

Ordlista Engelsk/Svensk

Version 1.3

**Baserat på “Standard glossary of terms used in
Requirements Engineering ver 1.3”**

Upphovsrätt

Dokument får kopieras i delar eller som helhet för icke-kommersiell användning utan SQEBs medgivande, så länge som referens till källan uppges. Användning av dokumentet i kommersiellt eller utbildningssyfte får endast ske efter skriftligt medgivande av SQEB.

Den senaste versionen av SQEB Ordlista finns publicerad på <http://www.sqeb.se/>.

INNEHÅLL

1. Ändringshistorik	3
2. Medverkande	3
3. Syftet med detta dokument	3
4. Omfattning	3
5. Disposition	3
6. Normgivande referenser	4
7. Varumärken	4
8. Definitioner	5
<i>A</i>	5
<i>B</i>	6
<i>C</i>	8
<i>D</i>	10
<i>E</i>	11
<i>F</i>	12
<i>G</i>	13
<i>H</i>	13
<i>I</i>	14
<i>L</i>	15
<i>M</i>	15
<i>N</i>	17
<i>O</i>	17
<i>P</i>	17
<i>Q</i>	19
<i>R</i>	20
<i>S</i>	23
<i>T</i>	27
<i>U</i>	27
<i>V</i>	28
<i>W</i>	29
9. Bilaga A	30

1. Ändringshistorik

1.0	2011-09-15	Första utgåva
1.3	2015-03-31	Efter uppdatering av Glossary till ver 1.3.

2. Medverkande

Medverkande till detta dokument har varit

Beata Karpinska

Ingvar Nordström

Daniel Säther

Chris Hofstetter

Ninna Morin

3. Syftet med detta dokument

Syftet med detta dokument är att tillhandahålla en standardiserad ordlista av begrepp på engelska och svenska att användas av de personer som arbetar med kravhantering för att försäkra sig en gemensam förståelse av grundläggande begrepp och aktiviteter.

4. Omfattning

Dokumentet innehåller begrepp och definitioner, på engelska och svenska, som är relaterade till kravhantering, affärs och systemanalys, generell programvaruutveckling och relaterade områden.

5. Disposition

Ordlistan är uppbyggd i två kolumner, engelska och motsvarande svenska översättning. Den är sorterad efter den engelska definitionen. Vissa definitioner har föredragit före andra synonymer; i dessa fall finns synonymen också angiven

6. Normgivande referenser

Nedanstående referenser har använts som utgångsmaterial för denna ordlista. Dessa referenser var vid publicering av denna ordlista de senast gällande. Vid användning av denna ordlista bör hänsyn tas till om det har utkommit nyare versioner av nedanstående standarder.

- IEEE Standard 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology
- IEEE Standard 829-1998 IEEE Standard for Software Test Documentation
- IEEE Standard 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- IEEE Standard 1012-2004: IEEE Standard for Software Verification and Validation
- IEEE Standard 1059-1993: IEEE guide for software verification and validation plans
- IEEE Standard 1220-1998: IEEE Standard for Application and Management of Systems Engineering Process
- IEEE Standard 1233-1998 IEEE Guide for Developing System Requirements Specifications
- IEEE Standard 1362-1998 IEEE Guide for Information Technology-System Definition – Concept of Operations (ConOps) Document
- ISO 9000:2005. Quality Management Systems – Fundamentals and Vocabulary.
- ISO/IEC 12207:1995. Information Technology – Software Lifecycle Processes.
- ISO/IEC 14598-1:1999. Information Technology – Software Product Evaluation - Part 1: General Overview.
- ISO 15504-9: 1998. Information Technology – Software Process Assessment – Part 9: Vocabulary
- ISO 31000: Risk Management - Principles and Guidelines on Implementation
- IEC 31010: Risk Management - Risk Assessment Techniques
- ISO/IEC 73: Risk Management – Vocabulary
- ISTQB Glossary of testing terms ver. 2.2

7. Varumärken

I detta dokument används följande registrerade varumärken:

- BABOK är registrerat varumärke av IIBA
- CMM och CMMI är registrerade varumärken av Carnegie Mellon University
- BPMN är registrerat varumärke av Business Process Management Initiative (BPMI), nyligen sammanslaget med Object Management Group
- RUP är registrerat varumärke av Rational Software Corporation
- TMMi är registrerat varumärke av TMMi Foundation
- UML är registrerat varumärke Object Management Group
- SysML är registrerat varumärke Object Management Group

8. Definitioner

A	
acceptance criteria: The exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity [IEEE 610].	acceptanskriterier: De sluttillstånd eller krav som en komponent eller ett system ska uppnå för att accepteras av en användare, kund eller annan auktoriserad person eller organisation.
accuracy: The capability of the software product to provide the right or agreed results or effects with the needed degree of precision [after ISO/IEC 25000].	exakthet: även precision, noggrannhet. Möjligheten, förmågan hos systemet (inkluderat programvara, data och maskinvara) att erbjuda rätt och exakt det resultat och den effekt som eftersträvas. [after ISO 9126]
activity diagram: A graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.	aktivitetsdiagram: En grafisk representation av arbetsflöden med stegvisa aktiviteter och åtgärder, innehållande stöd för val, iteration och samtidiga händelser
actor: A type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject [OMG].	aktör: En roll som samspelar med subjektet (t.ex. utbyte av signaler och data) men som är externt i förhållande till subjektet [OMG].
ad hoc review: See <i>Informal review</i> .	ad hoc-granskning: Se <i>informell granskning</i> .
adaptability: The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the product considered [after ISO/IEC 25000]. See also <i>Portability</i> .	anpassningsmöjlighet: Programvaruproduktens kapacitet att anpassas till olika omgivningar och miljöer utan att produkten behöver ändras nämnvärt. [after ISO 9126] Se även <i>flyttbarhet</i> .
agile manifesto: A statement on the values that underpin agile software development. The values are: - Individuals and interactions over processes and tools - Working software over comprehensive documentation - Customer collaboration over contract negotiation - Responding to change over following a plan.	[det] agila manifestet: En beskrivning av de gemensamma värden som all agil programvaruutveckling vilar på: - Individer och samspel framför metoder, processer och verktyg - Körbar programvara framför omfattande dokumentation - Kundsamarbete framför kontraktsförhandlingar - Anpassning till förändring framför följandet av en plan.
agile software development: A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.	agil programvaruutveckling: En grupp av programvaruutvecklingsmetoder baserade på iterativ och inkrementell utveckling, där krav och lösningar växer fram genom samarbete mellan självorganiserande tvärfunktionella team.
agreeing on requirements: see <i>Requirements acceptance</i> .	överenskommelse om krav: Även kravbeslut. Se <i>kravacceptans</i>

apprenticing: A process of learning from the customer about his job. The customer teaches the Requirement Engineer – like a master and a student.	kundlärande: En lärandeprocess om kundens arbete. Kunden/ slutanvändaren) undervisar kravingenjören medan kravingenjören själv utför slutanvändarens uppgifter – liknande lärare och elev.
artifact: One of outcomes produced during the development of software. Some artifacts (e.g., use cases, class diagrams, and other UML models, requirements and design documents) help describe the function, architecture, and design of software. Other artifacts are concerned with the process of development itself - such as project plans, business cases, and risk assessments.	artefakt: Ett av resultaten som producerats i programvaruutveckling. Vissa artefakter (t.ex. användningsfall, klassdiagram och andra UML-modeller, krav och utvecklingsdokument) stödjer beskrivandet av funktionen, arkitekturen och utvecklingen av programvaran. Andra artefakter har att göra med utvecklingsprocessen i sig, som projektplaner, verksamhetsfall och riskutvärderingar
assessment: Activity of determination of quantitative or qualitative value of a product, service, activity or process in regard to given quality or acceptance criteria.	utvärdering: Aktiviteten att bedöma det kvantitativa och kvalitativa värdet av en produkt, en service, en aktivitet eller process i förhållande till ett givet kvalitets- eller acceptanskriterium
attractiveness: The capability of the software product to be attractive to the user [after ISO/IEC 25000]. See also <i>Usability</i> .	tilltalande: En egenskap hos ett program eller system som innebär att programmet eller systemet är tilltalande för användaren. [efter ISO 9126] Se även <i>användbarhet</i> .
attribute: A characteristic of an object.	attribut: En egenskap för ett objekt
audit: An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and/or procedures based on objective criteria, including documents that specify [IEEE 1028]: (1) the form or content of the products to be produced (2) the process by which the products shall be produced (3) how compliance to standards or guidelines shall be measured.	revision: även audit. Oberoende granskning med syfte att kontrollera att bestämda aktiviteter har utförts på rätt sätt enligt givna instruktioner, standarder, specifikationer eller procedurer baserat på objektiva kriterier som specificerar [IEEE 1028]: (1) format och innehåll hos den producerade produkten (2) processen som styr produktionen av produkten (3) hur standarder och riktlinjer skall tillmötesgå.
availability: The degree to which a component or system is operational and accessible when required for use. Often expressed as a percentage [IEEE 610].	tillgänglighet: Den andel i tid som en komponent eller ett system är i drift eller är tillgänglig när den/det behövs. [IEEE 610]
B	
ba: see <i>Business Analysis, Business Analyst</i> .	BA: se också <i>verksamhetsanalytiker</i>
baseline: A specification or software product that has been formally reviewed or agreed upon, that thereafter serves as the basis for further development, and that can be changed only through a formal change control process [IEEE 610].	baseline: Sammanfattande begrepp för att publicera ingående granskade delar i ett system och delarnas versioner. Denna konfiguration fungerar sedan som grund för vidare utveckling och kan endast ändras efter en formell ändringsprocess. [Efter IEEE 610]

behavioral diagram: In UML a type of diagram that depicts behavioral features of a system or business process. This includes activity, state machine, and use case diagrams as well as the four interaction diagrams. See also <i>Interaction diagrams</i> .	beteendemässigadiagram: Ingår i UML. Visar betendeegenskaper hos ett system eller affärs/verksamhets-process. Inkluderar aktivitets-, tillståndsmaskins- och användningsfallsdiagram samt de fyra interaktionsdiagrammen. Se <i>interaktionsdiagram</i>
benefit: Value delivered to stakeholders [T Gilb].	nytta: Värde som levererats till intressenter [efter T Gilb].
BPMN: see <i>Business Process Modeling Notation</i> .	BPMN: se <i>Business Process Modeling Notation</i> ,
business analysis (BA): A discipline providing a set of activities, tools and methods aiming to establish business needs, problems and goals and determine proper solutions allowing to satisfy those needs and resolve given business problems. See also <i>System Analysis</i> .	verksamhetsanalys: Samlade uppgifter, kunskap, verktyg och tekniker som behövs för att identifiera organisatoriska behov och bestämma lösningar för organisatoriska verksamhetsproblem [BABOK]. Se också: <i>Systemanalys</i>
business Analyst: A person responsible for determining the business needs, expectations and goals of stakeholders, in order to determine solutions to business problems [after BABOK]. See also <i>System Analyst</i> .	verksamhetsanalytiker: En person som ansvarar för att identifiera verksamhetsbehov, hos kund och intressenter, för att bestämma lösningar till verksamhetsproblem [BABOK]. Se också: <i>Systemanalytiker</i> .
Business constraint: The limitations on the project's flexibility to implement the requested solution [BABOK].	verksamhetsbegränsningar: Begränsningarna rörande projektets flexibilitet att implementera den önskade lösningen [BABOK]
business domain: (1)The set of classes that represent objects in the business model being implemented. (2) In general – an area of the business being a subject of or impacting the planned solution.	affärsområde: Även verksamhetsområde. (1) en grupp av klasser som representerar objekt i den affärsmodell som skall föras in. (2) Allmänt - ett område hos verksamheten som är målet för eller påverkas av den planerade lösningen.
Business need: See <i>Need</i> .	verksamhetsbehov: se <i>Behov</i>
business process: A collection of activities designed to produce a specific output for a particular customer or market.	verksamhetsprocess: Även verksamhetsprocess. En uppsättning av aktiviteter för att producera ett specifikt utdata eller resultat för en särskild kund eller marknad
Business Process Modeling Notation (BPMN): A graphical notation that depicts the steps in a business process. BPMN depicts the end to end flow of a business process. The notation has been specifically designed to coordinate the sequence of processes and the messages that flow between different process participants in a related set of activities [BPMN.ORG].	BPMN: Business Process Modeling Notation. En grafisk representation som skildrar stegen i en affärs eller verksamhetsprocess. BPMN visar processflödet från början till slut. Notationen är speciellt utvecklad för att koordinera flödessekvensen och meddelanden som utväxlas mellan olika processdeltagare i relaterade aktiviteter [efter BPMN. ORG].

C	
Capability Maturity Model (CMM): A five level staged framework that describes the key elements of an effective software process. The Capability Maturity Model covers best practices for planning, engineering and managing software development and maintenance [CMM]. See also: <i>Capability Maturity Model Integration (CMMI)</i> .	CMM (Capability Maturity Model): även processmognadsmodell för egenskaper. En modell för att utvärdera och certifiera mognadsgraden hos programvaruorganisationer.
Capability Maturity Model Integration (CMMI): A framework that describes the key elements of an effective product development and maintenance process. The Capability Maturity Model Integration covers best-practices for planning, engineering and managing product development and maintenance. CMMI is the designated successor of the CMM [CMMI]. See also: <i>Capability Maturity Model (CMM)</i> .	CMMI (Capability Maturity Model Integration): även integrerad processmognadsmodell för egenskaper. Ett ramverk som beskriver nyckelelement för att bedriva en effektiv produktutveckling och underhållsprocess. CMMI täcker metoder för planering, design och hantering av produktutveckling. CMMI är efterföljaren till CMM.
change control board (CCB): See <i>Configuration Control Board</i> .	ändringsråd: Se <i>CCB (Configuration Control Board)</i> .
change management: (1) A structured approach to transitioning individuals, teams, and organizations from a current state to a desired future state. (2) Controlled way to effect a change, or a proposed change, to a product or service.	ändringshantering: (1) Ett strukturerat tillvägagångssätt för att överföra individer, team och organisationer från ett tillstånd till ett önskvärt framtida tillstånd. (2) På ett kontrollerat sätt genomföra [föreslagna] förändringar i en produkt eller en tjänst. Se även konfigurationshantering.
change request: An official document requesting modification of existing features, requirements or functions or new ones.	ändringsbegäran: Dokument som beskriver ett önskemål att förändra existerande eller nya funktioner eller krav i ett system, inklusive påverkansanalys kring ändringen. Kallas även CR eller change request.
changeability: The capability of the software product to enable specified modifications to be implemented [after ISO/IEC 25000]. See also <i>Maintainability</i> .	flexibilitet: En egenskap hos programvara att tillåta enkel implementering av specificerade ändringar. [efter ISO 9126] Se även <i>underhållbarhet</i> .
class: A class describes a set of objects that share the same specifications of features, constraints, and semantics. Class is a kind of classifier whose features are attributes and operations.	klass: Definierar attributen och metoderna för en mängd objekt. Alla objekt av klassen (instanser av klassen) delar samma beteende, och har samma mängd attribut (varje objekt har sin egen uppsättning).
class diagram: A type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.	klassdiagram: Ingår i UML. En samling statiska modellelement, t.ex. klasser och typer, och deras innehåll och inbördes relationer.
client: see <i>Customer</i> .	se <i>kund</i>
commitment: The degree of obligation of meeting the requirement.	åtagande: Graden av förpliktelse att uppfylla kravet.
completeness of a requirement: The degree to which a requirement contains all necessary information.	komplett krav: Nivån som indikerar hur väl ett krav innehåller all nödvändig information

complexity: The degree to which a component or system has a design and/or internal structure that is difficult to understand, maintain and verify.	komplexitet: Svårighetsgraden att förstå, underhålla eller testa ett system eller en komponents design eller implementation.
compliance: The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions [after ISO/IEC 25000].	typgodkännande: Kapacitet hos programvaran att ansluta till standarder, gängse normer eller legala bestämmelser och liknande föreskrifter. [efter ISO/IEC 25000]
communication diagram: In UML a diagram that shows instances of classes, their interrelationships, and the message flow between them. For details refer to UML specification [OMG].	kommunikationsdiagram: Ingår i UML. En typ av diagram som beskriver programflödet mellan olika objekt.
component: A minimal software item that e.g. can be tested in isolation.	komponent: En programvarudel som kan testas isolerat från övriga komponenter och isolerat från systemet.
component diagram: In UML a diagram that depicts the components that compose an application, system, or enterprise. For details refer to UML specification [OMG].	komponentdiagram: Ingår i UML. Beskriver de komponenter som tillsammans utgör en applikation, ett system eller en del av ett system.
composite structure diagram: In UML a diagram that depicts the internal structure of a classifier (such as a class, component, or use case), including the interaction points of the classifier to other parts of the system. For details refer to UML specification [OMG].	sammansättnings-/strukturdiagram: Ingår i UML. Beskriver den interna strukturen av t.ex. en klass, en komponent eller användningsfall. Inkluderar samarbetspunkter med andra delar av systemet. För detaljer se UML specification [OMG].
conceptual model: A model that represents concepts (entities) and relationships between them. The aim of conceptual modeling is clarifying and expressing the meaning of terms and concepts used by domain experts to address the business problem, and establishing the correct relationships between different concepts.	konceptuell modell: En modell som beskriver begrepp (entiteter) och relationer mellan dessa. Målet med konceptuell modellering är förklara betydelsen av termer och begrepp som används av domänexperter för att beskriva affärsproblem (-behov) och etablera rätt relationer mellan olika begrepp.
configuration control board (CCB): A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes [IEEE 610].	CCB (Configuration Control Board): även ändringsråd. En grupp människor som är ansvariga för att utvärdera och godkänna eller förkasta föreslagna ändringar av konfigurationsobjekt och för att säkra införandet av godkända ändringar. [IEEE 610]
consistency: The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a component or system [IEEE 610].	överensstämmelse: Grad av likformighet, standardisering och avsaknad av motsägelser bland dokument eller delar av en komponent eller ett system. [IEEE 610]
constraint: A statement of restriction that modifies a requirement or set of requirements by limiting the range of acceptable solutions. See also <i>Business Constraints</i> , <i>Technical Constraints</i>	restriktion: även begränsning. Information som förändrar ett krav, eller grupp av krav, genom att begränsa omfattningen av acceptabla lösningar. Se även <i>verksamhetsbegränsning</i> , <i>teknisk begränsning</i>
context: System view from any useful perspective [TGilb].	sammanhang: Även kontext. Systemvy från något användbart perspektiv [TGilb].

context diagram: A diagram that represents the actors outside a system that could interact with that system.	kontextdiagram: Ett diagram som visar aktörer utanför ett system som kan samarbeta med detta system
contractor: see <i>Supplier</i> .	se <i>leverantör</i> .
coverage: The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite.	täckningsgrad: Den nivå, uttryckt i procent, till vilket ett specificerat objekt har blivit exekverat av en testsvit.
criticality of requirements: Evaluation of the risk of a requirement by evaluating the damage in case of non-fulfillment of a requirement.	(krav)kritikalitet: Även (krav)prioritet, (krav)konsekvens). Riskutvärdering av krav genom att bedöma den skada som skulle kunna uppstå om kravet inte uppfylls
customer: Current or potential buyer or user of the products or service of an individual or organization, called the supplier, seller, or vendor.	kund: Aktuella eller potentiella köpare eller användare av de produkter eller tjänster av en individ eller organisation, kallad leverantören, säljaren eller tillverkaren.
Customer Requirement: The result of eliciting, consolidating, and resolving conflicts among the needs, expectations, constraints, and interfaces of the product's relevant stakeholders in a way that is acceptable to the customer. [after CMMI]	kundkrav: Resultatet av insamling, konsolidering och lösande av konflikter hos de krav, förväntningar, restriktioner och gränssnitt som produktens intressenter har på ett sätt som är godtagbart för kunden. [efter CMMI]
D	
data flow diagram: A graphical representation of the sequence and possible changes of the state of data objects, where the state of an object is any of: creation, usage, or destruction.	dataflödesdiagram: En grafisk representation av sekvensflödet och möjliga tillståndändringar hos ett dataobjekt där objektet har tillstånden creation, usage eller destruction
decision table: A table showing combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects), which can be used to design test cases..	beslutstabell (tidigare kodvillkorstabell): En tabell som visar invärden och/eller stimuli med motsvarande utvärden och/eller funktion (verkan). Används för att ta fram testfall.
defect: A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system. [ISTQB]	defekt: Se <i>fel</i> .
defect management: The process of recognizing, investigating, taking action and disposing of defects. It involves recording defects, classifying them and identifying the impact [IEEE 1044].	felhantering: Processen att identifiera, utreda och vidta åtgärder samt avföra fel. Den omfattar registrering och klassificera av fel samt fastställande av deras påverkan. [Efter IEEE 1044]
defect management tool: A tool that facilitates the recording and status tracking of defects and changes. They often have workflow-oriented facilities to track and control the allocation, correction and re-testing of defects and provide reporting facilities.	felhanteringsverktyg: Ett verktyg som underlättar registrering och tillståndsspårning av fel. I felhanteringsverktyg finns ofta stöd för att följa ett arbetsflöde och för styrning, tilldelning, åtgärd och omtestning av händelser eller fel, och det tillhandahåller hjälp för rapportering.

deliverable: Any (work) product that must be delivered to someone other than the (work) product's author.	leverabel: En produkt som formellt skall överlämnas till en utpekad mottagare.
Delphi method: A structured communication technique used to conduct interactive forecasting. It involves a panel of experts [Linstone75].	delphi-metoden: En strukturerad kommunikationsteknik som används för att genomföra interaktiva prognoser. Den involverar en panel av experter [Linstone75].
dependency: A dependency is a reliance of some kind, of one set of components on another set of components, or one set of requirements or other artifacts on another set [TGilb].	beroende: Ett beroende av något slag mellan en uppsättning komponenter och en annan uppsättning av komponenter eller en uppsättning krav eller andra artefakter på en annan uppsättning [TGilb].
deployment diagram: In UML a diagram that shows the execution architecture of systems. For details refer to UML specification [OMG].	driftsättningsdiagram: Ingår i UML. Visar den exekverande arkitekturen i ett system
E	
efficiency: The capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions [after ISO/IEC 25000].	effektivitet: Programvaruproduktens kapacitet att tillhandahålla tillräckliga prestanda i relation till använda resurser och under definierade villkor. [efter ISO/IEC 25000].
elicitation: The act of obtaining information from other people. In the context of Requirements Engineering, elicitation is the process of gathering requirements from stakeholders.	insamling: Även elicitering, identifiering. Processen att fånga in och identifiera behoven och restriktionerna hos olika intressenter för ett programvarusystem. I kravhantering betyder detta att samla in krav från intressenter
end user: see <i>User</i> .	slutanvändare: se <i>användare</i>
entity: (1) An element or set of elements that has a distinct, separate existence, although it need not be a material existence. (2) An abstraction from the complexities of some domain.	enhet: Även entitet. (1) En del eller delmängd som har en tydlig, separat förekomst, (2) Ett avskiljande i komplexiteten i en domän.
entity-relationship diagram: see <i>Entity-relationship model</i> .	entitets-sambands-diagram: se <i>enhets-sambands-modell</i>
ERD: see <i>Entity-relationship diagram</i> .	ERD: se <i>enhets-sambands-diagram</i>
entity-relationship model: An abstract and conceptual representation of data. Entity-relationship model consists of a set of entities, characterized by attributes and linked by relationships.	enhets-sambands-modell: En avskiljande och konceptuell representation av data. Den består av en grupp enheter som kännetecknas av attribut och har beroendelänkar.
ERM: see <i>Entity-relationship model</i> .	ERM: se <i>enhets-sambands-modell</i> .
error: A human action that produces an incorrect result [IEEE 610].	misstag: (1) En mänsklig åtgärd som producerar ett felaktigt resultat. [IEEE 610] (2) även felaktighet. Se <i>fel</i> .

estimate: A numeric judgment about a future, present or past level of a scalar system attribute. This includes all performance and cost attributes. Estimates are usually made where direct measurement is: impossible (future), or impractical (past), or uneconomic (current levels) [TGilb].	estimat: En numerär bedömning om en framtida, nuvarande eller tidigare nivå på ett skalärt systemattribut. Omfattar alla prestanda och kostnadsattribut. Används oftast när direkta mätningar inte är möjliga (framtid) eller opraktiska (tidigare) eller oekonomiska (nuvarande). [TGilb].
extreme programming: A software engineering methodology used within agile software development whereby core practices are programming in pairs, doing extensive code review, unit testing of all code, and simplicity and clarity in code. See also <i>Agile software development</i> .	extreme programming (XP): En programvaruutvecklingsmetodik, som används inom agil programvaruutveckling, i vilken de centrala aktiviteterna innefattar parprogrammering, omfattande kodgranskning, enhetstestning av all kod samt enkel och tydlig kodning. Se även <i>agil programvaruutveckling</i> .
F	
failure: Deviation of the component or system from its expected delivery, service or result [Fenton].	felsymptom: även felyttring. Avvikelse från programvarans eller systemets förväntade beteende eller resultat. [Efter Fenton]
failure mode: The physical or functional manifestation of a failure. For example, a system in failure mode may be characterized by slow operation, incorrect outputs, or complete termination of execution [IEEE 610].	feltillstånd (tidigare felsymptomsläge): Den fysiska eller funktionella yttringen av ett felsymptom. Ett system i felsymptomsläge kan t.ex. ge tecken på långsam funktion, felaktiga utvärden eller exekveringsavbrott. [IEEE 610]
Failure Mode and Effect Analysis (FMEA): A systematic approach to risk identification and analysis of identifying possible modes of failure and attempting to prevent their occurrence.	FMEA (Failure Mode and Effect Analysis): Ett systematiskt angreppssätt för att identifiera och analysera möjliga risker för felsymptom/felyttringar och för att förhindra att de uppstår.
fault: see <i>Defect</i> .	fel: En brist i en komponent eller ett system som kan få komponenten eller systemet att misslyckas med att utföra sin önskade funktionalitet, t.ex. en felaktig kodsats eller datadefinition. Ett fel kan, om det uppträder under exekvering, orsaka ett felsymptom/en felyttring hos komponenten eller systemet.
feature: An attribute of a component or system specified or implied by requirements documentation (for example reliability, usability or design constraints) [IEEE 1008].	feature: även finess. Ett kännetecken hos en komponent eller ett system som har specificerade eller underförstådda krav (t.ex. tillförlitlighet, användbarhet eller designrestriktioner). [Efter IEEE 1008]
FMEA: see <i>Failure Mode and Effect Analysis</i> .	FMEA: se <i>Failure Mode and Effect Analysis</i> .
formal review: A review characterized by documented procedures and requirements, e.g. inspection.	formell granskning: En granskning som innebär att man har dokumenterade procedurer och krav för granskning, t.ex. en inspektion.
FPA: See <i>Function Point Analysis</i> .	FPA: Se <i>analys av funktionspoäng</i>

function: A description of “what” a system does. A function has a corresponding implied purpose and is a fundamental part of a system description: a system consists of function attributes, performance attributes, resource (cost) attributes and design attributes. All attributes exist with respect to defined specified conditions. A function can often be decomposed into a hierarchical set of sub-functions [TGilb].	funktion: En beskrivning av “vad” ett system gör. En funktion har ett syfte och ingår som en grundläggande del i en systembeskrivning. En funktion är oftast nedbruten hierarkiskt i delfunktioner.
Function Point: A unit of measurement to express the amount of business functionality provided by an information system to a user.	funktionspoäng: En måtenhet för att uttrycka mängden av funktionalitet som ett system ger en användare
Function Point Analysis (FPA): Method aiming to measure the size of the functionality of an information system. The measurement is independent of the technology. This measurement may be used as a basis for the measurement of productivity, the estimation of the needed resources, and project control.	FPA (Function Point Analysis): även analys av funktionspoäng. En metod som används för att mäta mängden funktioner i ett informationssystem. Mätningen är oberoende av teknologi. Den här mätningen kan användas som grund för mätning av produktivitet, resursbehov och projektstyrning.
Functional requirement: A requirement that specifies a function that a component or system must perform [IEEE 610].	Funktionellt krav: Ett krav som specificerar en funktion som en komponent eller ett system måste utföra. [IEEE 610]
Functionality: The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions [after ISO/IEC 25000].	funktionalitet: Programvaruproduktens kapacitet att tillhandahålla funktioner som motsvarar ställda och underförstådda behov när programvaran används under definierade förhållanden. [after ISO/IEC 25000].
G	
goal: A desired state or result of an undertaken. Goals should be measurable and defined in time so that the progress can be monitored.	mål: Ett önskat tillstånd eller resultat av ett åtagande. Mål skall vara mätbara och tidsbestämda så att framstegen kan övervakas
GUI: Graphical User Interface. A type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators.	GUI: även grafiskt användargränssnitt. En typ av användargränssnitt som gör det möjligt för användare att interagera med elektronisk utrustning genom grafiska ikoner och visuella indikatorer.
GUI prototyping: The activity of creating prototypes of GUI of software applications or other products. See also <i>GUI</i> .	grafiska gränssnittsprototyper: Aktiviteten att skapa GUI-prototyper för programvaruapplikationer eller andra produkter. Se även <i>GUI</i> .
H	
Horizontal prototype: A prototype that provides a broad view of an entire system or subsystem, focusing on user interaction more than low-level system functionality.	horisontella prototyper: En prototyp som tillhandahåller en bred bild över ett helt system eller delsystem och fokuserar på användarinteraktion mer än systemfunktionalitet på låg nivå.

horizontal traceability: Tracing between artifacts on the same level of abstraction (for example, tracing from a requirement to corresponding risk item).	horisontell spårbarhet: Spårning mellan artefakter på samma abstraktionsnivå (till exempel spårning från ett krav till motsvarande riskobjekt).
/	
impact: Estimated or actual numeric effect of a design idea on a requirement attribute under given conditions.	påverkan: Beräknat eller aktuell effekt av en design-idé till ett kravattribut under givna förutsättningar
incremental development model: A development lifecycle where a project is broken into a series of increments, each of which delivers a portion of the functionality in the overall project requirements. The requirements are prioritized and delivered in priority order in the appropriate increment. In some (but not all) versions of this lifecycle model, each subproject follows a 'mini V-model' with its own design, coding and testing phases.	modell för inkrementell utveckling: En utvecklingsmodell där projekt delas upp i en serie av inkrement, där varje inkrement levererar en del av funktionaliteten som ingår i projektkraven. Kraven prioriteras och levereras i prioriteringsordning i det tilldelade inkrementet. I vissa, men inte alla, versioner av projektets livscykel följer man en "mini-V-modell", med egna faser för design, kodning och testning.
informal review: A review not based on a formal (documented) procedure.	informell granskning: En granskning som inte baseras på en formell (dokumenterad) procedur.
interaction diagram: A subset of behavioral diagrams in UML which emphasize object interactions. This includes communication, interaction overview, sequence, and timing diagrams. See also <i>Behavioral diagram</i> .	interaktionsdiagram: En delmängd av beteendediagram som betonar samarbetet mellan objekt. Inkluderar kommunikations-, Interaktionsöversiktsdiagram-, sekvensdiagram och timingdiagram
interaction overview diagram: A variant of an activity diagram which overviews the control flow within a system or business process.	interaktionsöversiktsdiagram: Ingår i UML. En variant på ett aktivitetsdiagram som på ett övergripande sätt visar flödet i ett system eller affärs/verksamhets-process
inspection: A type of peer review that relies on visual examination of documents to detect defects, e.g. violations of development standards and non-conformance to higher level documentation. The most formal review technique and therefore always based on a documented procedure [IEEE 610, IEEE 1028]. See also <i>peer review</i> .	inspektion: En typ av kollegial granskning som innefattar en visuell undersökning av dokument med syfte att hitta fel, t.ex. avvikelser från utvecklingsstandarder eller att dokumentet inte följer dokumentation från högre nivåer. Det är den mest formella granskningstekniken och baseras därför alltid på en dokumenterad procedur. [Efter IEEE 610, IEEE 1028] Se även <i>kollegial granskning</i> .
installability: The capability of the product to be installed in a specified environment [after ISO/IEC 25000]. See also <i>Portability</i> .	installerbarhet: Förmågan hos en produkt att kunna installeras i en specificerad miljö [efter ISO/IEC 25000]. Se även <i>portabilitet</i> .
integration: The process of combining components or systems into larger assemblies.	integration: Processen att kombinera komponenter eller system till större grupper, objekt eller enheter.
interoperability: The capability of the software product to interact with one or more specified components or systems [ISO/IEC 25000]. See also <i>Functionality</i> .	interoperabilitet: Programvarans kapacitet att samverka med en eller flera specificerade komponenter eller system. [Efter ISO 9126] Se även <i>funktionalitet</i> .

interview: A conversational technique where the interviewer is asking the responder to obtain information on specified topic.	intervju: En konversationsteknik där den som intervjuar ställer frågor för att få information om visst ämne av den som svarar
iterative development model: A development lifecycle where a project is broken into a usually large number of iterations. An iteration is a complete development loop resulting in a release (internal or external) of an executable product, a subset of the final product under development, which grows from iteration to iteration to become the final product.	modell för iterativ utveckling: En process för utveckling där ett projekt är indelat i ett antal, vanligtvis många, iterationer. En iteration är ett fullständigt utvecklingsvarv som resulterar i en utgåva (intern eller extern) av en exekverbar produkt. En delmängd av den slutliga produkten växer fram för varje iteration för att till sist utgöra den slutliga produkten.
K	
Kano model: A model of customer satisfaction dividing product attributes into three categories: threshold, performance, and excitement.	Kano-modellen: En modell över kundnöjdhet som delar in produktattribut i tre kategorier: basattribut, förväntade attribut och sensationella attribut.
L	
learnability: The capability of the software product to enable the user to learn its application [after ISO/IEC 25000]. See also <i>Usability</i> .	inlärningsmöjlighet: Kapaciteten hos en programvaruprodukt att möjliggöra för användaren att lära sig dess funktionaliteter. [efter ISO/IEC 25000]. Se även <i>användbarhet</i> .
lifecycle model: A partitioning of the life of a product or project into phases [CMMI].	livscykelmodell: En uppdelning i faser av livstiden hos en produkt eller ett projekt.[CMMI]. Se även <i>livscykel för programvara</i> .
M	
maintainability: The ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment [after ISO/IEC 25000].	underhållbarhet: Hur lätt en programvara kan modifieras för att rätta fel, tillgodose nya krav, underlätta framtida underhåll eller anpassa till förändrad miljö. [efter ISO/IEC 25000].
maintenance: Modification of a software product after delivery to correct defects, to improve performance or other attributes, or to adapt the product to a modified environment [IEEE 1219].	underhåll: även förvaltning. Modifiering av en programvaruprodukt efter leverans för att rätta defekter, förbättra prestanda eller andra egenskaper, eller anpassa produkten till en förändrad miljö. [IEEE 1219]
maturity: (1) The capability of an organization with respect to the effectiveness and efficiency of its processes and work practices. See also <i>Capability Maturity Model</i> (2) The capability of the software product to avoid failure as a result of defects in the software. [ISO 9126] See also <i>reliability</i> .	mognad: (1) Förmågan hos en organisation med avseende på hur verksamma och effektiva dess processer och arbetssätt är. Se även <i>CMM</i> , <i>TMM</i> . (2) Kapacitet hos en programvaruprodukt att undvika negativa effekter som följd av fel i programvara. Se även <i>tillförlitlighet</i> .
maturity level: Degree of process improvement across a predefined set of process areas in which all goals in the set are attained [after CMMI].	mognadsnivå: Uppnådd processförbättringsgrad över en fördefinierad mängd processområden, i vilka alla mognadsmål för alla processområden måste vara uppnådda. [efter CMMI]

maturity model: A structured collection of elements that describe certain aspects of maturity in an organization, and aid in the definition and understanding of an organization's processes. A maturity model often provides a common language, shared vision and framework for prioritizing improvement actions.	mognadsmodell: En strukturerad samling aspekter som beskriver en organisations mognad som även kan användas för att definiera och förstå en organisations processer. En mognadsmodell tillhandahåller ofta ett gemensamt språk, en delad vision och ett ramverk för att prioritera förbättringsaktiviteter.
measure: The number or category assigned to an attribute of an entity by making a measurement [ISO 14598].	mått: Ett mätvärde (numeriskt värde) eller kategori som man tilldelar en egenskap hos en enhet genom att utföra mätning. [ISO14598]
measurement: The process of assigning a number or category to an entity to describe an attribute of that entity [ISO 14598].	mätning: Processen (utförandet) att tilldela ett numeriskt värde eller en kategori till en enhet för att beskriva en egenskap hos enheten. [ISO14598]
metric: A measurement scale and the method used for measurement [ISO 14598].	mätetal: Ett värde som anger hur många gånger den valda enheten ingår i ett mätvärde eller en storhet. [ISO14598]
milestone: A point in time in a project at which defined (intermediate) deliverables and results should be ready.	milstolpe: Ett delmål i ett projekt, som oftast ska vara uppfyllt innan man går vidare i projektet.
mind-map: A diagram used to represent words, ideas, tasks, or other items linked to and arranged around a central key word or idea. Mind maps are used to generate, visualize, structure, and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.	mindmap: Ett diagram som används för att representera begrepp, idéer, aktiviteter eller andra saker, som är kopplade till eller arrangerade kring ett nyckelord eller en central ide. Mindmaps används för att generera, visualisera, strukturera och klassificera idéer samt som en hjälp vid instudering, organisering, problemlösning, beslutsfattande och vid skrivande.
model: A system of assumptions, concepts and relationships between them allowing to describe (model) in an approximate way a specific aspect of reality.	modell: En mängd av antaganden, koncept och beroenden som gör det möjligt att beskriva någon verklighetsaspekt på ett ungefärligt sätt
modeling language: Any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules.	modelleringsspråk: Ett artificiellt språk som kan användas för att uttrycka information eller kunskap eller system i en struktur som är definierad av enhetliga regler.
modeling tool: A tool that supports the creation, amendment and verification of models of the software or system [Graham].	modelleringsverktyg: Ett verktyg som stödjer framtagande, rättning och verifiering av programvaru- eller systemmodeller [Graham].
moderator: The leader and main person responsible for an inspection or other review process.	moderator: Den person som är ledare och huvudansvarig för en inspektion eller annan granskningsprocess.
module: See <i>Component</i> .	modul: Se <i>komponent</i> .
MoSCoW: A prioritization technique allowing to prioritize requirements by allocating an appropriate priority expressed in the following terms: Must have, Should have, Could have and Will not have this time but will have in the future.	MoSCoW: En prioriteringsteknik som stödjer kravprioritering genom att tilldela följande prioriteringsgrader: Must have (måste vara med), Should have (bör vara med), Could have (kan vara med), Will not have (Ska inte ha denna gång men vill ha i framtiden).

N	
need: Something desired by a defined stakeholder [TGilb].	behov: Även önskemål. Något som önskas av en intressent [TGilb].
non-functional requirement: A requirement that does not relate to functionality, but to attributes such as reliability, efficiency, usability, maintainability and portability.	icke-funktionellt krav: Krav som inte relaterar till funktionaliteten i ett program utan till egenskaper såsom tillförlitlighet, effektivitet, användbarhet, underhållbarhet och portabilitet.
O	
object: In OOAD an instance of a class. See also: <i>Class, Object-oriented analysis and design.</i>	objekt: En instans i en klass (OOAD)
object diagram: In UML a diagram that depicts objects and their relationships at a point in time, typically a special case of either a class diagram or a communication diagram.	objektdiagram: Ingår i UML. Ett diagram som visar objekt och deras relationer på en specifik tidpunkt. Visar vanligtvis en ögonblicksbild av ett klass- eller kommunikations-diagram.
objective: see <i>Goal</i> .	målsättning: se <i>mål</i>
object-oriented analysis and design: A software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior. OOAD encompasses Object-oriented analysis (OOA) and Object-oriented design (OOD). OOA applies object-modeling techniques to analyze the functional requirements for a system. OOD elaborates the analysis models to produce implementation specifications.	objektorienterad analys och utveckling: En programvaruutvecklingsteknik som modellerar ett system i form av grupper av samarbetande objekt. Varje objekt representerar någon form av intresse i systemet som modelleras och kännetecknas av dess klasser, tillstånd (dataelement) och dess beteende. OOAD omfattar objektorienterad analys (OOA) och objektorienterad utveckling (OOD). OOA använder objektmodelleringstekniker för att analysera kraven för ett system. OOD förädlar analysmodellen till realiseringsspecifikationer.
OOA: see <i>Object-oriented analysis and design</i> .	OOA: se Object-oriented analysis and design
OOAD: see <i>Object-oriented analysis and design</i> .	OOAD: se Object-oriented analysis and design
OOD: see <i>Object-oriented analysis and design</i> .	OOD: se Object-oriented analysis and design
operability: The capability of the software product to enable the user to operate and control it [after ISO/IEC 25000]. See also <i>Usability</i> .	funktionsduglighet: även praktisk användbarhet, driftsduglighet. Programvaruproduktens kapacitet att möjliggöra att en användare kan använda och behärska programmet [efter ISO/IEC 25000]. Se <i>användbarhet</i> .
P	
pair Programming: A software development approach whereby lines of code (production and/or test) of a component are written by two programmers sitting at a single computer. This implicitly means ongoing real-time code reviews are performed.	parprogrammering: En metod för programvaruutveckling där varje kodrad (i produktion och/eller test) i en komponent är skriven av två programmerare vid en gemensam dator. Detta för att implicit utföra kodgranskning samtidigt som kod skrivs.
package diagram: In UML a diagram that shows how model elements are organized into packages as well as the dependencies between packages. For details refer to UML specification [OMG].	paketdiagram: Ingår i UML. Visar hur modellelement är organiserade i paket samt paketens inbördes relationer. För detaljer se UML-specifikation [OMG].

peer review: A review of a software work product by colleagues of the producer of the product for the purpose of identifying defects and improvements. Examples are inspection, technical review and walkthrough.	kollegial granskning: En granskning av en produkt, utförd av kolleger till den som tagit fram produkten, i syfte att hitta fel och föreslå förbättringar. Exempel är inspektion, teknisk granskning och genomgång.
performance: The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate [IEEE 610]. See also <i>Efficiency</i> .	prestanda: Förmågan hos ett system eller en komponent att uppfylla ställda krav på funktionalitet samtidigt som satta begränsningar för beräkningstider och systemkapacitet hålls. [Efter IEEE 610] Se även <i>effektivitet</i> .
point of view: A certain perspective on the system or requirements.	synpunkt: Ett visst perspektiv på system eller systemkrav
portability: The ease with which the software product can be transferred from one hardware or software environment to another [after ISO/IEC 25000].	portabilitet: även flyttbarhet. Anger hur lätt en programvaruprodukt kan flyttas från en hårdvaru- eller programvarumiljö till en annan [efter ISO/IEC 25000].
prioritization: A process of establishing requirement's implementation relative importance.	prioritering: En process för att upprätta den relativa betydelsen vid kravinförande
priority: An evaluation of the importance/urgency of a requirement.	prioritet: En utvärdering av hur viktigt/brådskande ett krav är.
Problem: Is the description of what a customer wishes to do in order to realize its business processes. The problem describes or helps to describe the needs of a customer.	problem: Även behov. En beskrivning av vad kunden önskar göra för att kunna genomföra sina affärsprocesser. Problemet beskriver eller hjälper till att beskriva kundens behov.
process: A set of interrelated activities, which transform inputs into outputs [ISO 12207].	process: En mängd samverkande aktiviteter som transformerar indata till utdata. [ISO 12207]
process assessment: A disciplined evaluation of an organization's software processes against a reference model [ISO 15504].	processutvärdering: En disciplinerad utvärdering av en organisations programvaruutvecklingsprocesser mot en referensmodell [efter ISO 15504].
process improvement: A program of activities designed to improve the performance and maturity of the organization's processes, and the result of such a program [CMMI].	processförbättring: En plan med aktiviteter designade för att förbättra effektiviteten och mognaden i en organisations processer, avser även resultatet av en sådan plan. [CMMI]
process model: (1) A framework wherein processes of the same nature are classified into an overall model, e.g. a development improvement model. (2) A method-independent process description of development processes.	processmodell: (1) Ett ramverk, i vilket processer av samma typ klassificeras, t.ex. en förbättringsmodell för utveckling. (2) En metoderberoende beskrivning av utvecklingsprocessen.
product: A composition of software, hardware and other outputs of the production process	produkt: En sammansättning av programvara, hårdvara och andra resultat av en produktionsprocess.

Product Backlog: A set of user stories, requirements or features that have been identified as candidates for potential implementation, prioritized, and estimated [BABOK].	Produktbacklog: en uppsättning användarberättelser, krav eller features som har identifierats som kandidater för möjlig implementering, prioriterats och estimerats [BABOK]
product requirement: A refinement of customer requirements into the developers' language, making implicit requirements into explicit derived requirements. [after CMMI]	produktkrav: En detaljering av kundkrav till utvecklarens språk, omvandlar implicita krav till explicit härledda krav. [efter CMMI]
product risk: A risk directly related to the quality of the product - potential failure areas (adverse future events or hazards) in the software or system See also <i>Risk</i> .	produktrisk: En risk direkt relaterad till produktens kvalitet – potentiella felområden (tänkbara kommande händelser och hos) i programvaran eller systemet. Se även <i>risk</i> .
project: A project is a unique set of coordinated and controlled activities with start and finish dates undertaken to achieve an objective conforming to specific requirements, including the constraints of time, cost and resources [ISO 9000].	projekt: Ett projekt är en unik mängd av samordnade och kontrollerade aktiviteter där start- och sluttid för aktiviteterna är underordnade och samstämmiga med speciella krav, såsom krav på tid, kostnad och resurser. [Efter ISO9000]
project risk: A risk that surround the project's capability to deliver its objectives See also <i>risk</i> .	projektrisk: En risk som omfattar projektets förmåga att leverera utifrån sina målsättningar. Se även <i>risk</i> .
prototype: An early sample or model built to test a concept or process or to act as a thing to be replicated or learned from. In Requirements Engineering prototypes can be used for requirements elicitation and validation.	prototyp: Ett tidigt prov eller en modell som är konstruerad för att testa ett koncept eller för att användas för att få lärdomar. Inom kravhantering kan prototyper användas för att identifiera och validera krav
Q	
QA: see <i>Quality Assurance</i> .	QA: se <i>kvalitetssäkring</i> .
quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations [IEEE 610].	kvalitet: Till vilken grad en komponent, ett system eller en process uppfyller ställda krav och/eller användares/kunds behov och förväntningar. [Efter IEEE 610]
Quality Assurance (QA): Part of quality management focused on providing confidence that quality requirements will be fulfilled [ISO 9000].	kvalitetssäkring: även kvalitetsförsäkran. Den del av kvalitetshanteringen som fokuserar på att tillgodose den konfidens som krävs för att ställda kvalitetskrav kan uppfyllas. [ISO 9000]
quality attribute: A feature or characteristic that affects an item's quality [IEEE 610].	kvalitetsegenskap: De viktigaste egenskaperna som påverkar en produkts kvalitet. [IEEE 610]
quality characteristic: See <i>Quality attribute</i> .	kvalitetskaraktäristik: Se <i>kvalitetsegenskap</i> .
Quality Control (QC): Part of Quality Management focused on fulfilling of quality requirements [ISO 9000].	kvalitetskontroll: Del av kvalitetsstyrning som fokuserar på uppfyllande av kvalitetskrav [ISO 9000].

quality management: Coordinated activities to direct and control an organization with regard to quality. Direction and control with regard to quality generally includes the establishment of the quality policy and quality objectives, quality planning, quality control, quality assurance and quality improvement [ISO 9000].	kvalitetsstyrning: Samordnande aktiviteter för att styra och kontrollera en organisation beträffande kvalitet. Styrning och kontroll beträffande kvalitet generellt, omfattar upprättandet av policy och kvalitetsmålsättningar, kvalitetsplanering, kvalitetskontroll, kvalitetssäkring och kvalitetsförbättringar. [ISO 9000]
R	
Rational Unified Process (RUP): A proprietary adaptable iterative software development process framework consisting of four project lifecycle phases: inception, elaboration, construction and transition.	Rational Unified Process: Ett patenterat tillämpbart iterativt utvecklingsprocessramverk som består av fyra projektlivscykelfaser: förberedelse, etablering, konstruktion och överlämning.
recoverability: The capability of the product to re-establish a specified level of performance and recover the data directly affected in case of failure [after ISO/IEC 25000]. See also <i>Reliability</i> .	återhämtningsförmåga: Produktens kapacitet att återuppta en specificerad nivå av prestanda och att återfå direkt påverkade data när fel inträffat [efter ISO/IEC 25000]. Se även <i>tillförlitlighet</i> .
redundancy: Multiple occurrence of the same information in different places.	redundans: Förekomst av samma information på flera ställen
release: A version of the solution released for installation and use by the customer/end users.	leverans: Även utgåva. En version av en lösning avsedd för installation och användning hos kund och/eller slutanvändare
reliability: The ability of the product to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations [after ISO/IEC 25000].	tillförlitlighet: Kapaciteten hos en produkt att utföra specificerad funktionalitet under vissa förutsättningar för en viss tidsperiod eller med ett visst antal användningar av programvaran [efter ISO/IEC 25000].
replaceability: The capability of the product to be used in place of another specified software product for the same purpose in the same environment [after ISO/IEC 25000]. See also <i>Portability</i> .	ersättningsbarhet: Kapaciteten hos produkten att användas istället för en annan specificerad produkt för samma ändamål och i samma miljö [efter ISO/IEC 25000]. Se även <i>portabilitet</i> .
requirement: ((1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a product, service, product component, or service component to satisfy a supplier agreement, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2) [after CMMI]	krav: (1) Ett villkor eller en egenskap som en användare behöver för att en lösa ett problem eller uppnå ett mål (2) ett mål för ett system eller delsystem ska uppfylla ett kontrakt, en standard, en specifikation eller andra formellt tvingande dokument. (3) En dokumenterad representation av ett villkor eller egenskap beskrivet i (1) eller (2) [Efter CMMI]
RD: see <i>Requirements Development</i> .	RD: se <i>kravutveckling</i>
RE: see <i>Requirements Engineering</i> .	RE: se <i>kravhantering</i>
RM: see <i>Requirements Management</i> .	RM: se <i>kravhantering (även kravledning)</i>

requirements acceptance: A process of formal agreement that the content and scope of the requirements are accurate and complete between all relevant stakeholders [BABOK].	kravacceptans: En process med formell överenskommelse mellan alla relevanta intressenter om att innehåll och omfattning av kraven är precisa och kompletta [BABOK].
requirements analysis: A set of tasks, activities and tools to determine whether the stated (elicited) requirements are unclear, incomplete, ambiguous, or contradictory, and then documenting the requirements in a form of consistent model.	kravanalys: En uppsättning av uppgifter, aktiviteter och verktyg för att avgöra om de fastställda (identifierade) kraven är oklara, ofullständiga, tvetydiga eller motsägande, och sedan dokumentation av kraven på ett konsekvent sätt
requirement attribute: Descriptive information about a requirement that enriches its definition beyond the statement of intended functionality. Examples include origin, rationale, priority, owner, release number, and version number [Wiegars].	kvalitetsattribut: Beskrivande information om ett krav som utökar dess definition med mer än bara beskrivning av den tilltänkta funktionaliteten. T.ex. ursprung, förklarande resonemang, ägare, utgåva och versionsnummer. [efter Wiegars]
Requirements Development (RD): Collection of activities, tasks, techniques and tools to identify, analyze and validate requirements. Includes the process of transforming needs into requirements. In CMMI model, Requirements Development is an engineering process area at Maturity Level 3.	kravutveckling: En uppsättning av uppgifter, aktiviteter, tekniker och verktyg för att identifiera, analysera, specificera och validera krav. Omfattar också processen för att omvandla behov till krav
requirements elicitation: see <i>Elicitation</i> .	kravidentifiering: Även insamling. <i>se identifiering</i> ,
Requirements Engineering (RE): A sub-discipline of systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems [Laplane]. Requirements Engineering discipline involves the following sub-processes: requirements elicitation, analysis and negotiation, specification, system modeling, requirements validation and requirements management.	kravhantering: En underdisciplin till systemutveckling och programvaruutveckling som fastställer mål, funktioner och restriktioner i system bestående av hårdvara och programvara [Laplane]. Kravhantering innefattar följande delprocesser: kravidentifiering, kravanalys och förhandling, kravspecificering, systemmodellering, kravvalidering och kravledning
Requirements Management (RM): A continuous process of documenting, analyzing, tracing, prioritizing, communicating, agreeing on requirements and managing requirements' changes. In CMMI model, Requirements Management is a Project Management process area at Maturity Level 2	kravledning: Även kravhantering. En kontinuerlig process med dokumentering, analys, spårning, prioritering, kommunikering och överenskommelse om krav samt hantering av kravändringar. I CMMI-modellen är kravledning ett processområde för projektledning på mognadsnivå 2.
requirements management tool: A tool that supports the recording of requirements, requirements attributes (e.g. priority, knowledge responsible) and annotation, and facilitates traceability through layers of requirements and requirements change management. Some requirements management tools also provide facilities for static analysis, such as consistency checking and violations to pre-defined requirements rules.	kravhanteringsverktyg: Verktyg som stödjer registrering av krav, kravattribut (t.ex. prioritet, kunskap, ansvar) och noteringar samt underlättar spårbarhet genom nivåer av krav och kravhantering och kravändringar. Vissa kravhanteringsverktyg har också stöd för statisk analys såsom kontroll av överensstämmande och överträdelser av fördefinierade krav eller regler.

requirements model: A representation of user requirements using text and diagrams. Requirements models can also be called user requirements models or analysis models and can supplement textual requirements specifications.	kravmodell: Ett sätt att representera användarkrav med hjälp av text och diagram. Kan också kallas analysmodell och vara en bilaga till textbaserad kravspecifikation
requirements phase: The period of time in the software lifecycle during which the requirements for a software product are defined and documented [IEEE 610].	kravfasen: Den tidsperiod i programvarans livscykel då kraven för programvaruprodukten definieras och dokumenteras. [IEEE 610]
requirements source: The source from which requirements have been derived. Requirements sources can be stakeholders, documents, business processes, existing systems, market etc.	kravkälla: Källan varifrån kraven härletts. Det kan vara intressenter, dokument, verksamhetsprocesser, existerande system, marknad etc.
requirements specification : A specification describing the business problem area. Requirements specification is usually provided by the customer and contains a description of the required capabilities of a solution from the customer's point of view.	kravspecifikation: En specifikation som beskriver verksamhetsproblem eller verksamhetsproblemområde (Kundkravspecifikation skrivs normalt av kunden och innehåller en beskrivning av de önskade lösningsmöjligheterna ur kundens synpunkt).
requirements traceability: The ability to define, capture and follow the traces left by requirements on other elements of the software development environment and the trace left by those elements on requirements [Pinheiro F.A.C. and Goguen J.A].	kravspårbarhet: Förmågan att definiera, fånga, och följa länkar till krav i andra element (även andra krav) i programvaruutvecklingen och vice versa.
Requirements Traceability Matrix (RTM): A document, usually in the form of a table, which correlates any two baselined documents that require a many to many relationship to determine the completeness of the relationship.	kravspårbarhetsmatris: Ett dokument, normalt i form av en tabell, som länkar två fastställda (baselined) dokument eller objekt där det finns behov av många-till-många-kopplingar för att fastställa om relationen är komplett
Requirement Validation: confirmation by examination that requirements (individually and as a set) define the right system as intended by stakeholders [after ISO/IEC/IEEE 29148].	kravvalidering: Bekräftelse genom kontroll att kraven (individuella eller som en grupp) definierar rätt system som intressenterna avsett det [after ISO/IEC/IEEE 29148].
Requirement Verification: confirmation by examination that a requirement or a set of requirements has been reviewed to ensure the characteristics of good requirements are achieved [after ISO/IEC/IEEE 29148].	kravverifiering: Bekräftelse genom kontroll att ett krav eller en uppsättning av krav har granskats för att säkerställa att egenskaper för bra krav är uppfyllda [after ISO/IEC/IEEE 29148]
review: An evaluation of a product or project status to ascertain discrepancies from planned results and to recommend improvements. Examples include management review, informal review, technical review, inspection, and walkthrough [IEEE 1028].	granskning: En utvärdering av en produkt eller systemstatus med syfte att upptäcka avvikelser från planerat resultat och för att rekommendera åtgärder. Exempel kan vara informella granskningar, inspektioner och genomgångar. [Efter IEEE 1028]

reviewer: The person involved in the review that identifies and describes anomalies in the product or project under review. Reviewers can be chosen to represent different viewpoints and roles in the review process.	granskare: En person som deltar i en granskning och som ska identifiera och beskriva avvikelser/felaktigheter i produkten eller projektet som granskas. Granskare kan utses för att representera olika kompetensområden och roller i granskningsprocessen.
risk: (1) The effect of uncertainty on objectives, whether positive or negative [ISO 31000]. (2) A factor that could result in future negative consequences; usually expressed as impact and likelihood.) [ISTQB].	risk: En faktor som kan resultera i en framtida negativ konsekvens oftast uttryckt i sannolikhet och påverkan.
risk analysis (assessment): The process of assessing identified risks to estimate their impact and probability of occurrence (likelihood).	riskanalys: En process för att analysera identifierade risker för att fastställa sannolikheten för att de inträffar samt den påverkan som fås om så sker.
risk control: The process through which decisions are reached and protective measures are implemented for reducing risks to, or maintaining risks within, specified levels.	riskkontroll: En process där beslut tas och skyddande åtgärder implementeras för att reducera riskerna till (eller behålla riskerna på) en viss nivå.
risk identification: The process of identifying risks using techniques such as brainstorming, checklists and failure history.	riskidentifiering: Processen för att identifiera risker genom att använda tekniker som idékläckning, checklistor och felhistorik.
risk level: The importance of a risk as defined by its characteristics impact and likelihood. A risk level can be expressed either qualitatively (e.g. high, medium, low) or quantitatively.	risknivå: Viktigheten för en risk definierad genom egenskaperna påverkan och sannolikhet hos risken. En risknivå kan uttryckas kvalitativt (t.ex. hög, medium, låg) eller kvantitativt.
risk Management: Systematic application of procedures and practices to the tasks of identifying, analyzing, prioritizing, and controlling risk.	riskhantering: Ett systematiskt tillvägagångssätt i syfte att identifiera, analysera, rangordna, definiera åtgärder för att åtgärda risker, hänskjuta risker, minimera risker eller avfärda risker.
risk mitigation: See <i>Risk control</i> .	riskminskning (tidigare risknedsättning): Se <i>riskkontroll</i> .
RTM: See <i>Requirements Traceability Matrix</i> .	RTM: Se <i>kravspårbarhetsmatris</i> .
RUP: See <i>Rational Unified Process</i> .	RUP: Se <i>Rational Unified Process</i> .
S	
SA: see <i>System Analysis, System Analyst</i> .	SA: se <i>Systemanalys, Systemanalytiker</i> .
safety: The capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property or the environment in a specified context of use [ISO/IEC 25000].	säkerhet: Förmågan hos ett system att uppnå acceptabla risknivåer för att människor, verksamhet, programvara, egendom och miljö inte ska komma till skada vid en speciell användning. [ISO 9126]
scalability: The capability of the software product to be upgraded to accommodate increased loads [Gerrard].	skalbarhet: Kapaciteten hos en programvara att bli uppgraderad till en anpassad ökad belastning.

scenario: (1) A projected course of action, events or situations leading to specified result. (2) An ordered sequence of interactions between specified entities (e.g. a system and an actor). (3) In UML: an execution trace of a use case.	scenario: (1) Tilltänkta åtgärdsvägar, händelser eller situationer som leder till ett specificerat resultat. (2) En ordningsföljd av samverkan mellan specificerade enheter (t.ex. ett system och en aktör). (3) En väg i ett användningsfall i UML.
scope: The extent of influence of something. Scope can apply to anything, like a specification, or a specified system or project [TGilb].	omfattning: Beskriver omfånget av något. Kan användas t.ex. i en specifikation eller ett definierat system eller produkt
scrum: An iterative incremental framework for managing projects commonly used with agile software development. See also <i>Agile software development</i> .	scrum: Ett iterativt-inkrementellt ramverk för hantering av projekt, vanligen använt i agila programvaruutvecklingsmetoder. <i>Se även agil programvaruutveckling.</i>
security: Attributes of software products that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data [ISO/IEC 25000]. See also <i>Functionality</i> .	informationssäkerhet: Egenskaper hos en programvaruprodukt med avseende på dess förmåga att förhindra otillåten användning eller åtkomst till program eller data. [ISO 9126] <i>Se även funktionalitet.</i>
sequence diagram: In UML it is a structured representation of behavior as a series of sequential steps over time. Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. For details refer to UML specification [OMG].	sekvensdiagram: Ingår i UML. En strukturerad bild av beteenden som en serie av sekventiella steg över tid. Används för att illustrera hur processer samarbetar med varandra och i vilken ordning
SMART: A mnemonic, giving criteria to guide in the setting of objectives. SMART stands for: • Specific – target a specific area for improvement. • Measurable – quantify or at least suggest an indicator of progress. • Assignable – specify who will do it. • Realistic – state what results can realistically be achieved, given available resources. • Time-related – specify when the result(s) can be achieved.	SMART: En förkortning för ett hjälpmedel för att ta fram effektiva mål. SMART betyder • Specifikt – Det ska vara tydligt vad som ska uppnås • Mätbart – kvantifiera eller åtminstone föreslå en mätindikator • Accepterat – den som ska uppfylla målet måste vara medveten om det och accepterat det • Realistiskt – Det ska vara möjligt att uppnå målet med tillgängliga resurser • Tidsbundet – Det ska finnas en tydlig deadline och prioritering för målet
SME: See <i>Subject Matter Expert</i> .	SME: <i>Se områdesexpert</i>
signoff: see <i>Requirements acceptance</i> .	överenskommelse: <i>se kravacceptans</i>
software lifecycle: The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software lifecycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. Note these phases may overlap or be performed iteratively.	livscykel för programvara: Den tidsperiod som börjar när en idé för programvaran föds och slutar när programvaran inte längre finns tillgänglig för användning.

software quality: The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs [ISO/IEC 25000].	programvarukvalitet: Den totala funktionaliteten och egenskaperna hos en programvaruprodukt med syfte att tillfredsställa fastställda och underförstådda krav. [Efter ISO 9126]
software quality characteristic: See <i>Quality attribute</i> .	kvalitetskaraktäristik för programvara: Se <i>kvalitetsegenskap</i> .
solution: The answer to the needs of a customer or business requirements.	lösning: (1) Realiserandet av ett krav. (2) En utvecklingsidé som, om den realiseras, förväntas leda till en del eller full belåtenhet för att lösa ett (definierat) problem
solution model: A model describing the solution area from different views on the system.	Lösningssmodell: En modell som beskriver en lösning av ett system sett ur olika perspektiv
solution scope: All the capabilities, features and qualities that must be delivered by a solution in order to meet the stated business needs or objectives. See also: <i>Scope</i> .	lösningens omfattning: Alla möjligheter, features och kvaliteter som måste levereras av en lösning för att möta fastställda verksamhetsbehov och målsättningar. Se också: <i>Omfattning</i>
solution specification: A specification describing the business solution area from different points of view. Specific types of a solution specification are: Functional Specification, System Requirement Specification or Software Requirements Specification.	Lösningsspecifikation: Även funktionsspecifikation, systemkravspecifikation eller Software Requirements Specification (SRS). Beskriver lösningen.
specification: A document that specifies, ideally in a complete, precise and verifiable manner, the requirements, design, behavior, or other characteristics of a component or system, and, often, the procedures for determining whether these provisions have been satisfied [IEEE 610].	specifikation: Ett dokument som specificerar (idealiskt på ett komplett, precist och verifierbart sätt) krav, design, beteende och egenskaper hos en komponent eller ett system och. Ofta ingår procedurer som används för att avgöra om dessa mål har blivit uppfyllda. [Efter IEEE 610]
sponsor: A stakeholder responsible for contracting or paying for the project.	sponsor: En intressent som är ansvarig för kontrakt eller finansiering av ett projekt.
stability: The capability of the product to avoid unexpected effects from modifications in the software [after ISO/IEC 25000]. See also <i>Maintainability</i> .	stabilitet: Kapaciteten hos en produkt att inte påverkas negativt av modifieringar i programvaran [after ISO/IEC 25000].] Se även underhållbarhet.
stakeholder: Any person who has an interest in an IT project. Project stakeholders are individuals and organizations that are actively involved in the project, or whose interests may be affected as a result of project execution or project completion. [TGilb].	intressent: En person som har intresse i ett IT-projekt. Projektintressenter är individer och organisationer som bör vara aktivt inblandade i projektet och vars intresse kan bli påverkat som ett resultat av projektgenomförande eller färdigställande. [TGilb].
standard: Formal, possibly mandatory, set of requirements developed and used to prescribe consistent approaches to the way of working or to provide guidelines (e.g., ISO/IEC standards, IEEE standards, and organizational standards) [CMMI].	standard: En mängd formella, tvingande alternativt rådgorande, krav framtagna för att säkerställa ett konsekvent arbetssätt. (t.ex. ISO/IEC standarder, IEEE standarder och organisatoriska standarder). [Efter CMMI]

state machine: A behavioral model composed of a finite number of states, transitions between those states, and actions, similar to a flow graph.	tillståndsmaskin: En abstrakt modell bestående av ett antal tillstånd och övergångar mellan dessa
state transition: A transition between two states of a component or system.	tillståndsovergång: En övergång mellan två tillstånd i ett system eller en komponent.
state machine diagram: see <i>State machine</i> .	tillståndsmaskindigram: se <i>tillståndsmaskin</i>
structure diagram: A type of UML diagram that depicts the elements of a specification that are irrespective of time. This includes class, composite structure, component, deployment, object, and package diagrams.	strukturdiagram: Ett UML-diagram som visar elementen i en specifikation oberoende av tidsaspekten. Inkluderar diagram för klasser, sammansättning, komponent, införande och paketering.
Subject Matter Expert (SME): A type of a stakeholder possessing specific experience, knowledge and expertise in a given aspect of the problem domain.	områdesexpert: Även SME. En intressent som har specifika erfarenheter, kunskap och skicklighet i ett speciellt område
suitability: The capability of the product to provide an appropriate set of functions for specified tasks and user objectives [after ISO/IEC 25000]. See also <i>Functionality</i> .	ändamålsenlighet (tidigare lämplighet): Produktens kapacitet att tillhandahålla ändamålsenlig funktionalitet för specificerade uppgifter och användarmål [efter ISO/IEC 25000]. Se även <i>funktionalitet</i> .
supplier: A person, group or organization providing the solution.	leverantör: En person, gruppering eller organisation som tillhandahåller en lösning
SysML: see <i>Systems Modeling Language</i> .	SysML: se <i>Systems Modeling Language</i> .
Systems Modeling Language (SysML): A general-purpose modeling language for systems engineering applications. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems.	Systems Modeling Language (SysML): Ett universellt modelleringsspråk för systemhanteringsapplikationer. Det stödjer specificering, analys, verifiering och validering av stor mängd system och system-av system
system: A collection of components organized to accomplish a specific function or set of functions [IEEE 610].	system: En samling komponenter organiserade på ett sätt så att en specifik funktionalitet eller mängd av funktionalitet uppnås. [IEEE 610]
System Analysis (SA): A set of activities, methods, techniques, tools focused on the translation of the business requirements into systems requirements. It describes a system and its limitations to the environment and provides a well-founded understanding of the environment and the system requirements.	systemanalys: En grupp aktiviteter, metoder, tekniker och verktyg som fokuserar på att omvandla affärs/verksamhetskrav till systemkrav. Det beskriver ett system och dess begränsningar och tillhandahåller en väl underbyggd förståelse för miljön (systemets omgivning) samt systemkrav.
system Analyst: A technically-oriented person, who researches given business problem, plans software solutions, recommends software and systems, and coordinates development to meet business or other requirements. The task of System Analyst is to develop business requirements into system requirements (expresses as technical specifications).	systemanalytiker: En tekniskt orienterad person som undersöker ett givet verksamhetsproblem, planerar och rekommenderar programvarulösningar och koordinerar utvecklingen för att uppfylla affärskrav eller andra krav. Uppgiften för en systemanalytiker är att förädla affärskrav till tekniska specifikationer.

system boundary: The boundary between a system and its context.	systemgräns: Gräns mellan ett system och dess omgivning
T	
testability: The capability of the product to enable modified software to be tested [after ISO/IEC 25000]. See also <i>Maintainability</i> .	testbarhet: Kapaciteten hos produkten att tillåta ändrad programvara att kunna testas [efter ISO/IEC 25000]. Se även <i>underhållbarhet</i> .
Testability of a requirement: The degree to which a requirement is stated in terms that permit establishment of test designs (and subsequently test cases) and execution of tests to determine whether the requirements have been met [IEEE 610].	testbara krav: Krav som är skrivna på sådant sätt att de möjliggör etablering av testdesign (och därefter testfall) samt exekvering av tester för att kontrollera om kraven har blivit uppfyllda. [Efter IEEE 610]
technical constraint: Any restrictions that are related to the architecture of the solution such as hardware and software platforms, programming language or technology, and software that must be used [BABOK].	designrestriktioner: Restriktioner som relateras till lösningsarkitekturen som plattformar för hårdvara och programvara, programspråk eller teknologi. Och programvara som måste användas [BABOK]
timing diagram: In UML a diagram that depicts the change in state or condition of a classifier instance or role over time. For details refer to UML specification [OMG]	timingdiagram: Ingår i UML. Visar tillståndändringar och hur villkor och roller ändras över tiden
traceability: The ability to identify related items in documentation and software, such as requirements with associated tests. See also horizontal traceability, vertical traceability.	spårbarhet: Möjligheten att identifiera relaterade element i dokumentation och programvara, som t.ex. systemkrav, med tillhörande tester. Se även <i>horisontell spårbarhet, vertikal spårbarhet</i> .
U	
UCP: See <i>Use Case Points</i> .	UCP: Se Användningsfallspoäng
UML: See <i>Unified Modeling Language</i> .	UML: Se <i>Unified Modeling Language</i>
Unified Modeling Language (UML): A standardized general-purpose modeling language in the field of software engineering. UML includes a set of graphic notation techniques to create visual models of software-intensive systems like use case diagrams, activity diagrams, class diagrams and many more.	Unified Modeling Language: Ett standardiserat universellt modelleringsspråk inom programvaruutveckling. UML innehåller en uppsättning av grafiska komponenter för att skapa en visuell modell av programvaruintensiva system såsom användningsfall, aktivitetsdiagram, klassdiagram med mera
understandability: The capability of the product to enable the user to understand whether the product is suitable, and how it can be used for particular tasks and conditions of use [after ISO/IEC 25000]. See also <i>Usability</i> .	förståelighet: Produktens kapacitet att hjälpa användaren att förstå huruvida produkten är ändamålsenlig och hur den kan användas för speciella uppgifter och villkor [efter ISO/IEC 25000]. Se även <i>användbarhet</i> .
usability: The capability of the product to be understood, learned, used and attractive to the user when used under specified conditions [after ISO/IEC 25000].	användbarhet: Produktens kapacitet att förstås, läras in, användas av och vara attraktiv för användaren när den används under specificerade förhållanden [efter ISO/IEC 25000].

use case: A sequence of transactions in a dialogue between an actor and a component or system with a tangible result, where an actor can be a user or anything that can exchange information with the system.	användningsfall: En sekvens av transaktioner i en dialog mellan en aktör och en komponent eller ett system med ett verkligt resultat, där en aktör kan vara en användare eller en godtycklig entitet som kan utbyta information med systemet.
use case diagram: In UML a diagram that shows use cases, actors, and their interrelationships. For details refer to UML specification [OMG].	Användningsfallsdiagram: Ingår i UML. Visar användningsfall, aktörer och deras förhållanden
Use Case Points (UCP): A software estimation technique used to calculate the estimated effort for a software development project. UCP is best applicable when the Unified Modeling Language (UML) and Rational Unified Process (RUP) methodologies are being used for the software design and development. UCP requires writing requirements for the system in a form of use cases. The software size (UCP) is calculated based on elements of the system use cases with factoring to account for technical and environmental considerations.	användningsfallspoäng: Även Use Case Points (UCP). En beräkningsteknik för att uppskatta arbetsmängden i ett utvecklingsprojekt för programvara. UCP kan bäst tillämpas när metodiker för Unified Modeling Language (UML) och Rational Unified Process (RUP) används för design och utveckling av programvara. UCP förutsätter att systemkraven skrivs i form av användningsfall. Storleken på programvaran beräknas baserat på element av systemanvändningsfall med hänsyn tagen till tekniska och miljömässiga överväganden.
user: A person who uses a software product.	användare: En person som använder en programvaruprodukt
user story: User stories are short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template: As a <type of user>, I want <some goal> so that <some reason> [after Cohn].	användarberättelse: En kort och enkel beskrivning av en funktionalitet som, ur perspektivet av en person (vanligtvis en användare eller en kund) som har behovet, beskriver den nya möjligheten. De följer normalt en enkel mall: Som en <användartyp> vill jag <något mål> så att <motiv> [efter Cohn]
V	
validation: Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled [ISO 9000].	validering: Bekräftelse genom undersökning och genom framläggande av sakliga bevis för att kraven för en specifikt avsedd användning eller specifik tillämpning har uppfyllts. [ISO 9000]
vendor: See <i>Supplier</i> .	leverantör: En person, grupp eller organisation som tillhandahåller en lösning
verification: Confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled [ISO 9000].	verifiering: Bekräftelse genom undersökning och genom framläggande av sakliga bevis för att specificerade krav har uppfyllts. [ISO 9000]
version: A specific form or variation of something.	version: En angiven, unikt definierad, form eller variant av något
vertical prototype: A prototype that is more complete elaboration of a single subsystem or function.	vertikal prototyp: En prototyp som har reducerats till ett fåtal, men detaljerade funktioner

vertical traceability: The tracing of requirements through the layers of development documentation to components.	vertikal spårbarhet: Spårning av krav genom lager av utvecklingsdokumentation ner till komponenter.
vision: An image of the project's deliverable as the solution to the stated need or problem.	vision: En bild av projektets leverabler som lösningen på ett uttalat behov eller problem
Vision Statement: A short text stating the why, what, and who of the desired product from a business point of view [after BABOK].	visionsbeskrivning: En kort text som beskriver varför, vad och vem för den efterfrågade produkten utifrån verksamhetsperspektiv [efter BABOK]
V-model: A framework to describe the software development lifecycle activities from requirements specification to maintenance. The V-model illustrates how testing activities can be integrated into each phase of the software development lifecycle.	V-modell: Ett ramverk som beskriver aktiviteterna inom programvaruutveckling. V-modellen beskriver ur ett testperspektiv hur testaktiviteter kan integreras i varje fas av livscykeln för programvaruutveckling.
W	
walkthrough: A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content [Freedman and Weinberg, IEEE 1028]. See also <i>Peer review</i> .	genomgång: En steg-för-steg-presentation utförd av författaren till ett dokument med syfte att samla information och skapa en gemensam förståelse av innehållet. [Freedman and Weinberg, IEEE 1028] Se även <i>kollegial granskning</i> .
workshop: A kind of meeting focused on specific (previously defined and announced to the participants) topic, usually involving stakeholders representing different areas or/and domains for a short, intensive period.	workshop: En mötestyp som fokuserar på ett speciellt område som har utannonserats till deltagarna. Normalt omfattar det intressenter från olika områden under en kort, intensiv, period.
X	
XP: See <i>extreme programming</i> .	XP: Se <i>extreme programming</i> .

9. Bilaga A

Förteckning över icke-normgivande källor som använts vid framtagande av denna ordlista.

[Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442- 20672-0

[BABOK] A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide),
<http://www.iiba.org/BABOK-Guide/BABOK-Guide-Online/appendix-a-glossary.aspx>

[BPMN.ORG] Object Management Group/Business Process Management Initiative

[CMM] M. Paulk, C. Weber, B. Curtis and M.B. Chrissis (1995), *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison-Wesley, ISBN 0-201-54664-7

[CMMI] M.B. Chrissis, M. Konrad and S. Shrum (2004), *CMMI, Guidelines for Process Integration and Product Improvement*, Addison Wesley, ISBN 0-321-15496-7

[Cohn] <http://www.mountaingoatsoftware.com/agile/user-stories>

[Fenton] N. Fenton (1991), *Software Metrics: a Rigorous Approach*, Chapman & Hall, ISBN 0-53249-425-1

[Freedman and Weinberg] D. Freedman and G. Weinberg (1990), *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, ISBN 0-932633-19-6.

[Gerrard] P. Gerrard and N. Thompson (2002), *Risk-Based E-Business Testing*, Artech House Publishers, ISBN 1-58053-314-0.

[Gilb and Graham] T. Gilb and D. Graham (1993), *Software Inspection*, Addison-Wesley, ISBN 0-201-63181-4.

[Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2

[Laplante] Laplante, Phil (2009). *Requirements Engineering for Software and Systems (1st ed.)*. Redmond, WA: CRC Press. ISBN 1-42006-467-3.

[OMG] OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2.

[Pinheiro F.A.C. and Goguen J.A] Pinheiro F.A.C. and Goguen J.A., *An object-oriented tool for tracing requirements*, in: *IEEE Software* 1996, 13(2), pp. 52-64

[TGilb] see: <http://gilb.com>, Planguage Concept Glossary