

## Vedlegg A: Terminologi

**Agent** (utføringsressurs) Konkret person eller selvstendig dataverktøy ("aktiv" prosess som f.eks. en web-tjener), eller grupper av slike. Brukt i prosessmodellering.

**Aktivitet** Samtidig eller sekvensielt prosess-steg, enten på overordnet prosjektnivå (en hel utviklingsfase) eller det udelelige arbeid som gjøres av en utvikler på noen programvareobjekter innenfor en utviklingsfase. Brukt i Prosessmodellering og Prosjektplan, se disse.

**Analyse** Systematisk gjennomgang av en modell for å forbedre modellens interne eller eksterne konsistens, f.eks. analyse av en prosessmodell. Kan også brukes om analyse av måledata, jfr. CHECK-fasen i PDCA.

**Arbeidsrom -- eng. workspace** Operativ og midlertidig arbeidskontekst for en aktivitet, med et sett uversjonerte programvareobjekter i form av filer eller databaser, og med tilhørende data-verktøy. Se også: Aktivitet, Programvareobjekt og Verktøy. Brukt i prosessmodellering.

**Balanced Scorecard (BSC)** Teknikk der mulige mål skrives ned og karakteriseres med indikatorer på sannsynlig nytte og/eller ulempe.

**Bruker** En som jevnlig benytter levert programvare, f.eks. en firmaansatt eller en privatperson. Vi skiller mellom interne brukere (videreforedlere) og eksterne brukere (sluttbrukere). Se Kunde for den som betaler for programvaren.

**Capability Maturity Model (CMM)** Forbedringsrammeverk laget av Software Engineering Institute (SEI) i USA i 1987-95. Det har et ledelsesorientert syn på forbedring og fokuserer på standardisering, forutsigbarhet og kontroll. Det baserer seg på at organisasjonens prosessmodenheter karakteriseres ved ett av fem nivåer, der 1 er dårligst og 5 er best. Hvert nivå har en rekke Key Process Areas (KPA) som igjen er oppdelt i underområder. En slik prosessmodenhetsvurdering ("assessment") må gjøres av sertifiserte "CMM-assessorer". Deretter kan det lages en forbedringsplan med konkrete tiltak om hva som må iverksettes for å komme opp på neste nivå. Vi kan si at nivå 3 tilsvarer en god ISO-9001 sertifisering. Se også: Forbedringsrammeverk.

**Case-basert resonnering (CBR) -- eng. case based reasoning** Kunnskapsbasert teknikk for å løse problemsituasjoner ved hjelp av analogi med tidligere problemer og løsninger, representert som "cases" (enkeltilfeller av erfaringer). Fordelen er at vi slipper å lagre og systematisere store datamengder for å kunne komme i gang med gjenbruk av konkrete erfaringer, gjerne fra en litt annen kontekst.

**Data** Meningsbærende mønstre dvs. symboler, så som som tall og bokstaver. Disse kan behandles automatisk, uten innsikt i dypere meningsinnhold, f.eks.  $2 + 4 = 6$ . Se Informasjon, som er data tillagt mening.

**Delphi-metode** Systematisk metode for å komme frem til felles anslag (estimer) for en eller flere egenskaper ved et et fenomen, ved at flere eksperter fremlegger ("gjetter"), diskuterer og gradvis harmoniserer sine anslag. Hvert anslag består av et trippel -- laveste, mest sannsynlige og høyeste anslag.

**Erfaringer** Kan være rå, innsamlede måledata (som antall feil eller påløpte kostnader) eller prosjektoppsummeringer ("post-mortems"). Vi kan også ha bearbejdede og generaliserte erfaringer i form av gjenbrukbare prosessmodeller, estimeringsmodeller, sjekklister, risikomodeller osv. Erfaringer kan også nedfelles i gjenbrukbare programmoduler etc. Se også: Erfaringsbase, Gjenbruk.

**Erfaringsbase** Logisk sentralisert arkiv der ulike erfaringer lagres, slik at de senere kan benyttes i andre sammenhenger. En erfaringsbase er et logisk begrep, og kan inneholde heterogen og distribuert informasjon. Den kan realiseres som dokumenter i filer eller via web, som tradisjonelt databaseinnhold eller i regneark, som dataverktøy med interne regler, algoritmer eller modeller, og/eller som henvisninger til ressurspersoner (via CV'er o.l.). Se også: Erfaringsorganisasjon, Gjenbruk.

**Erfaringsorganisasjon -- eng. Experience Factory (EF)** Forbedringsmetode og tilhørende (del)organisasjon for å samle inn, forvalte, analysere (generalisere), raffinere, pakke inn og spre erfaringer. Hensikten er å veilede pågående prosjekter (kort sikt) og å fremme organisatorisk læring (lang sikt). Organisasjonen deles derfor i en utviklingsdel, en erfaringshåndteringsdel (EF) og evt. en ledelsesdel. Erfaringsorganisasjonen har ansvaret for en erfaringsbase. Se også: Erfaringsbase, QIP.

**Experience Factory (EF)** Se erfaringsorganisasjon.

**Factory Acceptance test (FAT)** Siste test av programvare hos utvikleren, før den installeres hos en kunde. Jfr. senere SAT.

**European Foundation for Quality Management (EFQM)** Den europeiske kvalitetsorganisasjonen som bygger på TKL-prinsipper (se TKL), og som årlig deler ut en kvalitetspris. Dens metodeverk er dokumentert i en bok, se litteraturlisten.

**Feedback-møte — i GQM-regi** Avholdes ca. hver måned under et forbedringsprosjekt. Det konkrete formålet er å grovanalysere, presentere og tolke siste periodes måledata og andre resultater. Andre formål er å opprettholde motivasjon, og eventuelt å justere metrikk og forbedringstiltak.

**Forbedringsorganisasjon** Den delen av en organisasjon som har som hovedmål å planlegge, koordinere og gjennomføre forbedringer. Et eksempel kan være samlingen av prosesstøttegrupper (SEPGs) for CMM, eller en erfaringsorganisasjon supplert med ledelsesstøtte. Se også: Erfaringsorganisasjon, SEPG.

**Forbedringsplan** Samling av forbedringstiltak og tilhørende ressurser (budsjett, tidsplan), sammen med målformuleringer, statusanalyse og antakelser, og evalueringsmetode med suksesskriterier og måleplan. Jfr. PLAN-fasen i PDCA. Se også: Forbedringsprogram (like under).

**Forbedringsprogram, Forbedringsprosess** Serie planlagte aktiviteter for hele PDCA-syklusen (dvs. meta-prosesser). Disse kan være foreslått av en forbedringsorganisasjon, av firmaledelsen og/eller av en prosjektleder. Et forbedringsprogram forutsetter ofte bruk av et forbedringsrammeverk. En forbedringsplan (se over) vil lages som del av et slikt program. Vi kan ha et overordnet forbedringsprogram for hele bedriften og/eller mer avdelingsvise- eller prosjektspesifikke forbedringsprogrammer. Se også: Forbedringsorganisasjon, Forbedringsplan, Forbedringsprosjekt, Forbedringsrammeverk, PDCA, Prosessforbedring.

**Forbedringsprosjekt** Ekstraaktivitet som planlegger, gjennomfører, vurderer og sprer resultater fra forbedringstiltak i et utviklingsprosjekt (dvs. lokal PDCA-syklus). Se Pilotprosjekt.

**Forbedringsrammeverk (for programvareutvikling) -- eng. software process improvement framework** Overordnet metodeverk som foreskriver et sett med aktiviteter og tilhørende retningslinjer for å oppnå prosessforbedring i en utviklingsorganisasjon. Eksempler på forbedringsrammeverk er CMM, ISO-9001, QIP (med GQM og EF), SPICE og TKL — se disse.

**Forbedringstiltak** Konkrete tiltak angitt i en forbedringsplan. Disse kan være på overordnet nivå, som etablering av en erfaringsbase eller en erfaringsorganisasjon. De kan også være på prosjektnivå, f.eks. som innføring av inkrementell systemutvikling, tekniske gjennomganger eller årsaksanalyse av feildata, evt. supplert med intern skolering. Se også: Forbedringsplan.

**Gjenbruk** Systematisk bruk av generalisert informasjon/kunnskap i et annet, men beslektet område. Eksempler er programvarebiblioteker og objektorienterte rammeverk, såvel som bearbejdede prosjekterfaringer i form av anbefalt "beste praksis" (prosessmodeller, sjekklister). Se Erfaringsbase.

**Goal Question Metric (GQM)** Metode utviklet av University of Maryland og NASA i 1980-90 for å definere minimale og relevante metrikker ved et gitt forbedringstiltak, som ledd i å etablere kvantitative suksesskriterier. Vi starter med en konkret målsetting (et s.k. GQM-mål, avledet fra mer overordnede mål) til forbedringstiltaket. Deretter identifiserer vi relevante spørsmål for å finne ut om målsettingen er nådd. Basert på spørsmålene kan vi foreslå måledata og tilhørende metrikker, slik at spørsmålene kan besvares. Alt dette gjøres gjennom intervjuer og ved et spesielt GQM-skjema. Se også: Erfaringsorganisasjon, Metrikk, QIP.

**Informasjon** Tolkede data, dvs. data tillagt mening. Et eksempel er at symbolene "3M" betyr 3 meter, dvs. en lengdeangivelse og ikke logo til det amerikanske firmaet 3M.

**ISO-9001** Forbedringsrammeverk som er en sertifiseringsstandard for programvareutvikling under ISO-9000 serien. Standarden angir 19 aktivitetsområder, så som produktforvaltning, dataanalyse, kvalitetssikring osv. For hvert av disse må en bedrift kunne dokumentere skriftlige rutiner (dvs. et kvalitetssystem) for hvordan disse aktivitetene skal eller bør utføres av utviklerne. Et ISO-sertifikat må fornyes jevnlig, med noe strengere krav etterhvert. Ca. 50 norske bedrifter er sertifisert etter ISO-9001. En mer restriktiv standard er den engelske TickIt, der 15 av disse bedriftene er sertifisert (jan. 1999). Se Forbedringsrammeverk.

**Kano-modell** Diagramteknikk laget av den japanske professoren Kano på 80-tallet, der sluttbrukernes oppfatning av et produkt deles i tre kategorier: attraksjoner, tilfredsstillere og selvfølgheter.

**Kunde** Den som betaler for levert programvare, f.eks. et firma eller en distributør. Se Bruker, som benytter programvaren.

**Kunnskap** "Bruk av fakta, sannheter eller prinsipper fra studier eller undersøkelser" (fra Webster's Dictionary). Dvs. at kunnskap er bearbejdet informasjon som aktivt kan brukes til å ta avgjørelser eller å oppnå en hensikt. Kunnskap kan bl.a. representeres som regler i et data-system, f.eks. at "3M" forstås som minimum bredde på et fotgjengerfelt. Kunnskap kan representere operasjonelle ferdigheter (taus kunnskap) eller være nedskrevet i et språk (eksplisitt kunnskap). Merk at Kunnskapsforvaltning ("knowledge management") -- se dette -- er mer ambisiøst enn informasjonsforvaltning ("information management").

**Kunnskapsforvaltning** Lagring, raffinering, viderutvikling og gjenbruk av kunnskap og informasjon som finnes i en organisasjon. Slik informasjon kalles ofte for organisatorisk minne.

**Kunnskapsoverføring** Kan gjøres ved sosialisering (gjennom arbeid i grupper), ved eksternalisering (artikulering i form av skriftlige modeller eller teorier), ved kombinerings (systematisering av flere slags kunnskap) og ved internalisering (å omgjøre formalisert kunnskap til operasjonelle ferdigheter). Mye av problemene med erfaringsbaser er å få den lagrede kunnskapen internalisert hos utviklerne. Se også: Erfaringsbase, Kunnskap.

**Kvalitet — generelt** “Helheten av egenskaper en enhet har, og som vedrører dens evne til å tilfredsstillende uttalte eller underforståtte krav” (NS-ISO-8402 -- Quality Vocabulary, 1996).

**Kvalitet --- i SPIQ** “Egenskaper ved produkter og tjenester som gir fornøyde kunder og brukere over hele produktets/tjenestens levetid — dvs. kunden og brukeren har alltid rett!  
Se også: Bruker, Kunde.

**Kvalitetssystem** Dokumenterte retningslinjer og tilhørende forvaltningsprosesser (meta-prosesser) for hvordan en bedrifts virksomhetsprosess (her utviklingsprosess) skal utføres. Se f.eks. ISO-9001.

**Metrikk** Samling av egenskaper (attributter) og deres definisjoner som til sammen karakteriserer et programvareprodukt og/eller en programvareprosess, sett i forhold til et formål og en konkret omgivelse. For hver egenskap angis navn, måleskala (f.eks. ordinal eller nominal) og retningslinjer for datainnsamling og -analyse. Eksempler på egenskaper er Antall linjer kode (LOC), Antall feil per LOC eller per timeverk, Ledetid osv. Metrikker kan være direkte eller indirekte (beregnete), objektive eller subjektive. Se også: QM, Måling.

**Modell** Abstraksjon av et fenomen (f.eks. et system eller en prosess). Modellen består av komponenter, relasjoner og oppførsel. Modellen er laget i henhold med et visst perspektiv (formål og omfang), og aspekter av fenomenet er ofte forenklet. Modeller kan være fysiske (modelljernbaner, skipsmodeller), være nedskrevet på papir (matematiske likninger, regler eller figurer), eller være inni en datamaskin som dermed kan analysere eller simulere deler av fenomenet. Det siste fordrer et formelt språk til å representere modellen, se Representasjon.

**Måling** Tilordning (innsamling) av en dataverdi (måleverdi) til en attributt, som igjen er definert ved en metrikk. Gjennom slik datainnsamling samles **måledata**. Se også: Metrikk.

**Måleplan** Separat plan, eller del av prosjektplan, for å samle inn måledata ifølge definert metrikk. Se også: Metrikk, Prosjektplan.

**Organisasjonslæring** Gruppers evne til å forbedre sine resultater, basert på ny kunnskap.

**Organisatorisk minne** Se Kunnskapsforvaltning.

**Pakke** Å gjøre informasjon enkelt tilgjengelig. Dette kan innebære å legge til kontekst (f.eks. om prosjekt) eller å generalisere erfaringer i form av modeller.

**Pilotprosjekt — i SPIQ-kontekst** Mest mulig “objektiv” undersøkelse av ett eller flere forbedringstiltak (prosessendringer) i en realistisk omgivelse. I praksis er et pilotprosjekt et vanlig utviklingsprosjekt, koplet til et omliggende forbedringsprosjekt som innfører og studerer effekten av de foreslåtte tiltakene. Se også: Forbedringsprosjekt, Utviklingsprosjekt.

**Plan-Do-Check-Act (PDCA)** Syklus av handlinger for å få til kontinuerlig forbedring. PDCA ble opprinnelig definert av W. Edwards Deming og brukt i TKL, senere i alle andre forbedringsrammeverk. Vi kan ha en ytre PDCA-syklus for hele bedriften, og flere indre PDCA-sykluser for konkrete og prosjektrettede tiltak, dvs. en dobbel syklus.

**Programvareobjekt** Navngitt og datamaskinlagret dokument, som et mellom- eller sluttresultat fra ulike utviklingsaktiviteter. Eksempler er kravspesifikasjon, konstruksjonsspesifikasjon, kildekode, testdata osv. Brukt i Prosessmodellering.

**Programvarekvalitet** Kvalitet til levert programvare. Se Kvalitet.

**Programvareprosess, Utviklingsprosess — eng. software (engineering) process** Mengde av delvis ordnede aktiviteter for å utvikle eller vedlikeholde et programvaresystem. Prosessen har både tekniske og administrative komponenter. Aktivitetene er koplet til agenter (mennesker eller selvstendige dataverktøy), samt konsumerte og produserte resultater (programvareobjekter). Prosessen vil reguleres av en prosjektplan og mer detaljert av en prosessmodell (retningslinjer, sjekklister, metoder, teknikker o.l.). Se også: Prosessmodell, Prosjektplan.

**Prosessforbedring — eng. software process improvement (SPI)** Systematisk og målrettet aktivitet, gjennom et forbedringsprogram i en utviklingsorganisasjon, for å forbedre kvaliteten til levert programvare og i tilhørende programvareprosesser. Se også: Forbedringsprogram, Kvalitet.

**Prosesskvalitet** Kvalitet til en utviklingsprosess, ved at denne leverer programvare av ønsket kvalitet, eller ved at prosessen skjer raskere, billigere, mer predikativt, eller i henhold til gitte retningslinjer (s.k. prosesskonformitet). Se også: Kvalitet, Programvarekvalitet.

**Prosessmodell** Representasjon eller beskrivelse av en programvareprosess, på et gitt abstraksjons- eller detaljeringsnivå, og beskrevet i et egnet prosessmodelleringspråk (se under). Prosessmodellen vil oftest være lagret i en datamaskin eller som et flytskjema på papir.

**Prosessmodelleringspråk (PML)** Abstrakt språk som muliggjør delvis formell beskrivelse og -analyse av prosesser. IDEF0, som er en formalisert flytskjema-notasjon, er eksempel på et slikt språk. Se Prosessmodell.

**Proessorientert utviklingsomgivelse (eng. PSEE)** Dataverktøy som tilbyr et PML, samt et sett av PML-baserte verktøy for modelldefinisjon, -analyse, -endring og -utføring. Vi kan sammenlikne det med et arbeidsflytverktøy. Se Prosessmodell.

**Representasjon** Den konkrete form som en modell har (se modell), f.eks. en modellbåt i tre, en likning uttrykt ved matematiske symboler, eller et dataprogram/modell uttrykt f.eks. i et formelt programmeringsspråk eller et databaseskjema.

**Quality Improvement Paradigm (QIP)** Forbedringsrammeverk utviklet ved University of Maryland og NASA i 1980-90. Det understreker kontinuerlig forbedring i små, konkrete skritt—dvs. “start der skoen trykker mest”. Det legger vekt på aktiv brukermedvirkning og uselvsk deling av resultater. QIP benytter GQM og EF som hjelpeteknologier, se disse.

**Quality Function Deployment (QFD)** Teknikk for å følge opp brukerbehov gjennom en verdikjede (sporbarhetskjede) fra kravspesifikasjon til levert produkt. Brukes av TKL, se dette.

**Rolle** Generisk person (f.eks. en utvikler eller en testansvarlig), eller en gruppe av slike. Brukt i prosessmodellering, før binding til konkret person/gruppe gjøres.

**Site Acceptance Test (SAT)** Siste akseptansetest av levert programvare hos en kunde, jfr. tidligere FAT.

**Software Engineering Process Group (SEPG)** Støttegruppe for forbedringsarbeid i CMM, og består av 5-7 erfarne utviklere. Hver SEPG kan veilede og koordinere 5-10 konkrete forbedringsprosjekter, samt gjennomføre overordnet forbedringsarbeid. Svarer dels til Erfaringsorganisasjon, se dette.

**Statistical Process Control (SPC)** Teknikk for å justere en kontinuerlig og stabil produksjonsprosess vha. målinger, statistiske analyser intervenserende tilbakekoplinger, jfr. CA-fasene i PDCA. F.eks. kan en produksjonsprosess for biler eller flasketapping “tunes” på denne måten,

mens programvareprosesser med stor grad av ikke-reproduserbarhet faller utenfor. Del av TKL, se dette.

**Software Process Improvement and Capability Etermination (SPICE), ISO/IEC 15504** Internasjonalt, standardisert forbedringsrammeverk, hovedsaklig basert på prosessvurdering (“assessment”).

**Total kvalitetsledelse (TKL) -- eng. Total Quality Management (TQM)** Generelt forbedringsrammeverk for alle slags virksomheter, ikke bare for programvareutvikling. TKL er opprinnelig utviklet av Deming og Juran i Japan. Det er brukt av mange store selskaper i Japan og USA, og i Europa under oppsyn av stiftelsen EFQM, der den norske foreningen NKFL er med. TKL vektlegger: 1) Overordnet kvalitetsforbedring gjennom forankring og engasjement fra samtlige ansatte og med vekt på organisatorisk utvikling og læring, 2) Brukerorientert kvalitetsforbedring gjennom QFD-verdikjeder, 3) Prosjektorientert kvalitetsforbedring med kontinuerlig prosessforbedring i små skritt og lokal bruk av PDCA, og 4) Bruk av noen praktiske teknikker — de sju gamle kvalitetsteknikkene (som SPC) og de sju nye ledelsesteknikkene. Se også: Forbedringsrammeverk, Organisatorisk læring, Programvarekvalitet, QFD, SPC.

**Utføring** Faktisk gjennomføring av en prosess, jfr. DO-fasen i PDCA, i henhold til en prosjektplan og/eller en prosessmodell. I siste tilfelle kan vi ha hjelp av et arbeidsflytverktøy (del av en PSEE) som tolker en formalisert prosessmodell i et PML. Se også: Prosessmodell, Prosjektplan, PSEE.

**Utviklingsprosjekt** Vanlig prosjekt i en utviklingsorganisasjon. Se Pilotprosjekt, der et utviklingsprosjekt kan inngå.

**Utviklingsprosess** Se Programvareprosess.

**Verktøy** (utføringsressurs) Kallbart, passivt dataprogram som arbeider på programvareobjekter. Eksempler er editor, oversetter og feilfinder. Brukt i prosessmodellering.

## Vedlegg B: Bibliografi

**Klaus-Dieter Althof, Andreas Birk, Christiane Gresse von Wangenheim, Carsten Tautz:**  
“CBR for Experimental Software Engineering”. I M. Lentz, B. Bartsch-Spörl, H. D. Burkhard, S. Wess: Case-Based Reasoning Technology - From Foundations to Application. Lecture Notes in Computer Science, Vol. 1400. Springer-Verlag, 1998, side 235-254.

**Bjørn Andersen:**  
“Endringsledelse – En praktisk forbedringsprosess”,  
Cappelen Akademisk Forlag, 1997, ISBN 82-4560207-8.

**Svein L. Andersen et al.:**  
“Produktivitet i programvareprosjekter – innsamling og analyse av ressursdata og kostandsdrivere”, ITUF R-31, 1994.

**Chris Argyris:**  
“Overcoming Organizational Defenses”, Prentice Hall, 1990, ISBN 0-205-12338-4.

**Chris Argyris, Donald A. Schön:**  
“Organizational Learning II”, Addison Wesley, 1996, ISBN 0-201-62983-6.

**Erik Arisholm, J. Skandsen, K. Sagli, Dag Sjøberg:**  
“Improving an Evolutionary Development Process - A Case Study”,  
Proceedings of the EuroSPI'99 Conference, Pori, Finland, 26.-27. oktober, 1999.

**Asbjørn Aune:**  
“Kvalitetsstyrte bedrifter”,  
Ad Notam serie, Gyldendal, Oslo, 1996, ISBN 82-417-0516-6.

**Victor R. Basili:**  
“The Experience Factory and its Relationship to Other Improvement Paradigms”,  
In Ian Sommerville and Manfred Paul (Eds.): “Proc. 4<sup>th</sup> European Software Engineering Conference, Garmisch-Partenkirchen, Germany, 13--16 Sept. 1993”, Springer Verlag LNCS 717, p. 68--83.

**Victor R. Basili, Gianluigi Caldiera, Hans-Dieter Rombach:**  
“The Experience Factory”, i boken ‘Encyclopedia of Software Engineering’, redigert av John J. Marciniak, Wiley, 1994, ISBN 0-471-54004-8, side 469–476.

**Victor R. Basili, Gianluigi Caldiera, Hans-Dieter Rombach:**  
”The Goal Question Metric Paradigm”, i boken ‘Encyclopedia of Software Engineering’, redigert av John J. Marciniak, Wiley, 1994, ISBN 0-471-54004-8, side 469–476.

**A.L. Bergheim:**  
“Prosjektstyring og erfaringsoverføring”, Hovedoppgave ved Institutt for Informasjonsvitenskap, Universitetet i Bergen, 1998.

**Andreas Birk, Ralf Kempkens, Dieter H. Rombach, Günther Ruhe:**  
“The Perfect Handbook: Vol 1: Improvement Methodology, Part: PIA Experience Factory: The PEF Model”, Fraunhofer IESE report 059.97/E 1997.

**Barry W. Boehm:**

“Software Engineering Economics”,  
Prentice Hall, 1981, ISBN 0-13-822122-7.

**George E.P. Box, William G. Hunter, J. Stuart Hunter:**

“Statistics for Experimenters,”  
John Wiley and Sons, 1978.

**Thor Breien et al:**

“Estimering- og fremdriftsmålemetoder for utviklingsprosjekter”. IT-industriens  
Utviklingsforum (ITUF)-rapport 29, 1992.

**Jennie Carroll:**

“The application of total quality management to software development”,  
Information Technology & People, No. 8, Vol. 4, 1996.

**Bruce Carter, Tony Hancock, Jean-Marc Morin, Ned Robins:**

“Introducing RISKMAN Methodology”,  
NCC Blackwell 1994, ISBN 1-85554-356-7.

**Philip B. Crosby:**

“Quality is Free: The Art of Making Quality Certain”, McGraw-Hill, 1979.

**Philip B. Crosby:**

“Quality Is Still Free: Making Quality Certain in Uncertain Times”, McGraw-Hill, 1996.

**Thomas H. Davenport, and Laurence Prusak:**

“Working Knowledge: How Organizations Manage What They Know”, Harvard Business  
School Press, 1998.

**Tom DeMarco, Timothy Lister:**

“Peopleware: Productive Projects and Teams”, Second Edition, Dorset House Publishing, 1999.

**W.Edwards Deming:**

“Out of the Crisis”. MIT Center for Advanced Engineering Study, 1986

**Elisabetta DiNitto, Alfonso Fuggetta (red.):**

“Process Technology”, Special Issue of Journal of Automated Software Engineering,  
Vol. 5, No. 1, Kluwer Academic Publishers, January 1998.

**Tore Dybå, Nils Brede Moe:**

”Rethinking the Concept of Software Process Assessment”,  
Proceedings of the EuroSPI’99 Conference, Pori, Finland, 26.-27. oktober, 1999.

**Leif Edvinsson, Michael S. Malone:**

“Intellectual Capital: The proven way to establish your company’s real value by measuring its  
hidden brainpower”, Piatkus, 1997, ISBN 0-7499-1767-9.

**Khaled El Emam, Nazim H. Madhavji (red.):**

“Elements of Software Process Assessment and Improvement”, IEEE Computer Society Press,  
1999.

**Khaled El Emam, J.-N. Drouin, and W. Melo (red.):**

“SPICE: The Theory and Practice of Software Process Improvement and Capability  
Determination”, IEEE Computer Society Press, 1998.

**William A. Florac and Anita D. Carleton:**

“Measuring the Software Process: Statistical Process Control for Software Process Improvement”, Addison-Wesley, 1999.

**European Foundation for Quality Management (EFQM):**

“EFQM Excellence Model – 1999”, EFQM Brussels, Belgium, ISBN 90-5236-082-0.

**ESPRIT Office Software Best Practice (ESSI):**

-- Information Package, 1997. CEC, Brussels.

**Paul J. Feltovich, Kenneth M. Ford, Robert R. Hoffman:**

“Expertise in Context: Human and Machine”, AAAI/MIT Press, 1997, ISBN 0-262-56110-7.

**Norman Fenton, Shari L. Pfleeger:**

“Software Metrics—A Rigorous & Practical Approach”, International Thomson Computer Press, 1997.

**Norman Fenton, Shari L. Pfleeger, Robert L. Glass:**

“Science and Substance: A Challenge to Software Engineers”, Vol. 11, No. 7, July 1994, side 86-95.

**David A. Garvin:**

“Building a Learning Organization”, Harvard Business Review, July-August 1993.

**Robert L. Glass:**

“Pilot Studies: What, Why, and How?”, Journal of Systems Software, Vol. 36, 1997, side 85-97.

**Robert L. Glass:**

“Computing Calamities: Lessons Learned from Products, Projects, and Companies that Failed”, Prentice-Hall, 1999.

**Robert B. Grady:**

“Practical Software Metrics for Project Management and Process Improvement”, Prentice-Hall, 1992.

**Robert B. Grady:**

“Successful Software Process Improvement”, Prentice Hall, 1997, ISBN 0-13-626623-1.

**Fritz H. Grupe, Robert Urwiler, Narender K. Ramarapu, Mehdi Owrand:**

“The application of case-based reasoning to the software development process”, Information and Software Technology 40, 1998, page 493-499.

**Luke Hohmann:**

“Journey of the Software Professional: a sociology of software development”, Prentice Hall 1997, ISBN 0-13-236613-4.

**George P. Huber:**

“Organizational learning: a guide for executives in technology-critical organizations”, International Journal of Technology Management, Vol. 11. Nos. 7/8, 1996.

**George P. Huber:**

“Organizational learning: The contributing processes and the literatures”, Organization Science, Vol. 2, No. 1, 1991.

**Watts S. Humphrey**

“Managing The Software Process”, Addison-Wesley, 1989.

**Watts S. Humphrey**

“Managing Technical People: Innovation, Teamwork, and the Software Process”. Addison-Wesley, 1997.

**Conny Johannson, Patrik Hall, Michael Coquard:**

“Talk to Paula and Peter - They are Experienced”, in Frank Bomarius: Proceedings from the Workshop on 'Learning Software Organizations', Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslautern, 1999.

**Joseph M. Juran and A. Blanton Godfrey (red)**

“Juran’s Quality Handbook”, Fifth Edition, McGraw-Hill, 1999.

**Magne Jørgensen, Dag Sjøberg, Reidar Conradi:**

“Reuse of software development experience at Telenor Telecom Software”, Proceedings of the EuroSPI'98 Conference, Göteborg, Sverige, 17.-18. oktober.

**Robert S. Kaplan, David P. Norton,**

“The Balanced Scorecard”,  
Harvard Business School Press, 1996, ISBN 0875846513.

**Barbara Kitchenham, Lesley Pickard, Shari Lawrence Pfleeger:**

“Case Studies for Method and Tool Evaluation”,  
IEEE Software, July 1995, side 52-62.

**David A. Kolb:**

“Experience as the source of learning and development”, Prentice Hall, 1994,  
ISBN 0-13-295261-0.

**P. Kruchten:**

“The Rational Unified Process – An Introduction”, Addison Wesley, 1999.

**David B. Leake:**

“Case-Based Reasoning, Experiences, Lessons and Future Directions”, AAAI/MIT Press, 1996,  
ISBN 0-262-62110-X.

**Morten Levin, Øystein Fossen, Reidar Gjersvik:**

“Ledelse og teknologi”, Universitetsforlaget, 1994, ISBN 82-00-03297-3.

**Barbara Levitt and James G. March:**

“Organizational Learning”, Annual Review of Sociology, Vol. 14, side 319-340, 1988.

**Hans Jørgen Lied, Tor Stålhane:**

“Experience from Process Improvement in a SME”,  
Proceedings of the EuroSPI'99 Conference, Pori, Finland, 26.-27. oktober, 1999.

**Jostein Lillestøl:**

“Kvalitet: Ideer og metoder—offensiv kvalitetsutvikling”, Fagbokforlaget 1994,  
ISBN 82-7674-033-2.

**Arthur J. Lowell:**

“Improving Software Quality—An Insiders’s Guide to TQM”, John Wiley and Son, 1993.

**Richard Messnarz, Colin J. Tully (red.):**

”Better Software Practice for Business Benefit : Principles and Experiences”, IEEE Computer Society Press, 1999.

**National Institute of Standards and Technology (NITS):**

“The Standard for Integration Definition for Function Modelling (IDEF0)”, FIPS 183, Dec 1993.

**Ikujiro Nonaka, Hirotaka Takeuchi:**

”The Knowledge-Creating Company”, Oxford University Press, 1995, ISBN 0-18-509269-4.

**H.L. Oh:**

“A Changing Paradigm in Quality”, IEEE Transactions on Reliability, Vol. 44, No. 2, June 1995.

**Wanda J. Orlikowski:**

“Learning From Notes: Organizational Issues in Groupware Implementation”, in Proceedings from the CSCW 92 Conference.

**Mark C. Paulk:**

“How ISO 9001 Compares with the CMM”, IEEE Software, 12(1):74-83, January 1995.

**Mark C. Paulk, Charles V. Weber, Bill Curtis, Mary B. Chrissis:**

“The Capability Maturity Model for Software: Guidelines for Improving the Software Process” (CMM v1.1), SEI Series in Software Engineering, Addison—Wesley, 1995, 640 p.

**Perfect Consortium:**

“Goal-Oriented Measurement Using GQM”, D-BL-GQM-2-PERFECT9090, ESPRIT Project 9090 “Perfect”, 1997.

**Perfect Consortium:**

“Perfect Improvement Approach”, D-BL-PIA-2-PERFECT9090, ESPRIT Project 9090 “Perfect”, 1997.

**Michael E. Porter:**

“Konkurransesfortrinn”, Tano as, 1992, ISBN 82-518-2949-6.

**Kevin Pulford, Annie Kuntzmann-Combells, Stephen Shirlaw:**

“A quantitative approach to software management—The *ami* handbook”, Addison-Wesley Publishers, Wokingham, 1996, ISBN 0-201-87746-5.

**Rational Software Corporation:**

”Om UML og UML-verktøyet Rational Rose”, se <http://www.rational.com>.

**Marty Sanders (Red.)**

“The SPIRE Handbook: Better, Faster, Cheaper Software Development in Small Organisations”, Dublin: Centre for Software Engineering Ltd, 1998.

**Peter Senge:**

“The Fifth Discipline: The Art and Practice of The Learning Organization”, Century Business Press, 1990, ISBN 0-0-7126-5687-1.

**Peter Senge et. al:**

“The Fifth Discipline Fieldbook: Strategies and tools for building a learning organization”. Doubleday, 1994, ISBN 0-385-47256-0.

**Peter Senge et. al:**

“The Dance of Change: The Challenges to Sustaining Momentum in Learning Organizations”, Doubleday, 1999, ISBN 0-385-493223.

**Reni van Solingen, Egon Berghout:**

“The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development”, McGraw-Hill, ISBN 0-07-709553-7, 1999.

**SPICE, Software Process Improvement and Capability dEtermination, ISO/IEC 15504:**

“Software Process Assessment—Part 1: Concepts and Introductory Guide”, Version 1.0, 1998. See website at <http://www.sqi.cit.gu.edu.au/spice/>

**The Standish group report:**

The Standish Group International, Inc., <http://www.standishgroup.com/chaos.html>.

**Svein Stensaasen:**

“Hvordan nå verdensklasse i kvalitet, og holde seg der?”, HøyskoleForlaget 1996, ISBN 82-7634-041-5.

**Tom Stonier:**

“Information and Meaning”, Springer 1997, ISBN 354076139X.

**Tor Stålhane, Kari Juul Wedde:**

”SPI – Why isn’t it more used?”, Proceedings of the EuroSPI’99 Conference, Pori, Finland 26.-27. oktober, 1999.

**Tor Stålhane, Kari Juul Wedde, Tore Dybå:**

“Data Driven Improvement for SMEs”, Proceedings of the EuroSPI’98 Conference, Göteborg, Sverige, 17.-18. oktober.

**Carsten Tautz, Klaus-Dieter Althof:**

“Using Case-Based Reasoning for Reusing Software Knowledge”, in David B. Leake, Enric Plaza: Case based reasoning research and development: second international conference, ICCBR’97. Lecture notes in artificial intelligence vol. 1266, Springer Verlag.

**TickIt:**

“Issue 4.0 of TickIt Guide”, [www.tickit.org/guide.htm](http://www.tickit.org/guide.htm), 1998.

**K. Uchimaru, S. Okamoto and B. Kurahara:**

“TQM for Technical Groups - Total Quality Principles for Product Development”, Productivity Press, Portland, Oregon, ISBN 1-56327-005-6.

**Christianne Gresse von Wangenheim, Klaus-Dieter Althof, Ricardo M. Barcia, Carsten Tautz:**

“Evaluation of Technologies for Packing and Reusing Software Engineering Experiences”, IESE Report No. 055.98/E, Fraunhofer IESE 1998.

**Sami Zahran:**

“Software Process Improvement: Practical Guidelines for Business Success”, Addison-Wesley, 1998, ISBN 0-201-17782-X.

**Richard E. Zultner:**

“TQM for Technical Teams”, Communication of the ACM, Vol. 36, No. 10, October 1993.

**Agnar Aamodt, Enric Plaza:**

“Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches”. *AI Communications* 1 (1994) 39-59.