olika anledningar och måste hanteras ordentligt (Princip 7 i 2) till exempel genom att skapa en formell *ändringsbegäran* eller genom att lägga till ett nytt objekt i en *produktbacklogg*.

Beslutsfattande, planering och kontroll av genomförandet av en förändring beror på utvecklings sättet och när i tiden förändringen inträffar.

I ett *linjärt* synsätt fattas beslutet om en förändring ofta av ett ändringsråd, till exempel ett Change Control Board (i projekt) eller ett Change Advisory Board (i drift). I ett mer *iterativt* tillvägagångssätt inkluderar produktägaren förändringen i produktbackloggen och prioriterar den nya uppgiften på lämpligt sätt.

6.8 Prioritering (L1)

Alla krav är inte lika viktiga [Davi2005]. Skattning och prioritering används för att bestämma de mest relevanta kraven för nästa produktrelease eller inkrement.

Skattning av kraven är grunden för prioritering, ofta bestämd genom att använda flera bedömningskriterier så som affärsvärde, angelägenhet, arbetsinsats, beroenden med flera.

Prioritering av ett krav beskriver vikten av ett enda krav jämfört med andra krav enligt vissa kriterier [Glin2020]. *Prioriteringen* som sådan genomförs baserat på ett eller flera kriterier; detta beror främst på den valda prioriteringstekniken.

Steg för prioritering:

- Definiera viktiga mål och begränsningar för prioritering
- Definiera önskade bedömningskriterier
- Definiera de intressenter som måste vara inblandade
- Definiera vilka krav som måste prioriteras
- Välj prioriteringsteknik
- Genomför prioritering

Prioriteringstekniker kan klassificeras som:

- *Ad-hoc* prioriteringstekniker
- *Analytiska* prioriteringstekniker

7 Verktøgsstød (L2)

Mål: Ge en översikt över syftet med kravhanteringsverktøg og aspekter for implementering

Längd: 30 minuter

Termer: Verktøg, kravhanteringsverktøg

Utbildningsmål

EO 7.1.1 Känna till de olika typerna av kravhanteringsverktøg (L1)

EO 7.2.1 Förklara vad man ska tänka på när man introducerar kravhanteringsverktøg (L2)

7.1 Verktøg for kravhantering (L1)

Kravhanteringsprocessen kan stødjas av verktøg som ger stød for dedikerade uppgifter og aktiviteter. Eftersom kravhanteringsprocess är ganska individuella (5), fokuserar befintliga kravhanteringsverktøg ofta bara på vissa aspekter inom kravhantering og stöder sällan alla aktiviteter. Innan ett verktøg väljs bör kravingenjör bestämma vilka uppgifter og aktiviteter under kravhanteringsprocessen som ska stødjas og hur. Vi skiljer mellan verktøg som ger stød for:

- Förvaltning av krav:
 - Definiera og spare kravattribut
 - Prioritera krav
 - Förvalta versioner og konfigurationer
 - Spårbarheten av krav
- Hantera kravförändringar
 - Styrning av kravhanteringsprocessen:
 - Mätning og rapportering av kravhanteringsprocessen
 - Mätning og rapportering av produktkvaliteten
- Hantera arbetsflødet i kravhanteringen
 - Dokumentation av kunskap om kraven:
 - Dela krav mellan olika intressenter
 - Skapa en gemensam forståelse for kraven
- Modellering av krav
- Samarbete inom kravhantering
- Testning/simulering av krav

Verktøg stødjer ofta en kombination av ovan nämnda funktioner. For att säkerställa bra stød for alla kravhanteringsuppgifter kan olika verktøg kombineras.

Ibland används andra typer av verktøg, till exempel kontors- eller ärendehanteringsverktøg, for att dokumentera eller hantera krav. Dessa verktøg har begränsningar og bör endast användas när kravhanteringsprocessen styr og kraven är styrande og då kraven är enhetliga og ganska stabila.

7.2 Introduktion av verktyg (L2)

Att välja ett kravhanteringsverktyg skiljer sig inte från att välja ett verktyg för något annat syfte. Målet, sammanhanget och kraven måste beskrivas innan valet kan lyckas [Fugg1993].

Ett lämpligt verktyg kan bara eftersökas när det korrekta kravhanteringsförfarandet och teknikerna har introducerats. Att introducera verktyg kräver tydlighet gällande ansvar och praxis för kravhantering. Följande aspekter är relevanta vid införandet av ett kravhanteringsverktyg:

- Ta hänsyn till alla livscykelkostnader utöver licenskostnader
- Överväg nödvändiga resurser
- Använd pilotprojekt för att undvika risker
- Utvärdera verktyget enligt definierade kriterier
- Instruera anställda i hur man använder verktyget

Referenser

- [AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.
- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.
- [BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.
- [Cock2001] A. Cockburn: Writing Effective Use Cases. Addison–Wesley, Boston 2001.
- [Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison–Wesley, Boston, 2004.
- [Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High–Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.
- [Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.
- [DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.
- [Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.
- [GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.
- [Glin2019] M. Glinz: Requirements Engineering I. Course Notes, University of Zurich, 2019. <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources>. Senast besökt juli 2020.
- [Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology. Version 2.0. <https://www.ireb.org/en/downloads/#cppe-glossary>. Senast besökt juli 2020.
- [GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.
- [GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.

- [GRL2020] Goal oriented Requirement Language. University of Toronto, Kanada
<https://www.cs.toronto.edu/km/GRL>. Senast besökt maj 2020..
- [Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.
- [HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, <http://leanpub.com/domainstorytelling>. Senast besökt juli 2020.
- [HuJD2011] E. Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3rd Ed, 2011.
- [ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.
- [ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.
- [ISO25010] ISO/IEC/IEEE 25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. International Organization for Standardization, Geneva, 2011.
- [Jack1995] M. A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison–Wesley, New York, 1995.
- [Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.
- [KaeA1984] N. Kano et al.: Attractive quality and must–be quality. Journal of the Japanese Society for Quality Control 1984, 14(2): 39–48. (in Japanese)
- [KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.
- [Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.
- [Laue2002] S. Lauesen: Software Requirements. Styles and Techniques. Addison–Wesley, Harlow, 2002.
- [Leff2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison–Wesley, Boston, 2011.
- [LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.
- [LiSZ1994] H. Lichter, M. Schneider–Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11): 825–832.

- [MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. *CibSE 2019*: 476–487.
- [Moor2014] C. W. Moore: *The Mediation Process – Practical Strategies for Resolving Conflicts*, 4th edition. John Wiley & Sons, Hoboken, 2014.
- [MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak: Easy Approach to Requirements Syntax (EARS). Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.
- [OleA2018] K. Olsen et al.: *Certified Tester, Foundation Level Syllabus – Version 2018*. International Software Testing Qualifications Board, 2018.
- [OMG2013] Object Management Group: *Business Process Model and Notation (BPMN)*, version 2.0.2. OMG document formal/2013–12–09 <http://www.omg.org/spec/BPMN>. Senast besökt juli 2020.
- [OMG2017] Object Management Group: *OMG Unified Modeling Language (OMG UML)*, version 2.5.1. OMG document formal/2017–12–05. <https://www.omg.org/spec/UML/About-UML/>. Senast besökt juli 2020.
- [OMG2019] Object Management Group: *OMG Systems Modeling Language (OMG SysML™)*, Version 1.6. OMG Document formal/19–11–01. <https://www.omg.org/spec/SysML/>. Senast besökt januari 2022.
- [Osbo1979] A. F. Osborn: *Applied Imagination*, 3rd revised edition. Charles Scribner's Sons, New York, 1979.
- [RoRo2012] S. Robertson and J. Robertson: *Mastering the Requirements Process*, 3rd edition. Addison–Wesley, Boston, 2012.
- [Rupp2014] C. Rupp: *Requirements–Engineering und Management*, 6. Auflage. Hanser, München, 2014. (in German).
- [Sdsc2012] Stanford d.school: *An Introduction to Design Thinking*. Hasso Plattner Institute of Design, Stanford, 2012. <https://dschool-old.stanford.edu/groups/designresources/wiki/36873>. Senast besökt juli 2020.
- [vLam2009] Axel van Lamsweerde: *Requirements Engineering: Från systemmål till UML–modeller till programvaruspecifikationer*. Chichester: John Wiley & Sons, 2009.
- [Vole2020] Volere: *Requirements Resources*. <https://www.volere.org>. Senast besökt juli 2020.
- [WiBe2013] K. Wiegers and J. Beatty: *Software Requirements*, 3rd edition. Microsoft Press, Redmond, 2013.