

Mått på kvalitet

-Hur IT-företag bedömer mjukvarukvalitet

Abstrakt

Denna uppsats handlar om mjukvarukvalitet. Kvalitet är ett svårdefinierat begrepp i många sammanhang och så även inom systemutveckling. Detta beror till viss del på att den produkt som ett system utgör inte är en fysisk produkt som vi kan ta i och bedöma utifrån dess fysiska attribut. Då målet med all kvalitetsutveckling är att skapa en produkt med hög kvalitet måste vi således hitta metoder och modeller för att främja detta. En sådan metod är att mäta mjukvarukvalitet. Frågan jag ställde mig inför denna uppsats var: Hur arbetar systemutvecklingsföretag i praktiken med produktkvalitet? För att svara på frågan har min forskning bedrivits i tre steg. Steg ett var en litteraturstudie i ämnet som visade att det finns väletablerade metoder och modeller för att mäta och bedöma mjukvarukvalitet. Steg två var att med utgångspunkt ur en sådan modell (ISO/IEC 9126) och m.h.a intervjuer undersöka hur man arbetar med produktkvalitet på ett antal systemutvecklingsföretag. Det sista steget var att jämföra den teoretiska modellen med resultatet av undersökningen. Slutsatsen var att man mäter vissa attribut hos mjukvaran men främst med hänvisning till kundens kravspecifikationen och inte med hänvisning till interna krav på att uppnå specifika kvalitetsmål.



1	<i>Inledning (bakgrund)</i>	3
1.1	Problemområde	4
1.2	Syfte	5
1.2.1	Delsyfte 1:.....	5
1.2.2	Delsyfte 2:.....	5
1.2.3	Delsyfte 3:.....	5
1.3	Avgränsningar	6
1.4	Språklig reservation	6
1.5	Uppsatsens disposition	7
2	<i>Metod</i>	8
2.1	Vetenskapsteori	8
2.1.1	Validitet och reliabilitet	8
2.1.2	Kvalitativ eller kvantitativ forskning	9
2.2	Val av metod	9
2.2.1	Litteraturstudie.....	9
2.2.2	Intervjun som undersökningsmetod	10
2.2.3	Alternativa metodval.....	11
2.3	Undersökningens genomförande	11
2.3.1	Intervjupersoner.....	12
2.3.2	Material.....	12
2.3.3	Procedur.....	13
3	<i>Teoretisk referensram</i>	15
3.1	Viktiga begrepp och definitioner	15
3.1.1	Mjukvarukvalitet.....	15
3.1.2	Produktkvalitet.....	16
3.2	Mått på kvalitet	17
3.2.1	Varför mått på kvalitet?	17
3.2.2	Tillämpningsområden för SQM.....	18
3.2.3	Kvalitetsmodellernas struktur	19
3.2.4	Interna och externa produktattribut.....	21
3.3	Vald modell	22
3.3.1	ISO/IEC 9126, en modell för utvärdering av mjukvarukvalitet	23
3.3.2	Functionality	24
3.3.3	Reliability	25
3.3.4	Usability.....	25
3.3.5	Efficiency.....	26
3.3.6	Maintainability.....	26
3.3.7	Portability	27
3.3.8	Utvärderingsnivåer	28
4	<i>Resultat</i>	29
4.1	Litteraturstudie	29
4.2	Intervju	29
4.2.1	Person- och företagsinformation.....	29



4.2.2	Strategiska mål för kvalitetsarbetet.....	31
4.2.3	Kvalitet på projektnivå	33
4.2.4	Produktkvalitet (kvalitetsmodellen).....	34
4.3	Modell vs. verklighet.....	36
5	Diskussion	37
5.1	Diskussion kring delsyfte 1 till 3	37
5.1.1	Delsyfte 1.....	37
5.1.2	Delsyfte 2.....	38
5.1.3	Delsyfte 3.....	39
5.2	Tänkbara orsaker till att man inte mäter kvalitet.....	41
5.2.1	Mjukvara är abstrakt.....	41
5.2.2	Ny teknologi	41
5.2.3	Kunden har dåliga kunskaper.....	42
5.3	Slutsatser.....	42
5.4	Uppslag till fortsatt forskning.....	43
5.4.1	Industrikunder är kunnigare.....	43
5.4.2	Kalitetsmodellen som konsumenthjälpmedel.....	44
5.4.3	Reflektioner över studien.....	44
6	Referenser.....	45
6.1	Böcker.....	45
6.2	Artiklar.....	46
6.3	Internetkällor.....	46
Appendix.....	47
A.1	Intervjumaterial	47
A.1.1	Frågemall.....	47
A.1.2	Figur	48
A.2	Intervjuer	49
A.2.1	Intervjupersoner.....	49
A.2.2	Intervjusvar.....	50



1 Inledning (bakgrund)

Kvalitet är ett begrepp som blivit alltmer centralt i den relativt unga IT-branschen. Detta kan bero på flera saker. En anledning är att investeringar i system numera kan räknas i sex- och sju-siffriga belopp och är förutom investeringskostnaden i många fall en framgångsfaktor för köparen av systemet. I och med att mer och mer kapital satsas på IT-projekt blir konsekvenserna av misslyckade satsningar allt större. Man kan förvänta sig att den pionjärande som rådit kommer att förbytas mot krav på att kunden får garanterad kvalitet och valuta för satsade pengar. Tabell 1:1 ger exempel på olika applikationers storlek och kostnad.

Tabell 1:1 Mjukvarustorlek och kostnad. MLOC: Million Lines Of Code (Möller & Paulish 1993, s. 2)

Produkt, Applikation	Storlek (MLOC)	Kostnad (M\$)
Mellanstort kommunikationssystem	1 - 2	50 - 100
Databassystem	0,4 - 1	9 - 22

En andra anledning är att branschen även har mognat och fler och fler företag inser vikten av kvalitet, både bland leverantörer och kunder. Bryant och Grogan (1996) menar att mjukvarukrisen kanske inte är något annat än växtvärk, en slags processpubertet och en uppväxttid. Forskarna menar alltså att branschen genomgår en mognadsprocess och förutsatt att man åtgärdar sina problem och förbättrar sina utvecklingsmetoder så kommer man att producera bättre kvalitet i framtiden.

För det tredje ser vi idag en fortsatt växande marknad för mjukvarusystem både i traditionella branscher och i helt nya affärsområden som e-handel. Utvecklingen har medfört att fler och fler funktioner i samhället har datoriserats. Bucci (1995) menar att allteftersom samhället blir mer beroende av mjukvara kommer kraven på dess kvalitet att öka.

Ett fjärde argument för begreppets aktualitet utmärks av framväxten av de standarder och metoder som kommit fram under 90-talet. ISO-9000 från *International Standards Organisation* innehåller speciella standarder för mjukvaruframställning och i Europa har även TickIT från *UK Department of Trade and Industry* haft ett visst genomslag. TickIT är speciellt inriktad på IT och mjukvara. Allra färskast är Svensk Programvaruindustri's kvalitetsmärke Spi2000. Metoder som TQM, *Total Quality Management* är ett exempel på metoder för processtyrning för kvalitet.

Till sist visar den så kallade FRISCOrapportens dystra siffror som säger att endast 10% av alla system som utvecklas är väl anpassade för människans behov (Pettersson, 1999) en mörk bild av kvaliteten på mjukvara. Alla ovanstående



argument sammantagna talar starkt för behovet av metoder och modeller som ger oss bättre möjligheter att utveckla system av hög kvalitet.

Generellt så eftersträvas kvalitet i de flesta sammanhang, i synnerhet av seriösa aktörer inom olika branscher. Kvalitetstänkandet är heller inget nytt. Redan under 1200-talet upprätthölls kvaliteten bland dåtidens producenter (hantverkare) av skråsystemet och lärlings förfarandet. Stolta hantverkare verkade som både utbildare och inspektörer skriver Ohlsson (1996).

De flesta av oss har en uppfattning om vad kvalitet är men det är ytterst svårt att finna ord som beskriver begreppet. Trots detta måste vi söka definiera kvalitet om vi ska kunna bedöma och jämföra olika produkter. Dahlbom och Mathiassen (1993) menar att frågan om kvalitet riktar sig mot artefakter, till exempel mjukvarusystem. Dahlbom och Mathiassen säger att när vi bedömer en artefakt så frågar vi oss helt enkelt, "Är den bra?". Svaret på vår fråga får vi genom en samlad bedömning av artefaktens funktionalitet, estetik och symbolik. Denna bedömning kan ske på två olika sätt där det ena sättet är baserat på metrics (mått) och det andra sättet är baserat på ett erfarenhetsmässigt omdöme. Väljer vi det första måste vi definiera begreppet kvalitet utifrån attribut som exempelvis reliability och maintainability och därefter måste vi bestämma hur vi ska mäta detta. Väljer vi det senare bedöms artefakten av någon person med kompetens och erfarenhet nog att ge ett värdefullt omdöme. Skillnaden är att genom att mäta speciella attribut försöker man skapa en objektiv bedömningsmodell emedan bedömningar utförda av kompetenta personer alltid kommer att vara subjektiva, vilket förminskar värdet av en jämförelse mellan olika bedömningar.

1.1 Problemområde

Kvalitetsarbetet ute hos system- och programutvecklingsföretag har uppmärksammats mycket på senare år och företagen själva har i många fall kommit långt i sitt arbete. Edvardsson (1992) talar om fyra olika stadier som visar hur långt man kommit i sin kvalitetsutvecklingen. De fyra stadierna är; kvalitetsinspektion, kvalitetsstyrning, kontinuerliga kvalitetsförbättringar och design för kvalitet. Utan att gå in närmare på de fyra faserna så handlar det främst om verksamhetsstyrning och projektstyrning utifrån ett kvalitetsperspektiv. Utgångspunkten för min uppsats är att om man inte är klar över vad som kännetecknar en kvalitativ mjukvaruprodukt så är det ytterst svårt att planera och ta fram riktlinjer för det kvalitetsarbete som ska generera den kvalitetsprodukt man tänkt sig. Redan fas ett där man inspekterar sin produkt bygger på att man vet vilka attribut man ska inspektera. Som ett komplement till de processtyrningsmetoder som finns kan man ta fram mätvärden på resurser, processer och de produkter man producerar för att kunna bedöma kvaliteten.

Efter att ha börjat studera ämnet mjukvarukvalitet började jag snart intressera mig mer och mer för den inriktning som framhäver nyttan av att mäta kvalitet. Detta angreppssätt som är så grundläggande inom många andra branscher där man till



exempel mäter olika materials hållfasthet och temperaturkänslighet. Vi har nog alla sett IKEA's maskin som testar hur många gånger en skrivbordslåda tål att dras ut under en viss belastning. För mjukvara blir testen inte lika handgripliga men behovet av kvalitetstester finns och detta ville jag studera närmare. Samlingsnamnet för denna forskning är SQM, *Software Quality Metrics* och det är ur detta som min uppsats tar sitt avstamp.

Mjukvarubranschen har ibland framställts som en bransch där kvaliteten inte alltid kommit i första hand och jag ville få en uppfattning om man var man befinner sig idag. Finns det mer att göra? Finns det nya teorier att använda? De modeller för kvalitetsmätning som jag fann i litteraturen är inte helt statistiska och det man mäter kan vara olika i olika sammanhang eller för olika mjukvaruprodukter. Mot bakgrund av detta ställde jag mig frågan hur man arbetar med produktkvalitet i verkligheten, hur tillämpas teorierna? Den "verklighet" jag var intresserad av var den typ av företag som utvecklar programvara och det vi i dagligt tal brukar kalla datasystem. Att modellerna inte är helt statistiska ger ett visst utrymme för olika tillämpningar och en enbart teoretisk studie hade inte gett ett svar på frågan hur man i praktiken tillämpar modellerna och teorierna. En empirisk undersökning av mjukvaruproducerande företag har på så vis givit ytterligare kunskaper om var man befinner sig idag och hur man arbetar med produktkvalitet.

1.2 Syfte

Syftet med denna uppsats är att finna en modell för bedömning av mjukvaruprodukter, undersöka hur man i systemutvecklingsföretag arbetar med produktkvalitet rent praktiskt och jämföra resultatet av studien. För att uppfylla syftet har jag angripit problemet i tre delsyften.

1.2.1 Delsyfte 1:

Att med hjälp av en förstudie av litteratur och forskning kring mjukvaruproduktkvalitet finna en väletablerad teoretisk modell för bedömning mjukvaruprodukter.

1.2.2 Delsyfte 2:

Att mot bakgrund av förstudien kartlägga och beskriva hur systemutvecklingsföretag arbetar med mjukvaruproduktkvalitet inom systemutveckling.

1.2.3 Delsyfte 3:

Att jämföra den av litteraturstudien funna kvalitetsmodellen med hur systemutvecklingsföretagen arbetar med produktkvalitet.

Arbetet i denna uppsats är alltså indelat i tre steg. Dels ämnar jag genom en litteraturstudie finna en kvalitetsmodell för bedömning av mjukvaruprodukter, dels tänker jag att med hjälp av intervjuer samla in information om hur man i dagsläget arbetar med produktkvalitet i systemutvecklingsföretag och tills sist har jag för



avsikt att jämföra resultatet av intervjuerna med den teoretiska kvalitetsmodell som litteraturstudien ger.

1.3 Avgränsningar

Då kvalitet är ett väldigt vitt begrepp som påverkas av allt som sker i en organisation kan en avgränsning vara på sin plats för att behålla fokus i uppsatsen. Fenton och Pfleegar (1996) menar att kvaliteten på en mjukvaruprodukt kan sägas vara resultatet av de processer och de resurser man har tillgång till. Faktorer som påverkar detta är exempelvis tillämpningen av processtyrningsmodeller och arbetsmiljö och trivsel. Men av grundläggande vikt är även att man kan bedöma kvaliteten på den produkt man levererar.

Jag begränsar mitt arbete i denna uppsats till att gälla produktkvalitet och de faktorer som bidrar till möjligheten att bedöma kvaliteten på en mjukvaruprodukt. Jag kommer därför inte att fördjupa mig i hur man kan förbättra sina processer och miljö. Att jag har valt att fokusera på produktkvalitet beror på följande: I mitt tycke var produktkvalitet mest intressant. Det är trots allt produkten som för en utomstående blir måttstocken på om företaget står för kvalitet, om produkten blir dålig spelar det ingen roll för kunden hur bra företagets interna processer är. Processerna och resurserna påverkar naturligtvis också slutprodukten men inom ramen för denna uppsats skulle en ansats som berör alla aspekter på kvalitet te sig alldeles för omfattande.

1.4 Språklig reservation

Jag hade initialt bestämt mig för att skriva magisteruppsatsen på svenska och kom tidigt igång med detta. Vartefter jag läste litteratur och artiklar visade det sig dock att nästan all litteratur jag fann var skriven på engelska, även den som var författad av svenska forskare. Detta var på intet sätt varken förvånade eller ett problem vid inläsningen av litteraturen. Dock har det visat sig något problematiskt att översätta vissa för teoridelen specifika termer då jag inte funnit någon svensk litteratur. Jag har även varit i kontakt med SIS, Standardisering I Sverige och Swedac, som ackrediterar svenska företag att utfärda ISO-certifikat, för att få en svensk kopia av ISO/IEC 9126 dock utan framgång (IEC, International Electrotechnical Commission). Att översätta viktiga termer med hjälp av ordbok ansåg jag inte vara ett alternativ då det kan innebära att den egentliga betydelsen går förlorad. Samtidigt kommer den läsare som eventuellt vill studera vidare i ämnet inte att känna igen de centrala termer som återfinns i den engelska litteraturen.

Med hänvisning till ovanstående resonemang har jag i de talrika fall där en svensk översättning inte har givit upphov till ökad förståelse valt att använda engelska och därmed i sammanhanget vedertagna uttryck. Detta trots att det i vissa sammanhang givit upphov till en något förfulande svengelska.



1.5 Uppsatsens disposition

Kapitel 1, *Inledning*, ger en introduktion till det valda ämnet för uppsatsen. Uppsatsens syfte och delsyften preciseras följt av en avgränsning.

Kapitel 2, *Metod*, inleds av en kort beskrivning av den vetenskapliga undersökningens grunder och viktiga fallgropar och ställningstaganden i forskningsprocessen. Därefter presenteras val av metod och en diskussion kring valet och alternativa metodval. Till sist finns en utförlig genomgång av denna undersökningens genomförande.

Kapitel 3, *Teoretisk referensram*, ger inledningsvis en beskrivning och definition av för uppsatsen centrala begrepp. Vidare beskrivs det teoretiska underlag som ligger till grund för uppsatsens undersökning. Kapitlet avslutas med en ingående beskrivning av den modell som författaren valt som utgångspunkt för jämförelsen mellan teori och verklighet. Denna modell kan också ses som ett resultat av delsyfte 1.

Kapitel 4, *Resultat*. Tyngpunkten i kapitlet ligger på en sammanfattande redovisning av resultatet av den empiriska undersökningen. Empirin består av en intervjuundersökning som avser att visa hur IT-branschen idag i praktiken arbetar med mjukvaruproduktkvalitet. Resultatet redovisar de huvudsakliga ståndpunkterna i respondenternas intervjusvar utan djupare diskussion. Denna diskussion förs istället i kapitel 5.

Kapitel 5, *Diskussion*. Kapitlet inleds med ingående diskussioner kring delsyftena där resultatet analyseras och ställs mot den teoretiska referensramen. Denna analys leder fram till en diskussion kring orsakerna till resultatet följt av hela undersökningens slutsatser. Avslutningsvis diskuteras uppslag till fortsatt forskning och reflektioner över uppsatsarbetets genomförande.

Håll till godo!



2 Metod

För att en undersökning ska kunna göra anspråk på att vara vetenskapligt krävs det enligt Lundahl och Skärvad (1992) dels att faktainsamlingen måste ha skett på ett vetenskapligt sätt; att den gjorts i syfte att utveckla, verifiera eller falsifiera teorier.

”En vetenskaplig undersökning är:

- inriktad på att ge teoretiska bidrag
- upplagda och genomförda med vetenskapliga arbetsmetoder” (Lundahl & Skärvad, 1992, s.36)

I metodkapitlet kommer jag att redovisa mitt tillvägagångssätt och argumentera för de vetenskapliga arbetsmetoder som jag använt mig av. ”Syftet med en detaljerad redovisning är dels replikation och dels evaluering. Med replikation menas att metoden för någon annan ska vara möjlig att upprepa under exakt identiska förhållanden. Evaluering innebär här en värdering av det empiriska förfarandet.”(Backman, 1998, s.37)

2.1 Vetenskapsteori

2.1.1 Validitet och reliabilitet

För att förhindra uppkomsten av mätfel i en undersökning bör man vara medveten om de fallgropar som finns. Det kan uppstå både slumpmässiga och systematiska fel. Dahmström (1996) menar att slumpmässiga fel är den effekt vi får om vi mäter samma element flera gånger med samma mätinstrument men att resultatet vi får blir olika. Detta brukar kallas för graden av tillförlitlighet eller reliabilitet. Dahmström (1996) menar vidare att systematiska fel är de fel som uppstår om den variabel som vi använder ej är ett lämpligt mått på den undersökta egenskapen. Det gäller således att mäta det man verkligen avser att mäta för att nå en hög validitet.

Ovan nämnda mätfel kan enligt Dahmström orsakas av; mätinstrumentet, mätmetoden, intervjuaren och respondenten. Mätinstrumentet vid en kvalitativ intervju är de frågor som ställs till respondenten och målet är att frågorna ska vara relevanta och att de mäter det de var avsedda att mäta, det vill säga att frågorna har hög validitet. Man bör därför vara försiktig med frågor som är ledande eller värdeladdade. Mätmetoden kan ge upphov till mätfel som beror på oklarheter i frågorna. Dahmström menar att här har intervjun som metod en fördel då intervjuaren omedelbart kan reda ut oklarheter kring frågorna.

Intervjuaren kan ha en stor inverkan på respondenten, i både positiv och negativ mening, man talar om intervjuareffekt. Det sociala tryck som intervjuaren utöver genom bl.a. tonfall och minspel kan ge upphov till mätfel. Respondenten kan också känna att han/hon bör ha vissa åsikter om vissa spörsmål, så kallad prestigebias. I min undersökning kan det till exempel tänkas att huruvida man har ett



kvalitetssystem och hur det tillämpas är en prestigeladdad fråga. Respondenten kan även utgöra en källa till mätfel. Respondenten kan avsiktligt eller oavsiktligt ge felaktiga svar.

Man bör vara medveten om detta då man utformar och genomför en undersökning för att minimera riskerna för försämrad validitet och reliabilitet vid själva informationsinsamlandet. Det är också viktigt att vara medveten om riskerna vid analysen av materialet så att man inte drar felaktiga slutsatser av "rätt" data.

2.1.2 Kvalitativ eller kvantitativ forskning

En viktig fråga i all vetenskaplig forskning är valet av metod. Det första valet vi då ställs inför är valet mellan ett kvalitativt och ett kvantitativt angreppssätt. Valet av metod får inte bero på slentrian eller tradition utan "det viktiga är att man väljer den metod som passar bäst för den frågeställning man arbetar med." (Holme & Solvang, 1991, s.84) Det är alltså syftet och frågeställningen som styr valet av metod, man måste överväga vilken metod som bäst kan ge svaret på det man vill undersöka. Trost (1993) menar att man något förenklat kan säga att om man tänker i banor av siffror och förhållanden som antyder ord som mer, fler eller längre så är man inne på ett kvantitativt tänkande. Trost menar vidare att om undersökningen syftar till att förstå eller att hitta mönster så ska man göra en kvalitativ studie.

Syftet med denna uppsats är att finna en modell för bedömning av mjukvaruprodukter, undersöka hur man i systemutvecklingsföretag arbetar med produktkvalitet rent praktiskt och jämföra resultatet av studien. Mot bakgrund av detta syfte ansåg jag att jag borde anta ett kvalitativt angreppssätt. Det första delsyftet uppfyllde jag med en litteraturstudie och resultatet var att jag fann en lämplig modell. Delsyfte två uppfyllde jag med ett antal kvalitativa intervjuer då jag fick information om hur man ute på olika företag söker kvalitetssäkra sina produkter.

2.2 Val av metod

Jag har i detta arbete använt mig av två kvalitativa metoder. Jag kommer nedan att beskriva de två metoderna och argumentera för valet av metod.

Metoderna är:

- Litteraturstudie
- Intervjuer

2.2.1 Litteraturstudie

I alla vetenskapliga arbeten ingår ett visst mått av litteraturgranskning. Det är en nödvändig process för att förvärva kunskap i det valda ämnet. Den kunskap som litteraturgranskningen ger oss hjälper oss i flera avseenden. Kunskaperna hjälper oss dels att formulera en forskningsbar problemställning, dels att hitta luckor i kunskapsmassan och dels att visa hur begrepp definierats, preciserats och används skriver Backman (1998). De luckor man hittar kan ge uppslag till intressanta områden att själv undersöka närmare. Att man genomför en litteraturgranskning kan



sägas vara så självklart att kommentarer är överflödiga men jag vill hävda att i mitt fall har litteraturgranskningen haft en viktigare roll än att bara sammanfatta ”vad som finns på området”.

Jag bestämde mig tidigt för att koncentrera mig på produktkvalitet och därmed blev litteraturgranskningen också ett viktigt led i processen att uppfylla delsyfte 1, att finna en väletablerad modell för bedömning av mjukvarukvalitet. Denna process gick bland annat ut på att jämföra och analysera de olika modeller som jag fann i litteraturen för att tillslut välja ut en. Litteraturstudien har även gett mig den kunskap och begreppsflora som krävs för att göra intervjuerna meningsfulla där det visat sig vara avgörande att använda en exakt terminologi och även i hög utsträckning kunna förklara speciella begrepp.

Att finna litteratur på området var något besvärligt och de mest betydelsefulla böckerna fann jag via fjärrlån från Mitthögskolan i Östersund och Högskolan i Luleå och på Chalmers Bibliotek. Artiklar från ABI/Inform Global artikeldatabas på Ekonomiska Biblioteket i Göteborg gav nyttig bakgrundsinformation. Sökningen av litteratur och böcker har skett med sökord som: ”software quality”, ”software metrics” och ”quality metrics”.

Resultatet av litteraturgranskningen och vald modell finns återgivet i kapitel 3.

2.2.2 Intervjun som undersökningsmetod

Den metod som jag ansåg bäst lämpad att tjäna mitt andra delsyfte var intervjun. Informationen inhämtas då genom att intervjuaren ställer frågor till intervjupersonen, i fortsättningen kallad respondenten, svaren från respondenten utgör undersökningens rådata.

Man brukar skilja mellan olika typer av intervjuer med avseende på olika särdrag. Ett av dessa särdrag är enligt Lundahl och Skärvad (1992) huruvida intervjun är standardiserad eller icke-standardiserad. En i hög grad standardiserad intervju följer en på förhand bestämd frågeformulering och ordningsföljd. Det är då viktigt att frågeformuleringen och ordningsföljden är densamma vid alla intervjutillfällen med de olika respondenterna. Syftet med hög standardisering är att svaren ska vara jämförbara vilket ger möjlighet till kvantitativ bearbetning. Den mest extrema formen av standardisering innebär att även svarsalternativen är bestämda i förhand vilket i praktiken närmast innebär en intervjuarledd enkätundersökning med slutna svar.

En icke-standardiserad intervju är avsevärt mera öppen, intervjuaren väljer då frågorna mera fritt. Huvudförutsättningen är att svaren täcker det informationsbehov som intervjuaren har. Den friaste formen innebär ett planlöst samtal vilket nästan enbart förekommer i psykologiska djupintervjuer. Givetvis ligger de flesta intervjuer mellan ytterligheterna och man pratar då om semistandardiserade intervjuer. I semistandardiserade intervjuer kan exempelvis frågorna vara bestämda men man



följer upp med följdfrågor i syfte att få djupare förståelse och man ges också möjligheten att följa upp intressanta sidospår.

Vid genomförandet av mina intervjuer har jag haft ett semistandardiserat angreppssätt. Min avsikt har varit att få information om respondenternas metoder och arbetssätt. Jag har alltså inte behövt följa frågemallen slaviskt utan jag har valt en metod som medger exempelvis möjligheten att ställa följdfrågor för att utveckla resonemang.

2.2.3 Alternativa metodval

Den empiriska delen i uppsatsen bygger på en kvalitativ intervjuundersökning. Valet har dock inte varit självklart, utan har stått mellan i huvudsak tre alternativ: *enkät*, *deltagande observation* eller *intervju*.

En enkät hade gett fördelen att jag kunde nå ut till fler respondenter vilket hade gett ett större svarsunderlag och möjligheten att dra statistiskt säkrade slutsatser. Samtidigt är min uppfattning att en enkätundersökning inte skulle gett tillfredsställande information även om en sådan undersökning hade givit mer generaliserbar data. Uppfattning grundar jag på att mitt ämne är så specialiserat att det varit av stor vikt att dels komma i kontakt med rätt person och att metodvalet har givit möjlighet att utveckla resonemang och förklara frågor som eventuellt varit oklara.

En deltagande observation skulle i mitt fall kunna ha genomförts där jag som deltagare i ett systemutvecklingsprojekt kunde följt kvalitetsarbetet "hands-on". Observationen hade gett data med hög reliabilitet då jag med egna ögon hade kunnat bevittna arbetet. Jag anser dock att observationen hade varit svår genomförbar på grund av främst två anledningar. Dels hade observationen varit väldigt tidskrävande då många projekt pågår under flera månader. Dels är jag tveksam till om något företag hade varit villigt att ställa upp och låta sig kritiskt granskas.

Valet av metod föll således på att göra intervjuer. Valet av intervju framför enkät berodde främst på följande: Semistandardiserade intervjuer ger både intervjuaren och respondenten stora möjligheter att beskriva förlopp och processer vilket passade syftet väl. Det har också varit viktigt att metoden medgivit möjligheten att förklara begrepp och omformulera frågor. Att ha intervjuat respondenterna öga mot öga talar också för högre reliabilitet jämfört med en enkätundersökning. Fördelen gentemot deltagande observation har främst varit att jag fått ett bredare underlag.

2.3 Undersökningens genomförande

Under de följande rubrikerna kommer undersökningens förfarande att i detalj beskrivas vilket ökar möjligheterna till replikation av undersökningen.



2.3.1 Intervjupersoner

Urvalet av intervjupersoner har inneburit ett arbete i två steg, dels att välja ut och kontakta ett antal företag intressanta ur mitt perspektiv och dels att finna ”rätt” intervjuperson på respektive företag. Mitt val av intervjupersoner har varit ickeslumpmässigt eftersom syftet med en kvalitativ undersökning är att som Holme och Solvang (1991) menar få en bredd och djupare uppfattning av det fenomen man studerar. En viktig förutsättning för en bra intervju är alltså att man intervjuar ”rätt” personer i förhållande till det syfte man har. För att nå ”rätt” personer har jag först kontaktat ett antal företag som arbetar med systemutveckling.

Urvalet av företag som jag intervjuade skedde utifrån två huvudkriterier:

- Den ena förutsättning när jag valde vilka företag jag skulle kontakta var att de arbetade med någon typ av program- eller systemutveckling.
- Det andra var att utvecklingsprocessen således ledde fram till en mjukvaruprodukt.

Denna ganska vida urvalsgrupp medgav att de flesta av för oss välkända IT-företag var intressanta. Mot bakgrund av mina egna kunskaper om ett antal företag tillsammans med de arbetsbeskrivningar företagen presenterat på sina hemsidor kontaktade jag ett antal företag.

Nästa steg var att finna en lämplig intervjuperson på respektive företag. För att försäkra mig om detta redogjorde jag vid kontakten med de tilltänkta företagen för mina avsikter. På varje företag har jag alltså aktivt försökt få kontakt med en person som kunde tänkas ha kunskaper om både de generella riktlinjer (verksamhets-/kvalitetsstyrning) och de praktiska tillvägagångsätt som man använder vid testning i olika faser av kvalitetssäkringen. Genom detta förfarande har jag försökt att försäkrat mig om att få kontakt med kunniga personer inom varje organisation. Gemensamt för de intervjuade var alltså ingen specifik yrkesgrupp eller titel utan att de besatt den information jag eftersökte inom den organisation där de arbetade.

Nio av de här kontakterna ledde till att en intervju bokades. Totalt har jag genomfört åtta intervjuer då en intervju på kort varsel avbokades på grund av en längre tids sjukdom hos den tilltänkte intervjupersonen. Ett av de utvalda företagen visade sig vid intervjutillfället inte själva ha någon egen systemutveckling längre. Därmed gav den intervjun ingen information om hur bedömningen av produktkvaliteten utfördes i det fallet men intervjun gav ändå värdefull information på övriga områden.

2.3.2 Material

Det material jag använt vid samtliga intervjuer har varit en frågemall (se appendix A.1.1) och en figur (se appendix A.1.2). Vid genomförandet av mina intervjuer har jag följt frågemallen som på så sätt legat till grund för intervjun. Frågorna i frågemallen har i vissa fall kompletterats med följdfrågor och förtydliganden vilket framgår av intervjuutskriften (se appendix A.2.2).



Frågorna i frågemallen har syftat till att täcka in mitt informationsbehov med till en början allmänna frågor kring syfte och mål med respektive företags kvalitetsarbete till att mot slutet nå uppsatsens kärnområde; kvalitetsfaktorer. Att börja med allmänna okontroversiella frågor kan enligt Lundahl och Skärvad (1992) vara ett bra sätt att få respondenten ”varm i kläderna”. Lundahl och Skärvad menar vidare att brett formulerade så kallade processfrågor ger respondenten större spelrum vilket också kan stimulera att få svar i kontroversiella frågor. Jag vill mena att detta har haft en viss betydelse i denna undersökning då det av respondenten kan ha upplevts som känsligt att beskriva det egna kvalitetssystemets styrkor och även brister. Frågorna har därför utformats på ett sätt som har givit respondenten möjligheten att beskriva exempelvis arbetsförlopp.

Frågemallen är utformad efter fyra huvudteman där de två första har varit av allmän karaktär och de två senare av mera preciserade frågor kring kvalitetsarbetet. Frågorna i den andra halvan av frågeformuläret riktar sig mot den typ av aktiviteter som rör bedömningen av mjukvaruprodukter. Aktiviteter som vilka kvalitetsfaktorer man undersöker och hur man mäter och testar produkten (se vidare i teorikapitlet, 3.).

Som diskussionsunderlag för frågorna kring kvalitetsfaktorer har jag använt mig av den modell för mjukvarubedömning som litteraturstudien gav. Denna figur är alltså en kopia på den av ISO/IEC utformade modellen för kvalitetsbedömning av mjukvaruprodukter, ISO/IEC 9126.

2.3.3 Procedur

Som jag tidigare nämnt ska man försöka uppnå god reliabilitet i en forskningsstudie. I en kvalitativ intervju kan man enligt Lundahl och Skärvad (1992) bland annat sörja för hög reliabilitet genom att genomföra intervjuerna på så likartade sätt som möjligt och därigenom minska yttre omständigheters påverkan på intervjun.

För att uppnå detta har jag bokat in en timme hos respondenten för att det ska finnas tillräckligt med tid och möjlighet till att utveckla frågeställningar och resonemang. Samtliga intervjuer utom en har genomförts av mig personligen som besöksintervjuer på respondentens arbetsplats i avskildhet. En av intervjuerna utfördes dock som telefonintervju då den tillfrågade respondenten befann sig i Jönköping. Vid samtliga intervjuer har jag upplyst respondenten om vem jag är och att intervjun kommer att användas som forskningsunderlag till en magisteruppsats. Jag har vid samtliga intervjutillfällen utom telefonintervjun använt mig av bandspelare för inspelning av frågor och svar.

Vidare har alla förutom att svara på frågorna fått kommentera hur man tänker och arbetar med de begrepp som figuren i appendix A.1.2 visar. Figuren presenterades i samband med fråga 16 för att samtidigt ge ett diskussionsunderlag till de följande frågorna. För att även vid telefonintervjun kunna diskutera figuren mailade jag den till respondenten på förhand. Syftet med att visa själva figuren var dels att få en uppfattning om man arbetar med liknande modeller och dels att ha ett material att



diskutera kring då det var tänkbart att man arbetade med detta men att man använde andra begrepp. Figuren är ganska komplicerad och det har därför varit av stor vikt att ta fram den i ett senare skede under intervjun då respondenten blivit varm i kläderna och då ett visst förtroende uppstått mellan respondent och intervjuare.



3 Teoretisk referensram

I följande kapitel har jag för avsikt att uppmärksamma och förklara något om de teorier och forskningsresultat som denna uppsats tar sitt avstamp ur. Jag avser även att förklara de för uppsatsen centrala begrepp så att det inte råder någon tvekan om betydelsen av begreppen i detta sammanhang.

3.1 Viktiga begrepp och definitioner

Innan jag går vidare i ämnet om hur man kvalitetsbedömer mjukvaruprodukter är det viktigt att definiera följande begrepp:

- Mjukvarukvalitet
- Produkt
- Produktkvalitet

3.1.1 Mjukvarukvalitet

Det finns idag en rad skilda uppfattningar om vad mjukvarukvalitet är. Samtidigt anser vissa forskare att det börjar bli hög tid att enas om en modell eller standard att bedöma mjukvarukvalitet utifrån. Bucci (1995) menar att många är av den åsikten att om man bara följer en kvalitetsstyrningsmodell till hundra procent så garanterar det kvaliteten på produkten. Bucci säger vidare att detta inte alltid stämmer och då är objektiva metrics/mått ett måste. Det betyder att alla aktörer inom mjukvarukvalitet (producenter, användare och certifierare) **måste** enas om ett minimalt antal metrics som beskriver mjukvarukvalitet. Skiljaktigheterna kring ämnet mjukvarukvalitet speglas i den flora av definitioner som finns.

Följande är sammanställda av Jones (1996).

- Dr. Barry Boehm: *"Achieving high levels of user satisfaction, portability, maintainability, robustness and fitness of use."*
- Phil Crosby (f.d. kvalitetschef på ITT) : *"Conformance to user requirements."*
- W. Edward Deming: *"Striving for excellence in reliability and functions by continous improvment in the process of development, supported by statistical analysis of the cause of failure"*.
- Watts Humphrey (SEI): *"Achieving excellent levels of fitness for use, conformance to requirements, reliability and maintainability."*
- Capers Jones: *"The absence of defects that would make software either stop completely or produce unacceptable results. Defects can be traced to requirements, to design, to code, to documentation or to bad fixes of previous defects. Defects can range in severity from minor to major."*
- James Martin: *"Software quality means being on time, within budget and meeting user needs."*
- Tom McCabe: *"High levels of user satisfaction and low defect levels, often associated with low complexity."*
- John Musa: *"Low defect levels, adherence of software functions to user needs and high reliability."*



- Bill Perry (QAI): *"High levels of user satisfaction and adherence to requirements."*

Vad vi kan se av de ovan presenterade definitionerna är att trots de många olika varianterna kan man urskilja vissa gemensamma beröringspunkter. Flera av citaten visar på den utbredda åsikten att kvalitet till stor del handlar om att uppfylla användarnas krav och att tillfredsställa användarnas behov. Vi ser också att begrepp som reliability och maintainability återkommer på flera ställen. Den definition som jag fann mest heltäckande fann jag i ISO/IEC standard 9126.

"Software quality: The totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs." (ISO/IEC 9126:1991)

Bache och Bazzana (1994) menar att man kan utläsa två viktiga delar i definitionen, "totality of features" och "stated or implied needs". Av den första delen, "totality of features", kan vi förstå att man syftar på att begreppet kvalitet kan brytas ned i ett antal attribut av samma typ som vi kan se i några av de andra ovan citerade definitionerna, begrepp som reliability, maintainability och portability. Den andra delen "stated or implied needs" antyder att vi inte kan anta att produktspecifikationen är helt fullständig och korrekt utan det kan finnas ickefunktionella attribut som inte är specificerade men som ändå är av grundläggande vikt för kunden. Det är alltså viktigt att förstå att kvalitet oftast innebära mycket mer än att endast uppfylla kraven i specifikationen.

3.1.2 Produktkvalitet

Jag fokuserar i denna uppsats på produktkvalitet och bör därför förklara detta begrepp innan jag går vidare. Historiskt sett har man förknippat en produkt med ett fysiskt föremål men numera räknar man även in tjänster i begreppet.

"Produkt: Något som kan erbjudas på en marknad för att tillfredsställa ett behov eller önskemål."
(Kotler & Armstrong, 1996, s.5)

Med denna ganska vida produktdefinition inställer sig frågan vad då en mjukvaruprodukt är. Att det finns en marknad för olika mjukvaruprodukter råder det ingen tvekan om men som produkt betraktad är mjukvara något abstrakt. Många gånger är det enda vi ser av produkten ett användargränssnitt och i vissa inbäddade system finns inte ens det. Sommerville (1996) menar att mjukvaruprodukter är mjukvarusystem levererade till kunden med dokumentation som beskriver hur man ska installera och använda systemet, ibland tillsammans med nödvändig hårdvara. I ISO/IEC 9000-3 definieras en mjukvaruprodukt som:

"Mjukvaruprodukt: Det kompletta set av datorprogram, procedurer och tillhörande dokumentation och data avsedd för leverans till en användare"



Sommerville beskriver vidare två huvudgrupper av mjukvaruprodukter:

1. Generiska produkter. Detta är självständiga system som produceras av utvecklande organisationer och säljs på den öppna marknaden till alla potentiella kunder.
2. Specialprodukter. Detta är system som beställs av en speciell kund. Mjukvaran är sedan specialframtagen för kunden.

Generiska produkter dominerar persondatormarknaden och företag som Microsoft är producenter av denna typ av system. Det som utmärker en generisk produkt är att produktspecifikationen är internt framtagen av marknadsavdelningen hos producenten. Specialprodukter förekommer främst bland företagskunder. Marknaden för specialprodukter ökar då exempelvis datoriserade styr- och kontrollsystem numera ingår som inbäddade system i många apparater, allt från bilar till hushållsmaskiner. Längre fram i kapitlet kommer jag att beskriva modeller för bedömning av mjukvaruprodukter och det är viktigt att poängtera att de kvalitetsmodeller jag kommer att beskriva är applicerbara på båda typerna av mjukvaruprodukter.

3.2 Mått på kvalitet

Som jag tidigare nämnt så måste vi för att objektivt kunna bedöma produkter mäta olika attribut hos den aktuella produkten. Vad grundar man då den åsikten på?

3.2.1 Varför mått på kvalitet?

Fenton och Pfleeger (1996) menar att mäta mjukvara har blivit av grundläggande vikt i mjukvaruutveckling. Att mäta kvalitet kan göras på flera plan. Utvecklare mäter mjukvarukaraktäristika för att få en uppfattning om systemkraven är fullständiga, ifall systemdesignen håller hög kvalitet och koden är redo för testning. Projektledare mäter process- och produktattribut för att kunna avgöra när systemet är klart för leverans och om man håller budgeten. Kunder mäter den slutgiltiga produkten för att avgöra om den tillgodoser deras krav och ifall produkten håller hög kvalitet. Personal för underhåll måste kunna bedöma en produkt för att se vad som behöver uppgraderas och förbättras.

Att sätta mått på vår omgivning är något som sker dagligen både i professionella sammanhang, exempelvis en läkare mäter förekomsten av olika substanser för att ställa diagnos, och i vårt dagliga liv där vi uppskattar tiden det tar att färdas en bestämd sträcka med avseende på hastighet och distans. Fenton och Pfleeger menar att måtten hjälper oss att förstå vår värld, interagera med omgivningen och att förbättra våra liv. Fenton och Pfleeger säger vidare att emedan mått är vida använt och självklart inom många områden så har det ansetts vara en lyx inom mjukvaruutveckling. Fenton och Pfleeger menar att i de flesta utvecklingsprojekt gäller att:



1. Vi misslyckas med att fastställa mätbara mål för våra mjukvaruprodukter. Till exempel så lovar man att produkten ska vara användarvänlig, tillförlitlig och möjlig att underhålla utan att specificera tydligt och objektivt vad dessa termer betyder. Resultatet av detta är att vi vid projektslut inte kan säga huruvida vi uppnått målen. "Gilb's Principle of Fuzzy Targets: projects without clear goals will not achieve their goals clearly." (Fenton & Pfleeger, 1996, s.10)
2. Vi misslyckas med att förstå och kvantifiera komponentkostnaden i mjukvaruprojekt. Till exempel kan vi i det flesta projekt inte differentiera designkostnader från kodnings- och testningskostnader. Vi kan inte kontrollera kostnader ifall vi inte mäter den relativa komponentkostnaden.
3. Vi kvantifierar och förutser inte kvaliteten på de produkter vi producerar. Alltså kan vi inte tala om för en potentiell användare hur tillförlitlig produkten kommer att vara med avseende på sannolikheten för att ett fel ska uppkomma under en given tid eller hur omfattande arbete som kommer att krävas för att flytta produkten till en annan hårdvara.
4. Vi tillåter hörsägen att övertyga oss om att använda nya utvecklingsmetoder som inte har utvärderats tillräckligt grundligt.

Ovanstående punkter tillsammans med de positiva erfarenheter av mätmetoder inom många andra ingenjörsvetenskaper och det faktum att mjukvara numera är föremål för enorma investeringar talar för behovet av mätbara kvalitetsfaktorer, menar Fenton och Pfleeger.

3.2.2 Tillämpningsområden för SQM

Vid en närmare studie av ovanstående punkter utkristalliserar sig några olika klasser av olika möjliga tillämpningsområden för mått på kvalitet. Vi kan dela in de här tillämpningsområdena i tre klasser:

- Processer
- Resurser
- Produkter

Processer innebär alla de aktiviteter som rör utvecklingen av mjukvara från det att man har en idé till det att man har en färdig produkt. Systemutvecklingsprocessen innebär enligt Andersen (1994) faserna; analys, utformning, realisering och implementering. Att säkra kvaliteten inom dessa faser gör man med hjälp av olika verksamhets- och projektstyrningsmodeller. Frågor som rör processer är ofta av den typen som vi vill kunna mäta. Det kan handla om hur lång tid det tar att bli färdig med en process och hur mycket det kommer att kosta. *Resurser* innebär allt det som behövs för att utföra en process det rör sig således om personal, material, metoder och verktyg som hårdvara och mjukvara. De mått vi har nytta av här avser mått på exempelvis hur mycket personal som behövs och vad behovet av hårdvara kommer att kosta. *Produkter* är i första hand de saker man levererar till kunden men begreppet inbegriper mer än så. Alla artefakter och dokument som produceras under utvecklingsarbetet är produkter av utvecklingsprocessen och kan mätas och bedömas. Exempelvis är det vanligt att man utvecklar prototyper för att utvärdera möjliga lösningar på systemdesign.



Vi kan alltså se att SQM kan användas i flera sammanhang men i fortsättningen är den teori jag tar upp specifikt inriktad på produktbedömning.

3.2.3 Kvalitetsmodellernas struktur

Vi har sett tidigare att kvalitet är ett svårdefinierat begrepp men om man utgår från att vi vill på något sätt kunna mäta kvalitet så måste vi definiera kvalitet utifrån specifika mjukvaruproduktattribut. Vi vill med andra ord kunna mäta hur väl dessa attribut uppfylls i vår produkt. De attribut man har identifierat ligger till grund för de kvalitetsmodeller som står till buds. I litteraturen finns många exempel på sådana kvalitetsattribut, även kallade kvalitetsfaktorer, för mjukvara (se tabell 3:1).

Tabell 3:1 Sammanställning av kvalitetsfaktorer (Ohlsson 1997, s. 27)

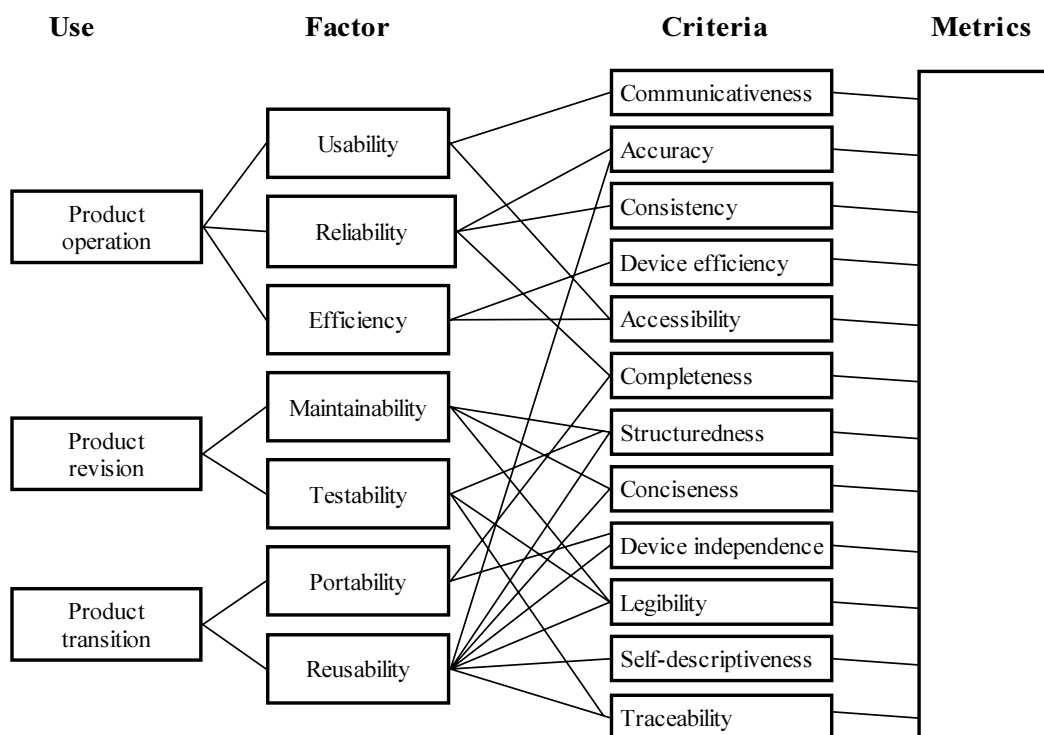
McCall, 1977	Boehm, 1978	Bowen, 1985	Murine, 1983	Övriga
Correctness		Correctness	Correctness	Correctness
Reliability	Reliability	Reliability	Reliability	Reliability
Efficiency	Efficiency	Efficiency	Efficiency	Efficiency
Usability	Human Engineering	Usability	Usability	Usability
Integrity		Integrity	Integrity	Integrity
Maintainability	Understandability	Maintainability	Maintainability	Maintainability
Flexibility	Modifiability	Flexibility	Flexibility	Flexibility
Testability	Testability		Testability	Testability
Portability	Portability	Portability	Portability	Portability
Reusability		Reusability	Reusability	Reusability
Interoperability		Interoperability	Interoperability	Interoperability
		Survivability		Survivability
			Intraoperability	Safety
		Expandability		Manageability
				Functionality
				Supportability

Vi ser i tabell 3:1 att merparten av kvalitetsfaktorerna återkommer hos flera av forskarna och skillnaderna består till största delen av antalet identifierade kvalitetsfaktorer. Andra skillnader är av mer terminologisk karaktär, till exempel kallar Boehm en kvalitetsfaktor för modifiability det alla de andra i tabellen kallar flexibility. Man kan också ifrågasätta behovet av allt för många faktorer, exempelvis kan man tänka sig att expandability skulle kunna ingå i faktorn maintainability i Bowens fall.

Begreppet kvalitet börjar nu bli lite mer hanterligt men kvalitetsfaktorerna är även de för abstrakta för att vi ska kunna mäta dem. För att vi ska kunna göra direkta mätningar krävs att vi dekomponerar ytterligare efter en slags "divide-and-conquer" strategi, vi delar upp varje begrepp för att vinna överblickbarhet och förståelse. Tanken är att varje faktor representeras av ett antal kriterier på en lägre nivå som är lättare att mäta än faktorerna. Varje kriterie i sin tur dekomponeras ytterligare i



direkt mätbara så kallade metrics. Forskningsteorierna bygger på att det finns ett samband mellan *metrics*, *factors* och *criteria*. För att kunna överblicka alla faktorer och kriterier, etc. och de relationer som sammanbinder dem så brukar man försöka fånga detta i kvalitetsmodeller. De här trädliknade modellerna (se figur 3:1) består således av vid roten kvalitetsfaktorer på en hög nivå, lätta att förstå ur användarperspektiv, och noderna är mätbara kriterier som ur teknisk synvinkel är lättare mäta.

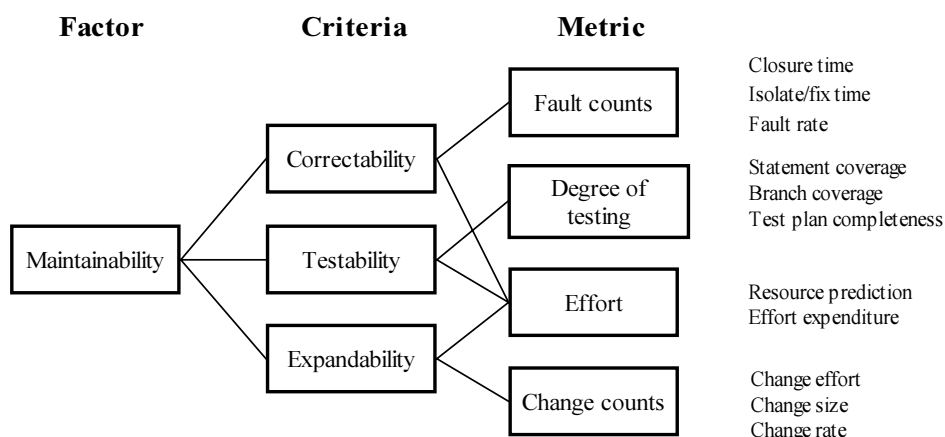


Figur 3:1 Exempel på kvalitetsmodell enligt McCall's FCM struktur.

Figur 3:1 visar en kvalitetsmodell som till sin struktur bygger på McCall's FCM, *Factor Criteria Metric* modell. Ibland lägger man till en livscykelaspekt i figuren, *use*, av McCall kallade *product operation*, *product revision* och *product transition*. Faktorerna anger vad som kännetecknar kvalitet hos en mjukvaruprodukt och det är de här vi vill kvantifiera och göra mätbara. Varje *factor* består av ett antal *criteria* som på en lägre nivå beskriver faktorerna. Varje kriterie dekomponeras till olika *metrics* och det är först i nästa steg som man verkligen tar fram mätvärden. Ett exempel på ett sådant mätvärde kan vara MTTF (mean time to failure) som ger en siffra på hur lång tid det tog innan ett fel uppstod vid en testkörning. Denna siffra härleder man då till *reliability* som ett mått på tillförlitligheten hos ett system. Denna typ av mått är speciellt viktiga i medicinska livsuppehållande system. Värt att notera



är också att i vissa kvalitetsmodeller kan ett criteria vara relaterat till flera factors, exempelvis ser vi att accessibility kan härledas ur både usability och efficiency.



Figur 3:2 Dekomponering av maintainability. (Fenton & Pfleegar 1996 s. 340)

I figur 3:2 ser vi ett exempel på hur man kan dekomponera maintainability hela vägen ner till det man verkligen mäter.

3.2.4 Interna och externa produktattribut

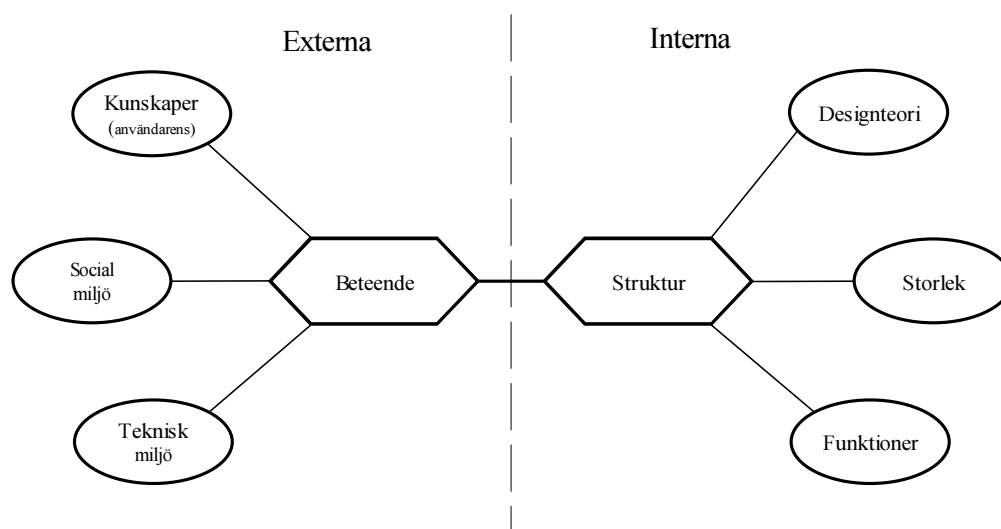
De attribut man har identifierat kan vara antingen externa eller interna. Externa attribut kännetecknas av att de beror av både produktens beteende och miljön som produkten verkar i. Till exempel, om vi vill mäta programkodens tillförlitlighet, reliability, så måste vi ta hänsyn till den dator som programmet körs på tillsammans med det sätt på vilket programmet används. Olika användare med skilda användarmönster kan uppleva programmet olika tillförlitligt. Det samma gäller för understandability eftersom användarens erfarenheter och kunskaper ger olika förutsättningar att förstå programvara och dokumentation. Ett problem med externa attribut är att det enda sättet att verkligen få veta till exempel hur många gånger systemet kraschade är att observera systemet under hela dess livscykel. Bache och Bazzana (1994) menar att detta först och främst är väldigt kostsamt men, av större vikt, informationen är helt irrelevant då systemet kommer att vara per definition avvecklat då man mätt klart.

Som systemutvecklare vill man redan under utvecklingsfasen kunna uppskatta hur systemet kommer att fungera. Vad man då gör är att man försöker förutsäga hur systemet kommer att bete sig. Man mäter då så kallade interna attribut. Interna attribut beror enbart av själva produkten. Genom att enbart studera produkten kan vi mäta interna attribut som storlek och komplexitet. Interna attribut är lättare att mäta och teorin bygger på att det finns ett samband mellan interna och externa attribut. Man antar exempelvis att strukturerad kod (internt attribut) ger en produkt som är lättare att underhålla (maintainability, externt attribut).



Ovanstående resonemang ger oss följande:

- Interna attribut hos produkten är de attribut som kan mätas direkt på själva produkten, vilket ger att interna attribut kan mätas genom att undersöka produkten separerad från sitt beteende och sin miljö.
- Externa attribut hos produkten är attribut som endast kan mätas tillsammans med hur produkten verkar i sin miljö. Här är alltså själva produktens beteende i en viss miljö av intresse.



Figur 3:3 Exempel på interna och externa produktattribut som påverkar upplevd kvalitet.

Vi kan se i figur 3:3 att interna attribut som design och funktioner påverkar produktens kvalitet men kvaliteten påverkas även av användarens kunskaper och den tekniska miljö som produkten verkar i. De interna attributen kan vi värdera genom att exempelvis studera dokument som beskriver designen och kodutskriften. Sådant kan värderas utan att produkten behöver "köras". Externa attribut å andra sidan påverkas av yttre omständigheter som användarens kunskaper och omgivande miljö. Omständigheter som är svårare att förutsäga.

3.3 Vald modell

Det centrala i kvalitetsmodellen är högnivåattributen, vanligtvis kallade kvalitetsfaktorer. Som vi sett tidigare presenterar olika forskare här en rad olika kvalitetsfaktorer men för att en modell ska kunna ha ett visst mått av genomslagskraft krävs dels att det finns en viss enighet kring vilken modell man ska använda och dels att modellen har en chans att spridas runt om i världen. Vid mina studier i ämnet har jag funnit att ISO/IEC har antagit en standard för bedömning av mjukvaruproduktkvalitet.



Med tanke på det renommé ISO/IEC har i övriga branscher ansåg jag därför att valet av ISO/IEC's kvalitetsmodell var att föredra framför en enskild forskares. Det finns även flera exempel i litteraturen på forskningsprojekt som resonerat på samma sätt och där man utgått från denna ISO/IEC standard. Exempelvis har Asnaghi et al. (1996) undersökt möjligheterna att bedöma och certifiera mjukvaruprodukter efter ISO/IEC 9126. Ett annat forskningsprojekt, Welzel och Hausen (1996), har utvecklat en metod för mjukvarubedömning efter samma standard. I Sverige har Svensk Programvaruindustri utgått från ISO/IEC 9126 för att kvalitetsmärka mjukvara med deras Spi2000 stämpel (Hedlund, 2000). Men det kanske tyngsta skälet som talar för ISO/IEC är att **en** universell modell dramatiskt förenklar jämförelsen av olika produkter.

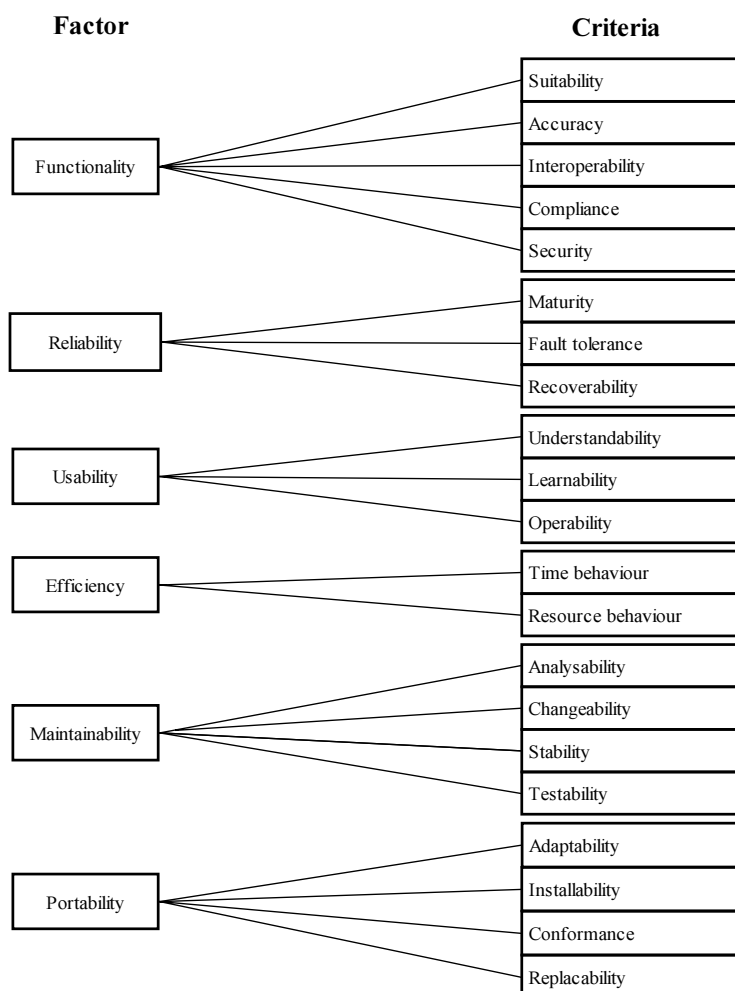
3.3.1 ISO/IEC 9126, en modell för utvärdering av mjukvarukvalitet

Jag kommer nedan att beskriva den kvalitetsmodell som heter ISO/IEC 9126, Software Product Evaluation: Quality Characteristics and Guidelines for their Use. Materialet nedan bygger på information hämtad från EAGLES (1995).

De kvalitetsfaktorer som utgör grunden i modellen är följande:

- functionality
- reliability
- efficiency
- usability
- maintainability
- portability

Tanken är att de sex faktorerna ska vara heltäckande, att alla typer av mjukvarukvalitet ska kunna beskrivas i termer av en eller flera av de sex kvalitetsfaktorerna. Jag avser att nedan närmare förklara innebörden av varje faktor och även förklara de underattribut som karaktäriserar varje kvalitetsfaktor. ISO/IEC 9126 är uppbyggd efter samma konceptuella mönster som de ovan beskrivna kvalitetsmodellerna där varje kvalitetsfaktor karaktäriseras av ett antal underattribut som i sin tur på olika sätt ska kunna mätas. En skillnad är dock att underattributen i ISO/IEC 9126 inte i något fall är relaterade till mer än en kvalitetsfaktor (se figur 3:4).



Figur 3:4 ISO/IEC 9126, kvalitetsfaktorer och underattribut.

3.3.2 Functionality

"A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs." (EAGLES 1995)

Mjukvarans functionality är kanske ett av de kvalitetsattribut man först kommer att tänka på. Functionality innebär att mjukvaran ska innehålla alla de funktioner som krävs för att mjukvaran ska kunna utföra sina uppgifter på ett tillfredsställande sätt. De tydligaste exemplen på funktioner är extern input och extern output där extern input kan vara filnamn och menyval och extern output kan vara rapporter eller meddelanden som mjukvaran genererar. För att kunna bedöma kvaliteten på denna functionality måste man dock ytterligare belysa begreppet.

Här listar ISO/IEC fem underattribut som ska möjliggöra en bedömning av functionality.



- Suitability, som avser de befintliga funktionernas lämplighet för den specifika uppgift de ska utföra.
- Accuracy, som avser förmågan att ge riktiga eller överenskomna resultat eller effekter.
- Interoperability, som avser förmågan att interagera med specificerade system.
- Compliance, som avser på det sätt mjukvaran följer applikationsrelaterade standarder, konventioner eller juridiska regleringar och liknande föreskrifter.
- Security, som avser förmågan att förebygga otillåten access, vare sig den är medveten eller omedveten, till program och data.

3.3.3 Reliability

"A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time." (EAGLES 1995)

Reliability är en av de faktorer som det forskats mest kring. Detta beror troligtvis på att mjukvarans reliability, eller snarare frånvaron av reliability, i vissa sammanhang kan få ödesdigra konsekvenser. Med reliability avser man hur tillförlitligt mjukvaran är. Fel i mjukvaran kan under vissa omständigheter orsaka systemkrascher som beroende på vad systemet används till kan ge upphov till olika allvarliga händelser. I olika system kan mjukvarufel ge upphov till ekonomiska förluster för företaget eller enskilda personer. I andra sammanhang kan människor dö på grund av mjukvarufel i medicinsk utrustning eller kärnkraftverk.

Reliability dekomponeras enligt följande:

- Maturity, som avser frekvensen av avbrott (failure) förorsakade av fel i mjukvaran.
- Fault tolerance, som avser förmågan att upprätthålla en viss utförandenivå vid uppkomst av fel eller ett felaktigt användande.
- Recoverability, som avser kapaciteten att återta sin utförandenivå och att återfå den data som direkt var påverkad vid avbrottet och den tid och kraft som behövdes för detta.

Värt att notera är att mjukvara inte påverkas av slitage eller åldrande. Begränsad reliability är ett resultat av felaktig kravspecifikation, design eller implementation. Systemkrascher på grund av denna typ av fel beror snarare av det sätt på vilket mjukvaruprodukten används än av hur länge produkten använts.

3.3.4 Usability

"A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users." (EAGLES 1995)

Fenton och Pfleeger (1996) menar att det som utvecklare kan vara lätt att glömma usability till förmån för functionality men usability är självklart mycket viktigt då



systemets framgång hos användaren till stor del beror på graden av usability. En allmän uppfattning är att man kan beteckna usability som systemets *användarvänlighet*. Systemet ska vara lättförståeligt och intuitivt, ett exempel kan vara att ett ordbehandlingsprogram bör påminna om andra ordbehandlingsprogram för att användaren konceptuellt ska kunna förstå uppbyggnaden av det nya systemet. Befintligheten av hjälpfunktioner främjar också graden av usability.

I ISO/IEC 9126 delas usability upp i följande underattribut:

- Understandability, som avser användarens prestation för att känna igen den logiska strukturen och dess tillämpningsbarhet.
- Learnability, som avser användarens ansträngning för att lära sig dess tillämpning.
- Operability, som avser användarens ansträngning för hantering och operationskontroll.

Användare betyder oftast användare av interaktiv mjukvara. Här innefattar begreppet exempelvis operatörer, slutanvändare och indirekta användare som är beroende av mjukvaran.

3.3.5 Efficiency

"A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions." (EAGLES 1995)

Hur effektiv en mjukvaruprodukt är märks för användaren oftast i hur snabbt produkten arbetar. Krav på snabbhet har man främst i realtidssystem men även i vanliga applikationer skapar långsam databearbetning förseningar och man upplever kvaliteten som dålig.

Efficiency dekomponeras enligt följande:

- Time behaviour, som avser svars- och bearbetningstider och med vilken hastighet funktioner kan utföras.
- Resource behaviour, som avser den mängd resurser som används och tiden för utnyttjandet vid utförandet av funktionen.

De resurser man talar om inkluderar annan mjukvara, hårdvara, material (printer papper och disketter) och personal för service och underhåll.

3.3.6 Maintainability

"A set of attributes that bear on the effort needed to make specified modifications." (EAGLES 1995)

System är sällan en helt statisk produkt utan man kan snarare utgå från att förändringar i mjukvaran kommer att göras även efter systemleverans.



Denna typ av förändringar eller systemunderhåll kan enligt Sommerville (1995) delas in i tre grupper:

- Korrigerande underhåll som innebär att man försöker finna och därefter rätta till de eventuella fel som uppstått.
- Adaptivt underhåll som innebär att man på grund av förändringar i omgivningen måste anpassa mjukvaran till de nya förutsättningarna.
- Förbättrande underhåll innebär att man lägger till nya funktioner eller försöker klargöra situationer som kan missuppfattas.

Fenton och Pfleeger (1996) nämner ytterligare en typ av underhåll:

- Preventivt underhåll som innebär att den som sköter underhållet rättar till fel som han upptäcker innan de har orsakat användaren något problem.

För att underlätta denna typ av arbete krävs det att mjukvaran lever upp till ett visst mått av maintainability.

Maintainability delas in i följande underattribut:

- Analysability, som avser den ansträngning som krävs för att diagnosticera brister eller felkällor, eller för identifiering av delar som måste förändras.
- Changeability, som avser den ansträngning som krävs för modifiering, fel borttagning eller anpassning till förändringar i omgivningen.
- Stability, som avser risken att andra delar oförväntat påverkas av modifieringar.
- Testability, som avser den ansträngning som krävs för att validera modifierad mjukvara.

Modifieringar kan behöva göras till följd av förändringar i omgivningen, förändrade krav och förändrade funktionskrav.

3.3.7 Portability

"A set of attributes that bear on the ability of software to be transferred from one environment to another." (EAGLES 1995)

Den sista av kvalitetsfaktorerna är portability som ska ge en indikation på hur flyttbar mjukvaran är. Ett exempel är det man brukar kalla plattformsoberoende där mjukvaran ska fungera på flera olika operativsystem. Begreppet är vidare än så och här menar man att det ska vara flyttbart till olika miljöer. Med miljö avser man olika organisationer, hårdvara eller mjukvara.

Följande underattribut har man identifierat:

- Adaptability, som avser möjligheten att anpassa mjukvaran till en specifik miljö utan andra hjälpmedel än de som tillhandahålls för detta syfte för den avsedda mjukvaran.
- Installability, som avser den ansträngning som krävs för att installera mjukvaran i en specifik miljö.
- Conformance, som avser hur väl mjukvaran följer de standarder och konventioner som avser portability.



- Replaceability, som avser möjligheten och prestationen för att använda mjukvaran i stället för annan specificerad mjukvara i den miljön.

3.3.8 Utvärderingsnivåer

Dahlbom och Mathiassen (1993) invänder att vissa av kvalitetsfaktorerna kan tänkas motverka varandra. Till exempel kan höga krav på usability få konsekvensen att man lägger mycket energi på att ge produkten tydliga grafiska gränssnitt vilket i sin tur kan påverka produktens efficiency genom att systemet blir långsammare. Tanken är dock inte att optimera varje enskild kvalitetsfaktor utan kvalitetsnivån på varje enskild faktor anges av det enskilda projektets karaktär. För att värdera varje kvalitetsfaktor finns en modell i ISO/IEC 9126 för hur man kan utvärdera de enskilda kvalitetsfaktorerna, se tabell 3:2.

Tabell 3:2 ISO/IEC 9126-6, 1993. Riktlinjer för val av utvärderingsnivå och exempel på applikationer (Asnaghi et al. 1996 s. 220)

Level	Environment	Person	Economic	Security	Ex. of application
D	Small damage to property	No risk to people	Negligible economic loss	No specific risk	Entertainment, household
C	Damage to property	Few people disabled	Significant economic loss	Protection against error risk	Fire alarm, process control
B	Recoverable environmental damage	Threat to human lives	Large economic loss	Protection of critical data and services	Medical system, financial system
A	Unrecoverable environmental damage	Many people killed	Financial disaster	Protection of strategic data and services	Railway systems, nuclear systems

Varje kvalitetsfaktor kan utvärderas oberoende av varandra med hänsyn till de fyra aspekterna och de fyra nivåerna i tabell 3:2, där nivå D är lägsta nivån och A högsta. Utefter hur man bedömer en faktor följer att man använder olika grundliga tekniker för att utvärdera varje faktor. Nivån i tabell 3:2 påverkar också vilket värde man anser att varje mått man mäter upp bör nå upp till. Exempelvis kan det räcka med manuell kodgranskning för ett system i nivå D men för ett system i nivå A kan det krävas 100%-ig testning i emulerad miljö där man försöker framkalla krissituationer för att se hur systemet reagerar.



4 Resultat

4.1 Litteraturstudie

Resultatet av litteraturstudien är att jag fann kvalitetsmodellen ISO/IEC 9126 som presenterades i Teorikapitlet under rubriken Vald modell. Modellen grundar sig på SQM-teorin som förespråkar metoden att mäta attribut hos mjukvara för bedömning av kvaliteten.

4.2 Intervju

Under denna rubrik avser jag att redovisa resultaten av frågorna i mina intervjuer. Då underlaget är för litet för att jag ska kunna dra statistiska slutsatser kommer jag att redovisa huvuddragen och representativa åsikter i de svar jag fått och inte kvantifierad data i form av diagram och procentsatser. Jag har valt att redovisa frågorna efter de olika huvudteman som jag delat upp frågorna i, vilka är följande:

- Person- och företagsinformation
- Strategiska mål för kvalitetsarbetet
- Kvalitet på projektnivå
- Produktkvalitet (kvalitetsmodellen)

4.2.1 Person- och företagsinformation

Första avsnittet avser att ge en bild av de personer och företag som jag intervjuat.

Intervjupersonens huvudsakliga arbetsuppgifter och kvalitetsansvar.

Person 1: Kvalitetsansvarig för södra Sverige. Arbetar främst med modeller för kvalitet i projektarbete och företagets Quality Assurance Plan.

Person 2: Arbetar som Corporate Account Representative. Företaget skräddarsyr och implementerar Microsoftlösningar och erbjuder även hjälp inför ISO-certifieringar av företag. Certifieringar av IS/IT-organisationers rutiner, struktur och dokumentation.

Person 3: Marknadsansvarig för Industrisystem och även personalansvarig. Arbetar med kvalitetsfrågor på så sätt att det är en utbildningsfråga.

Person 4: Ansvarig för försäljning, personal och uppdrag för en systemutvecklingsgrupp. Ansvarar för kvalitetsarbetet i systemutvecklingsgruppens uppdrag.

Person 5: Regionchef för systemservice. Som regionansvarig även ansvarig för kvaliteten. Ansvarar för att företaget uppfyller ISO-regler och rutiner.



Person 6: Chef för avdelningen Test och Metodik. Ansvarig för att utveckla kompetensen inom test och verifiering samt testmetodik som är en del av företagets kvalitetssäkring.

Person 7: Regionansvarig. Ansvarig för att avdelningen producerar resultat med hög kvalitet.

Person 8: Arbetar med verksamhetsutveckling och strategier för hur företag ska kunna förbättra sina programvaruprodukter. Tittar på utvecklingsmodeller för hur man tar fram programvara (produktnära), projektstyrningsmodeller och affärsmodeller.

Gemensamt för de intervjuade är att samtliga har ett visst kvalitetsansvar, antingen för kvaliteten på produkten som levereras eller för att utveckla kvalitetsarbetet i organisationen där de verkar.

Arbetar ditt företag ute hos kund och/eller med egna projekt?

Inom IT-branschen är det vanligt att arbeta både hos kund och med egna utvecklingsprojekt, en faktor som kanske påverkar kvalitetsarbetet. Bland företagen i undersökningen arbetar alla på ett eller annat sätt ute hos kund eller på uppdrag av en kund. Sätten att arbeta ute hos kund skilde sig åt dels mellan de olika företagen och dels mellan olika projekt inom samma företag.

De olika arbetssätten var huvudsakligen:

- Att skraddarsy en färdig produkt ute hos kund.
- Att ta hem ett uppdrag för utveckling för att sedan implementera lösningen.
- Att som konsult agera som resurs- eller kompetensförstärkning på plats hos kund.

De flesta företagen hade även så kallade inhouse projekt där man utvecklar egna produkter men hos dessa företag var ändå huvuddelen av det totala antalet projekt på uppdrag.

Lyder ni då under kundens kvalitetsmetod eller arbetar ni efter er egen metod?

Då man arbetar så mycket ute hos kund kan man fråga sig vems kvalitetsmetod som styr arbetet. På denna fråga fanns inget entydigt svar. Det man kan säga är att i de fall där man arbetat som konsult i ett projekt på plats hos kunden har man i de fall där kunden krävt det arbetat efter kundens kvalitetsmetod. I vissa fall har man dock vid denna typ av projekt då kunden inte haft en egen kvalitetsmetod använt sin egen metod. Några av företagen använde nästan alltid sin metod även ute hos kund. I de fall där man tar hem projektet anammar man sin egen kvalitetsmetod.

Är skillnaden mellan er och kundens metod stor?

Inte heller på denna fråga fanns ett generellt ja eller nej. Några tyckte att skillnaderna inte var så stora mellan kundernas kvalitetsmetod och deras egna. De flesta tyckte att det var olika från kund till kund.



Några kommentarer var:

- *"Det beror på hur IT-mogen organisationen är."*
- *"Stora kunder inom industrin är kunniga och vet vad de vill ha (även i kvalitet)."*
- *"Kunder som själva satsar på kvalitet förstår behovet av kvalitet."*
- *"Kunder som inte själva har IT som kärnverksamhet vet inte hur man kvalitetssäkrar en IT-tjänst."*

Vad är normalstorleken på era projekt? (tid, personalåtgång)

För att få en uppfattning om omfattningen på vanliga projekt och därmed också kvalitetsarbetets omfattning tillfrågades företagen om hur stora projekt de normalt hade. Normalprojektet skilde sig något mellan företagen vilket har ett visst samband med storleken på företaget. Normalprojektet var mellan 6 till 12 månader och 3 till 5 man och i extremfallen från några veckor till 24 månader och 15 man.

4.2.2 Strategiska mål för kvalitetsarbetet

Andra avsnittets frågor gäller övergripande frågeställningar kring företagets syfte och mål med kvalitetsarbetet.

Vad är företagets syn på kvalitet?

Jag ville få en uppfattning om hur man resonerar kring kvalitet och hur företagets officiella syn på kvalitet var. De flesta angav att man ansåg att kvaliteten var väldigt viktig, ingen ansåg att det var oviktigt. Man var överrens om att ett rykte som stod för kvalitet marknadsföringsmässigt talar för sig själv.

Några kommentarer var:

- *"Man ska uppfattas på marknaden som ett seriöst företag med hög kvalitet på en levererad tjänst."*
- *Kvalitet ska genomsyra hela verksamheten.*
- *"Det krävs kvalitetsrutiner på alla nivåer för att produkten ska bli kvalitetssäkrad."*
- *"Det är viktigt att utbilda konsulterna."*
- *"Vi har en kvalitetschef." (!)*

Vad har ni för övergripande mål med kvalitetsarbetet?

Inget företag idag skulle förmodligen påstå något annat än att de satsar på kvalitet men gör man en medveten satsning? En övervägande del av de tillfrågade svarade i termer av *nöjd kund*, *noll* eller *få fel, i tid* och *inom kostnadsramarna*. Av dessa var det *nöjd kund* och *nollfel* som flest nämnde.

Andra mål var:

- *att slå kundens förväntningar.*
- *att bli certifierade.*
- *att vara en bra arbetsplats.*
- *att lösningen ska ge (positiva) effekter hos kundens verksamhet.*

Varför är det viktigt? Konkurrensmedel, kundkrav, etc



Alla är överrens om att det är viktigt att satsa på kvalitet, men varför? Och har det vuxit fram som ett internt eller externt krav? Nästan alla ansåg att det främst är ett konkurrensmedel, även om några som svarade konkurrensmedel tyckte att kunderna ställer krav och till och med i vissa fall kräver ISO-certifiering.

Några åsikter var:

- *"Drar man igång kvalitetsarbetet efter kundkrav så har man inte förstått varför man ska satsa på kvalitet."*
- *"Vi är ingen drake, därför måste vi kunna visa att vi lever upp till industrins hårda krav på kvalitet."*
- *"Levererar vi kvalitet från början så slipper vi komma tillbaks och rätta till fel, vilket är effektivare för både oss och kunden."*

En svarade att det vuxit fram som ett krav internt i och med att systemen man levererar blivit större samtidigt som organisationen vuxit så behövde man ett kvalitetssystem i ryggen.

Kan det vara så att kunden inte är beredd att betala för den eventuella mertid och merkostnad som ett rigoröst kvalitetsarbete innebär?

Syftet med frågan var att försöka få en uppfattning om kvalitetsarbetet i IT-branschen hållits tillbaka av kunderna snarare än av leverantörerna. Många av de tillfrågade tyckte att det kunde vara så, men man sa ofta samtidigt att det är olika från kund till kund. Det fanns en viss tendens mot att större kunder som själva satsar på kvalitet har bättre förståelse. Okunniga kunder ansågs svårare att arbeta med.

Några kommentarer var:

- *"Det är till och med så att vi har tackat nej till jobb för att vi anser att kunden inte klarar av att leva upp till våra krav."*
- *"Kvalitet är en initial kostnad, gör du inte rätt från början så kostar det ännu mer men det syns inte inom projektets ramar utan det kommer som en förvaltningskostnad senare."*
- *"Det är inte alltid kunder förstår varför vi är dyrare än källarfirman."*

Ett par av de tillfrågade ansåg att det inte var så problematiskt då man förklarar i offerten vad man tar betalt för och sedan är det upp till kunden att bestämma sig.

Några kommentarer på tidsaspekten var:

- *"Man har ett slutdatum men förhandlingarna kan dra ut på tiden så startdatumet skjuts fram men ej slutdatumet."*
- *"Det är viktigt att de som sköter affärsprocessen förstår utvecklingsprocessen så att man inte lovar leverans på halva tiden. Det måste finnas en helhetssyn på kvalitet."*



4.2.3 Kvalitet på projektnivå

Inriktningen på frågorna är nu mer praktiska. Respondenterna har beskrivit hur de arbetar efter respektive systemutvecklingsmodell och hur kvalitetsaspekten påverkar arbetet i ett projekt.

Inför ett nytt projekt kan du beskriva hur man planerar kvalitetsarbetet.
Samtliga företag hade välgenomtänkta metoder för projektarbete.

De huvudsakliga hållpunkterna var följande:

- Förstudie.
- Ta fram kravspecifikation.
- Grov projekt- och tidsplan.
- Design.
- Utveckling.
- Systemleverans.

Integrerat med systemutvecklingen pågår kvalitetsarbetet. Kvalitetsarbetet kännetecknas av att man arbetar strukturerat efter modeller. Under projektets fortgång har man kontinuerligt checklistor som bockas av vid granskningar. Man har kodgenomgångar och man försöker i allmänhet testa systemdelar i den mån det går redan tidigt i projektet. Man dokumenterar det man gör för att bland annat öka spårbarheten. Vissa milstolpar i projektplanen granskas av kunden och den egna ledningen inom företaget.

Följer ni en utstakad metod?

På denna fråga svarade alla ja. Vilket tyder på att alla företagen har gjort en medveten kvalitetssatsning.

Har ni utvecklat er egen kvalitetsmetod?

Alla sade sig använda inom företaget egenutvecklade kvalitetsmetoder men flera påpekade att metoderna byggde på samlade erfarenheter och intryck från tidigare anställningar. Ett av företagen hade köpt en kommersiell produkt specifikt för systemutvecklingsprocessen.

Ser kvalitetsarbetet annorlunda ut vid olika typer av projekt? Blir olika faktorer/begrepp olika viktiga?

De flesta ansåg att man fick anpassa kvalitetsarbetet efter projektets karaktär. Att det var viktigt att man i kravspecificeringsfasen lyckas ta reda på kundens krav och behov så att detta styrde hur kvalitetsarbetet utformades.

Några kommentarer var:

- "Vår modell är generisk och den kan vi applicera på vilket projekt som helst."
- "Faserna i ett utvecklingsprojekt är alltid samma men dokumenten i respektive fas är olika beroende på projektets art."



- "Några basdokument behandlas i alla projekt, de pekar sen i sin tur på specifika dokument som tas fram för dom enskilda projekten."
- "Projektmålen visar vart tyngdpunkten i kvalitetsarbetet hamnar."

Ett av de tillfrågade företagen sålde en egenutvecklad produkt och för deras del såg projekten ganska lika ut, även hårdvarumiljön var alltid samma i deras fall.

Vilka olika ansvarsområden har man i kvalitetsarbetet? Vilka utför vad?

De olika intervjupersonerna svarade ganska överensstämmande på denna fråga. Man ansåg att den enskilde projektdeltagaren var ansvarig för sina arbetsuppgifter och därmed också att han eller hon var ansvarig för att man uppfyller de kvalitetskrav som den uppgiften kräver. Men i och med att projektledaren har ett övergripande projektansvar så får han eller hon en samordnande roll i kvalitetsfrågor. En åsikt som återkom var att man som projektdeltagare inte bara kan göra sin del och därefter lämna över till andra utan det är viktigt att man ser ett större sammanhang och hjälps åt. En av intervjupersonerna sa så här om vikten av att enskilda medarbetare förstår vikten av kvalitet, "Kvalitet ska komma underifrån. När kravet på oss att öka kvaliteten kommer underifrån då har man kommit långt".

4.2.4 Produktkvalitet (kvalitetsmodellen)

Resterande frågor har koncentrerats på hur man praktiskt testar kvaliteten på produkterna. Figuren (se appendix A.1.2) har utgjort ett viktigt underlag för diskussionen kring kvalitetsfaktorerna.

När under ett projektarbets fortlöpande testar/kontrollerar man olika kvalitetsfaktorer?

En klart övervägande del av svaren var att man testar fortlöpande under utvecklingsarbetet. Av några av svaren framgick att det finns en handfull olika test nivåer.

Följande huvudnivåer nämndes:

- Komponenttester. Man testar enskilda komponenter, vanligen programkod, bugtestning.
- Funktions- och integrationstester. Man testar funktioner i systemdelarna och man testar om olika systemdelar fungerar tillsammans och att hård- och mjukvara fungerar.
- Systemtest. Går i huvudsak ut på att testa enligt kravspecifikationen.
- Acceptanstest. Uppdragsgivaren testar systemet vilket leder till acceptans eller ej.

Några av de intervjuade nämnde också att man vanligtvis testar först i simulerad miljö innan man testar i den verkliga miljön.

Vilka begrepp anser ni är viktigast när man mäter kvalitet?

I samtliga fall är det vid fråga 16 som jag tagit fram figuren över kvalitetsfaktorerna. Diskussionen har då rört sig kring de begrepp som finns i figuren (se appendix



A.1.2). Generellt sett tyckte alla att alla faktorer var viktiga att ha i åtanke, även de som inte sett en liknade figur tidigare. De flesta ansåg också att det var svårt att generellt säga att någon speciell faktor är viktigare än någon annan.

Tre stycken av de intervjuade angav dock några faktorer som de ansåg var viktigare. En av dessa svarade functionality och reliability, en andra reliability och en tredje functionality. De funktioner som kunden själv har identifierat som sitt behov när det beställer ett system är ofta de funktioner som utgör systemets functionality och därmed är det ganska naturligt att man i så hög grad som möjligt uppfyller dessa krav. En av de intervjuade ansåg att industrin kräver 100%-ig reliability och därmed var det viktigast.

Några kommentarer var:

- *"För ett industrisystem kan man säga att reliability är skitviktigt!"*
- *"Det som kunden oftast formulerat i kravspecen är funktionaliteten."*

Några av de intervjuade tyckte att portability var det som var minst viktigt. Anledningen till detta var lite olika men en av de tillfrågade angav att man alltid installerade sin standardprodukt i IBM AS 400 miljö. Två andra av de intervjuade arbetade mot industrin och ansåg att varje projekt var så specialiserat att portability inte var speciellt angeläget.

Man kommenterade det så här:

- *"Att det (systemet) går att flytta till en annan plattform är mer en vinst för oss än för kunden."*
- *"Mjukvara skrivet för ett inbäddat system är inte portabelt."*

En genomgång av de övriga kvalitetsfaktorerna gav följande resultat. Usability arbetar man med på det viset att man har regler för gränssnittsutformning, ett företag hade klara regler för till exempel färgval. Man samarbetar här även med användarna. Efficiency arbetar man på i de fall där det anses viktigt och av svaren att döma är detta inte något man konsekvent undersöker.

En åsikt i sammanhanget var:

- *"I vissa sammanhang vet vi att vi har ett väl tilltaget standardminnesutrymme och då behöver vi inte göra minneseffektiv kod, då är det helt onödigt att lägga tid på att optimera koden."*

Maintainability arbetar de flesta av de tillfrågade med. Man har interna regler för hur koden ska struktureras, man förespråkar lättförståelig kod med talande variabelnamn framför avancerade optimala lösningar, eller som en person uttryckte det; programmering!

En av de intervjuade sammanfattade resonemanget med att förklara att kunden oftast ser sitt behov i den funktionalitet som man anger i kravspecifikationen. Som systemutvecklare är det då viktigt att i förstudien hitta rätt kravställare inom varje



område som kvalitetsfaktorerna tar upp. Till exempel, om kunden har en IT-avdelning som ska stå för systemunderhåll så är det bland dem som man ska hitta kravställare på maintainability.

Hur går ni till väga för att mäta kvalitet?

Det var få av de intervjuade som kunde ge konkreta exempel på denna fråga men jag redovisar de exempel som gavs. En av de intervjuade förklarade att de mätte en faktor som de kallade tillgänglighet (motsvarande reliability i figur 3:4). Detta mättes i antal minuter som systemet stod still under en månad. Gränsvärdet var 99.5% vilket motsvarar mindre än 10 min/dygn. Samma företag mätte också sin produktkvalitet i hur mycket garantiarbete man hade. Deras mål var 2% och under de senaste 4 åren hade man legat på 1.6%.

Ett exempel på usability som gavs var att man kan mäta antal knapptryckningar eller tabbar som krävs för att utföra en arbetsuppgift. Ett företag hade precis börjat undersöka och sätta mått på själva kvalitetsmetoden för att få siffror på till exempel hur många procent av det totala antalet fel som kvalitetstestet hittar.

Några kommentarer var:

- *"Kunden är inte så intresserad av uppfyllande av mätdata, han tittar mer på functionalitydelen därför har vi inte så mycket mätdata."*
- *"Specifika mätdata tar vi bara fram på vissa av faktorerna/kriterierna."*

Vilka typer av tester använder man?

De flesta förklarade att man testar systemet mot kravspecifikationen. Ett exempel var att man satte upp en krav- och verifieringsmatris där det för varje krav också finns ett testfall.

Konkreta exempel på test var:

- Bugtest. Man bugtestar koden företrädesvis tidigt i de mindre delkomponenter som utgör systemet.
- Flödestest. Man följer en process från start och ända fram till mål. Det totala antalet vägar genom systemet kan vara nära oändligt stort men man försöker i allmänhet testa minst 80 - 90 %.
- Kommunikationstester mot omgivande miljö. Man testar att datakommunikationen fungerar.
- Användartest. Man bjuder in användarna som får testa gränssnittet.

4.3 Modell vs. verklighet

Resultatet av jämförelsen mellan modell och verklighet förs i diskussionsform i nästa kapitel.



5 Diskussion

Min studie av systemutvecklingsföretag visar att kvalitetsmedvetenheten är väldigt hög. Som företag vill man förknippas med god kvalitet, och ett väl genomfört uppdrag ger nöjda kunder och ett gott rykte. Detta skapar i sin tur ett marknadsföringsmässigt värde som ligger till grund för tillströmningen av nya uppdrag och kunder.

5.1 Diskussion kring delsyfte 1 till 3

Jag avser nedan att föra en diskussion kring de delsyften jag haft i min studie. De tre delsyftena var följande:

1. Att med hjälp av en förstudie av litteratur och forskning kring mjukvaruproduktkvalitet finna en väletablerad teoretisk modell för bedömning mjukvaruprodukter.
2. Att mot bakgrund av förstudien kartlägga och beskriva hur systemutvecklingsföretag arbetar med mjukvaruproduktkvalitet inom systemutveckling.
3. Att jämföra den av litteraturstudien funna kvalitetsmodellen med hur systemutvecklingsföretagen arbetar med produktkvalitet.

5.1.1 Delsyfte 1

I delsyfte 1 har jag haft som mål att finna en väletablerad modell för bedömning av mjukvaruprodukter. Min utgångspunkt har således inte ifrågasatt om man överhuvudtaget har ett behov av en kvalitetsstrategi. Jag avser därför att nedan diskutera något av den kritik som huvudsakligen har framförts mot kvalitetssystem i allmänhet.

Det har i debatten framförts åsikten att kvalitetsarbetet blir en belastning. Dels som en administrativ belastning som tar tid och energi från kärnverksamheten och dels som en kreativitetshämmande belastning som begränsar programmerarens kreativitet. Den senare åsikten diskuteras bland andra Dahlbom och Mathiassen (1993). De ställer den retoriska frågan: ”I vilken utsträckning var dessa klassiska mästerverk (...musik, konst, fiktion...) resultatet av en externt kontrollerad och en tydligt ledd process?” Dahlbom & Mathiassen (1993, s. 163). Frågan är intressant och befogad men jag anser att det föreligger en tydlig skillnad mellan kvalitetskraven på ett konstverk och en kommersiell produkt. Skillnaden ligger i att en konstnär inte har några formella externa krav på sig att varje målning, etc. ska vara ett mästerverk, däremot förväntar sig köparen/beställaren av ett system varje gång ett ”mästerverk”. Enligt min mening kan det mycket väl stämma att kvalitetssystemet blir en kreativitetshämmande faktor som därmed kan förhindra x antal kreativa mästerverk från att skapas men kraven att alla system som levereras ska hålla en jämn och hög kvalitet talar för behovet av kvalitetssäkring. Man har helt



enkelt inte råd att som systemleverantör erbjuda kunden ett eventuellt mästerverk utan man måste kunna garantera kunden hög kvalitet varje gång, men kanske då på bekostnad av kreativa mästerverk.

Det har även förts fram en del kritik mot ISO-certifieringar. Inom IT-sektorn har man bland annat hävdad att ISO är allt för inriktad mot traditionell tillverkningsindustri och därför har andra kvalitetsstämplar (exempelvis TickIT) speciella för mjukvaruutveckling haft ett visst genomslag. Annan kritik som framförts mot ISO är att en ISO-stämpel inte garanterar bättre kvalitet. ”-Det vi garanterar är att vi gör på samma sätt varje gång. ISO är ingen garanti för att något är bra, utan för att man gör som man bestämt sig för att göra.” Dahlgren (2000, s. B01). Denna kritik riktar sig emot att när man ISO 9000-certifieras så förbinder man sig att alltid arbeta enligt de arbetsprocesser som man dokumenterat, vilket enligt kritikerna inte garanterar en kvalitetsprodukt. Samma kritiker menar att ISO-stämpel har större betydelse för de certifierade företagen som en marknadsföringsåtgärd än en kvalitetsgaranti. Något som till viss del bekräftas av svaren i min undersökning där flera av de intervjuade ansåg att en kvalitetsstämpel var ett viktigt konkurrensmedel.

Trots kritiken har jag valt att studera ISO/IEC 9126 närmare då jag efter att ha studerat ämnet mjukvarukvalitet är av åsikten att vi har mycket att vinna på att försöka mäta kvalitet. Ovan beskrivs hur kritiker anser att en ISO-certifiering endast garanterar att vi arbetar enligt en dokumenterad process som inte garanterar kvaliteten på produkten. Jag delar denna uppfattning men anser att ISO/IEC 9126 är ett steg i rätt riktning då den syftar till att skapa ett underlag för just bedömningen av själva produkten. Jag anser vidare att kvalitetsmodeller som till exempel ISO/IEC 9126 hjälper till att konkretisera problematiken kring att definiera begreppet kvalitet. Att försöka definiera kvalitet med hjälp av begrepp som maintainability och functionality tvingar oss att analysera olika sidor av kvalitet och även om ingen kvalitetsmodell kan sägas vara fulländad ger den ändå ett underlag för en kritisk granskning av produkten.

5.1.2 Delsyfte 2

Jag har i uppsatsen haft som delsyfte att mot bakgrund av förstudien kartlägga och beskriva hur systemutvecklingsföretag arbetar med mjukvaruproduktkvalitet inom systemutveckling. Resultatet tyder på att företagen till största delen lutar på att processen ger produkten den kvalitet man eftersträvar. Ingen av de intervjuade företagen mätte kvalitet efter den teori som beskrivits tidigare i uppsatsen. Begreppen i kvalitetsmodellen var dock inte obekanta utan man beaktade i olika utsträckning begreppen i utvecklingsarbetet dock utan att mäta. Hur arbetar man då med produktkvalitet?

Svaren i undersökningen visar att samtliga företag följer metoder som ett led i att uppnå högre kvalitet på sina produkter. De metoder man följer är systemutvecklingsmetoder och projektstyrningsmodeller. Mycket kortfattat kan man säga att systemutvecklingsmetoderna stödjer de aktiviteter som utvecklingsarbetet består av och projektstyrningsmodellerna beaktar problem förknippade med



exempelvis behovet av specialkompetenser och personal för projektet. Om vi översätter detta till termer från teoriavsnittet kan vi se att på det sätt man arbetar idag så har man metoder och modeller som stödjer kvalitetsarbetet främst med avseende på de processer och resurser som ingår i ett systemutvecklingsprojekt.

Man kan naturligtvis invända att i systemutvecklingsmetoden ingår moment då man testar produkten men detta sker mot kraven i kravspecifikationen vilket inte alltid är det samma som det kvalitetsmodellen tar upp. Exempelvis kan en produkt testas i enlighet med den funktionalitet (functionality) och de krav på snabba svarstider (efficiency) som en kund angett men produkten kan visa sig svår att utföra systemunderhåll (maintainability) på. Något som kunden kanske inte upptäcker inom ramen för projektet utan först flera år senare då kunden behöver modifiera systemet. Man kan då ifrågasätta kvaliteten på produkten trots att den motsvarade kundens ursprungliga krav. I ISO's definition på mjukvarukvalitet talar man också om "implied needs" (underförstådda behov) vilket innebär att alla kundens behov inte nödvändigtvis är klart uttryckta av kunden. Kvalitet kan alltså vara mer än att uppfylla kundens krav. Dahlbom och Mathiassen (1993) går på samma linje och menar att för att uppnå kvalitet räcker det inte att uppfylla kundkrav utan man måste även använda sin professionalitet för att nå längre än kundens förväntningar.

Flera av de intervjuade sade att man mäter det kunden tycker är viktigt samtidigt som flera uppgav i andra frågor att kunderna i många fall var okunniga om IT och dåliga på att ställa krav. Jag anser att resonemanget haltar betänkligt eftersom svaren från intervjupersonerna visar att man inte kan ta för givet att kunden alltid har kompetensen att ställa rätt krav på produkten och då borde följaktligen produkttesterna vara mer omfattande än vad kunden krävt. Samtidigt visar det på behovet av en helhetssyn där alla aspekter på kvalitet mäts med objektiva mått vilket skulle gynna både producent och konsument.

Som vi sett tidigare anser flera forskare att en välutvecklad och certifierad process inte nödvändigtvis ger hög kvalitet. Problemet med att förlita sig på att processen ger kvalitet är inte att den automatiskt leder till dålig kvalitet, processen kan mycket väl ge hög kvalitet. Frågan är hur du vet att kvaliteten är hög om du inte har några mått som visar att så verkligen är fallet? Mått på produkten ger dig möjligheten att se om processen ger ett bra resultat och måtten ger också indikationer på om förändringar i processen ger positiva eller negativa resultat. Det finns således inget motsatsförhållande mellan process och mått, måtten är ett nödvändigt komplement.

5.1.3 Delsyfte 3

Ett delsyfte med uppsatsen har varit att jämföra den av litteraturstudien funna kvalitetsmodellen med hur systemutvecklingsföretagen arbetar med produktkvalitet. De kvalitetsmodeller som jag studerat och den jag valde att utgå ifrån ger ett visst utrymme för olika tillämpningar främst med avseende på vilka metrics man ska mäta och vilka gränsvärden man ska fastställa. Modellerna ger också utrymme för att utveckla egna rutiner för hur man ska gå till väga för att mäta kvalitet. Jag ville således undersöka hur ett antal företag gjorde och vad man verkligen mäter.



En direkt jämförelse mellan teori och verklighet är inte lätt att göra då inga av företagen i undersökningen tillämpar teorierna och modellerna som jag berört i uppsatsen i sin helhet. Jag avser dock att nedan diskutera likheter och skillnader i arbetssätt mellan resultatet av min undersökning och den teoretiska modell som jag beskrivit.

Vid en första anblick är den första och mest uppenbara lärdomen av studien att man idag inte mäter kvalitet på det sätt som modellen ISO/IEC 9126 visar. Att mäta är dock inte helt främmande för de intervjuade företagen men man mäter inte lika utförligt och så heltäckande som modellen visar. Jag kan konstatera att de viktigaste begreppen i kvalitetsmodellen, kvalitetsfaktorerna, var kända bland de flesta av intervjupersonerna. På flera av företagen arbetar man på ett eller annat sätt med ett flertal av kvalitetsfaktorerna bland annat genom att man har vissa rutiner och kodningsregler som påverkar kvaliteten på produkten. Detta gäller främst usability där man har riktlinjer för användargränssnitt och maintainability där man har kodningsregler som ska ge strukturerad kod som är lätt att förstå och modifiera. Vi känner igen tankegångarna från teoriavsnittet, man arbetar efter antagandet att det finns ett samband mellan interna attribut (strukturerad kod) och externa attribut (maintainability).

Det man testar mest är functionality och reliability vilket till stor del beror på att det främst är här som kunden ställer krav. Enligt intervjuaren så är kundens kravspecifikation till stor del en specifikation av de funktionella krav man har på systemet, följaktligen så är det viktigt att den levererade produkten lever upp till ställda krav vilket man också noggrant testar. Reliability är viktigt då reliability handlar om systemets förmåga att fungera trots felaktiga inmatningar och andra felsituationer. Ett system som står stilla har kunden ingen nytta av och exempelvis inom industrin kan ett stillastående styrsystem stoppa hela produktionsflödet med allvarliga ekonomiska och/eller mänskliga konsekvenser som följd. Systemkrascher innebär också kostsamt garantiarbete för leverantören varför detta testas noggrant. Exempel som framkom i undersökningen var att för att minska konsekvenserna av eventuella felsituationer ser man över flödena i systemet så att mindre fel inte slår ut hela systemet och på så vis bibehåller man en viss funktionalitet. I kvalitetsmodellen betecknas detta som fault tolerance vilket påverkar graden av reliability.

Enligt min undersökning var efficiency och portability det man ansåg minst viktigt och man arbetade med detta endast i de fall då kunden specifikt ställde krav i de kategorierna. Det var intressant att konstatera att svaren korrelerade med en stor undersökning, presenterad av Bache och Bazzana (1994), genomförd på företag i Finland, Sverige, Danmark, Italien, Spanien och Storbritannien som visar att functionality och reliability rankas som viktigast och efficiency och portability som minst viktiga.

Vi kan se av resultatet och ovanstående resonemang att begreppen inte var okända ute på företagen. Man arbetar också till viss del med kvalitetsfaktorerna men teorin



att mäta produktkvalitet var tämligen okänd. Kvalitetsfaktorerna är välkända begrepp för systemutvecklare från andra sammanhang men kvalitetsmodellen och dess underattribut var okända. I de fall där jag kunde se likheter mellan kvalitetsmodellen och verkligheten var sambanden inte ett resultat av att företagen hade studerat en kvalitetsmodell.

Det som jag anser skiljer teori och verklighet förutom omfattningen i tillämpningen är angreppssättet. Det man mäter mäter man för att uppfylla kraven i kravspecifikationen och inte för att få värden på uppnådd kvalitet, en liten men dock väsentlig skillnad. Att uppfylla kraven i kravspecifikationen kan ge nöjda kunder men mått på kvalitet kan ge mer än så för systemleverantören. Mått på kvaliteten ger möjligheten att utvärdera hur förändringar i processen påverkar kvalitetsfaktorerna och informationen kan användas för att jämföra projekt och förbättra sina metoder.

5.2 Tänkbara orsaker till att man inte mäter kvalitet

Resultatet visar att man inte mäter kvalitet på det vis som teoriavsnittet visar. Mina studier i ämnet *Software Quality Metrics* och min undersökning visar att några av orsakerna till problematiken kring att mäta mjukvarukvalitet ligger i följande omständigheter:

- Mjukvara är abstrakt
- Ny teknologi
- Kunden har dåliga kunskaper

5.2.1 Mjukvara är abstrakt

Mjukvaran är inte en fysisk produkt som vi direkt kan mäta på utan attributen vi mäter är mer diffusa och abstrakta. Detta får till följd att det är svårt att komma överrens om vad man ska mäta och vilka inre attribut som relaterar till vilka yttre attribut, något som också speglas av den flora av kvalitetsattribut som uppvisas i tabell 3.1. Kvalitetsattributen diskuteras fortfarande livligt idag drygt 20 år efter det att den första boken, ”Software Metrics” av Tom Gilb, utgavs 1978. Att ISO/IEC antagit en modell för bedömning av mjukvarukvalitet kan ses som ett tecken på att man börjar enas om vilka som är de mest centrala begreppen.

Att det förekommit många likvärdiga modeller och att det inte förekommit så mycket vetenskapliga studier där effekterna av att mäta mjukvarukvalitet har undersökts har enligt min mening också varit en faktor som hämmat tillämpningen av kvalitetsmodellerna. Nu har vi under senare år sett en ökad forskning kring ISO/IEC 9126 vars resultat talar för att det idag finns de teoretiska och tekniska kunskaperna att mäta mjukvarukvalitet.

5.2.2 Ny teknologi

IT-branschen är relativt ung jämfört med andra branscher. Teorier och utvecklingsmodeller har inte alltid hållit jämna steg med den snabba teknikutvecklingen vilket gjort att kvalitetsarbetet kommit på efterkälken (Eklund &



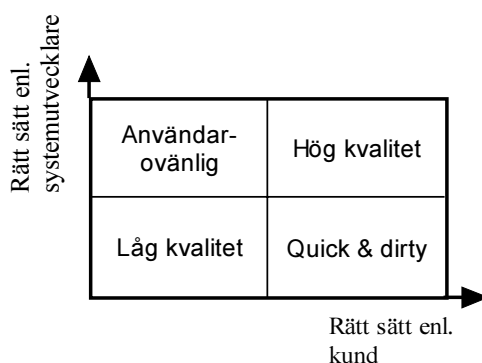
Fernlund 1998). Från början var mjukvaruprodukter koncentrerade kring enskilda beräkningar vilket medförde att programmen var välavgränsade och överblickbara. På senare år har det dock skett en teknikboom och idag involverar system ofta flera delsystem och programspråk som måste kunna interagera sinsemellan vilket har ökat komplexiteten avsevärt. Ökad komplexitet leder till ökad risk för fel men också till större svårigheter att testa systemen. En av respondenterna i min undersökning menar att kvalitetsarbetet inom IT-branschen hade kommit långt i slutet av 80-talet men lågkonjunkturen i början på 90-talet hämmade kvalitetsarbetet. När man sedan började satsa på kvalitet efter lågkonjunkturen hade den nya webbt tekniken blivit mycket komplex och svårtestad.

5.2.3 Kunden har dåliga kunskaper

En orsak till varför man inte mäter produktkvalitet är att kunden har dåliga kunskaper om IT vilket har medfört att man inte har ställt rätt och tillräckligt hårda kvalitetskrav. Att kunderna ställer hårda krav på produkten är en viktig faktor för att tvinga fram kvalitet. Många kunder har också av rädsla att halka efter investerat i teknik som man inte haft behov av. I min undersökning återkom ofta åsikten att kunden har dåliga kunskaper. Detta ställer inte bara till problem för kunden själv utan även för leverantören som ska försöka tillfredsställa en kund som inte känner sina egna behov.

5.3 Slutsatser

Studien av åtta systemutvecklingsföretag har lett fram till slutsatsen att systemutvecklingsföretag i dagsläget inte använder SQM-teorin för att bedöma mjukvaruprodukter utan man lutar i hög grad på att processerna ger eftersträvad kvalitet tillsammans med att man kontrollerar produkten mot kravspecifikationen (kundkrav).



Figur 5:1 Hög kvalitet nås om både systemutvecklare och kund är överrens.

Resultatet av min undersökning tyder på att man testar produkten efter kundkrav och leverantören når därmed förhoppningsvis långt på den vågräta axeln i figur 5:1 men för att samtidigt nå högt på den lodräta axeln bör leverantören ställa krav på att uppnå internt ställda kvalitetsmål.



Efter denna slutsats kan man med fog fråga sig om det är fel på verkligheten eller teorin. Jag anser att svaret, som så ofta, inte är svart eller vitt. Enligt min mening talar mycket för angreppssättet att mäta mjukvarukvalitet. Vad och hur man ska mäta har länge diskuterats men på senare år har olika forskningsprojekt visat att det är möjligt att mäta mjukvarukvalitet och man har visat att det finns en nytta med det. Min undersökning har också visat att man idag på företagen mäter vissa attribut (dock utan ett SQM-angreppssätt). Därför anser jag att det vore ett naturligt steg i kvalitetsarbetet att i framtiden ta steget mot att mäta mjukvarukvalitet fullt ut.

Detta leder oss in på varför man i dag inte mäter kvalitet, om det är fel på verkligheten. Enligt min mening är svaret att företagen inte har kommit dithän ännu. Flera av företagen i min undersökning var mitt uppe i att arbeta fram kvalitetsmetoder och bygga upp en organisation kring detta. Detta tyder på att man fortfarande har en lång väg att gå på vägen mot kvalitet i alla moment i systemutvecklingsprocessen. Andra företag hade kommit längre och där ansåg man att nästa steg kunde vara att ta fram mått på kvaliteten. IT-branschen är fortfarande en ung bransch och det tar tid att hitta formen för kvalitetsarbetet.

5.4 Uppslag till fortsatt forskning

5.4.1 Industrikunder är kunnigare

Redan under ett tidigt stadium under intervjuarbetet antydde några intervjupersoner att det var stor skillnad på kunskaperna om IT mellan de kunder man gjorde systemutvecklingsprojekt för. Bland de intervjuade företagen som arbetar mot både tjänsteföretag och tillverkande industriföretag sade man att stora industrier som Ericsson, Saab och Volvo generellt var mer inriktade på mätdata och att de ställde stora krav på kvaliteten. Jag fann detta intressant och var under det fortsatta intervjuarbetet uppmärksam på detta. Åsikten återkom i flera intervjuer, ofta helt spontant. Det var också de systemutvecklingsföretag som fokuserar mot industrin med utveckling av industrisystem som bäst kunde beskriva och ge exempel på specifika test och kvalitetsmätningar.

Ovanstående resonemang tyder på att det kan finnas en skillnad i synen på eller kunskapen om mjukvarukvalitet mellan industriföretag och tjänsteföretag. Detta skulle kunna bero på flera saker. Man har inom den tillverkande industrin en längre erfarenhet av att mäta kvalitet på sina produkter som kan ha påverkat synen på behovet av mått även på mjukvara. Med sin långa erfarenhet av kvalitetsarbete kan man vara bättre på att ställa krav på leverantören vilket har tvingat fram bättre metoder även hos leverantörerna. Den tillverkande industrin har även längre erfarenhet av ISO 9000 som ursprungligen skrevs speciellt för tillverkande industrier (Oskarsson & Glass, 1996).

Frågor för fortsatt forskning kan gälla om det stämmer att industrin generellt är bättre på att ställa krav på mjukvaran och om detta påverkat kvaliteten på de system man köpt.



5.4.2 Kalitetsmodellen som konsumenthjälpmedel

ISO/IEC 9126 är tänkt att kunna fungera som en bedömningsmodell för certifiering av mjukvaruprodukter. SQM-teorierna kan då användas inte bara som en metod för systemutvecklingsföretag att utvärdera sina processer och produkter utan skulle på så vis ge konsumenten ett viktigt hjälpmedel att bedöma mjukvara. Nyttan och det främsta användningsområdet för kvalitetscertifierade mjukvaruprodukter gäller, enligt min mening, i första hand generiska produkter då en potentiell kund på ett enkelt sätt skulle kunna jämföra likvärdiga produkter på marknaden genom att titta på certifieringen. Kunden skulle själv kunna jämföra uppnådda värden på de sex kvalitetsfaktorerna och avgöra vad man prioriterar, exempelvis efficiency framför usability etc.

För specialprodukter är nyttan för köparen inte lika uppenbar då varje systemlösning är unik och att uppnå en certifiering i det sammanhanget kan närmast ses som en marknadsföringsmässig framgång för producenten av systemet. Det har i min undersökning framkommit att kunderna är dåliga på att ställa krav. Ett förslag till fortsatt forskning är att undersöka om kvalitetsmodellen skulle kunna användas på ett omvänt sätt. En utveckling av kvalitetsmodellen skulle kanske kunna utformas som ett stöd för konsumenten vid upphandlingen av ett system. Ett hjälpmedel för att ställa rätt krav på den produkt man vill köpa.

5.4.3 Reflektioner över studien

Uppsatsarbetet har varit intressant och spännande. Det mest givande under arbetets gång har varit att komma ut till olika företag och göra intervjuer. Resultatet tyder dock på att mitt ämne är föga känt vilket kan tolkas som ett tecken på att det behövs mer forskning och resultat som visar på positiva effekter av kvalitetsmätning innan marknaden är redo. Jag är något förvånad över att kännedomen och tillämpningen av SQM var så låg som den var bland företagen. Trots att flera av företagen var kvalitetscertifierade enligt ISO/IEC 9000 hade de ändå ingen kunskap om ISO/IEC 9126.

Med facit i hand hade det varit intressant att ställa fler frågor kring hur man uppfattade möjligheter och begränsningar med ett sådant tillvägagångsätt (kvalitetsmätning). Vidare önskar jag att jag ställt direktare och precisare frågor om produktkvalitet. Det är lätt hänt att man inte håller isär begreppen när man diskuterar kvalitet och följden blir att man diskuterar kvalitet i allt för allmänna ordalag.



6 Referenser

6.1 Böcker

Andersen, E, S. (1994). *Systemutveckling - principer, metoder och tekniker (2:a uppl.)*. Lund: Studentlitteratur.

Asnaghi, V., et al. (1996). Software quality evaluation from research to industry: The Qseal Consortium approach. In S. Bologna and G. Bucci (Eds.), *Achieving Quality in Software* (pp. 213-222). London: Chapman & Hall.

Bache, R., Bazzana, G. (1994). *Software Metrics for Produkt Assessment*. London: McGraw-Hill Book Company.

Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.

Bryant, A., Grogan, J. (1996). Process Maturity - Growing older and wiser? Using Process Advisor for Process Assessment. In S. Bologna and G. Bucci (Eds.), *Achieving Quality in Software* (pp. 58-73). London: Chapman & Hall.

Bucci, G. (1995). The Impact of Software Quality. In P. Nesi (Ed.), *Objective Software Quality* (pp. 10-15). Berlin: Springer.

Dahlbom, B., Mathiassen, L. (1993). *Computers in Context -The Philosophy and Practice of Systems Design*. Oxford: NCC Blackwell.

Dahmström, K. (1996). *Från data insamling till rapport (2:a uppl.)*. Lund: Studentlitteratur.

Edvardsson, B. Thomasson, B. (1992). *Kvalitetsutveckling –ett managementperspektiv (2:a uppl.)*. Lund: Studentlitteratur.

Fenton, N, E., Pfleeger, S, L. (1996). *Software Metrics. A Rigorous and Practical Approach (2:nd ed.)*. London: International Thomson Computer Press.

Holme, I, M., Solvang, B, K. (1991). *Forskningsmetodik*. Lund: Studentlitteratur.

Jones, C. (1996). *Applied software measurement: assuring productivity and quality (2:nd ed.)*. New York: McGraw-Hill.

Kotler, P., Armstrong, G. (1996). *Principles of Marketing (7:th ed.)*. Englewood Cliffs: Prentice Hall.



Lundahl, U., Skärvad, P. (1992). *Utredningsmetodik för samhällsvetare och ekonomer* (2:a uppl.). Lund: Studentlitteratur.

Möller, K., H., Paulish, D., J. (1993). *Software Metrics. A practitioner's guide to improved product development*. London: Chapman & Hall.

Ohlsson, N. (1996). *Software Quality Engineering by Early Identification of Fault-Prone Modules*. (avhandling för Master Thesis, Linköping Universitet).

Oskarsson, Ö., Glass, R., L. (1995). *ISO 9000 i programutveckling*. Lund: Studentlitteratur.

Pettersson, S. (1999). *Rational Unified Process. En modell för kvalitetsfrämjande systemutveckling?*(avhandling för magisterexamen, Göteborgs Universitet).

Sommerville, I. (1995). *Software Engineering (5:th ed.)*. Essex: Addison-Wesley Publishers Ltd.

Trost, J. (1993). *Kvalitativa intervjuer*. Lund: Studentlitteratur.

Welzel, D., Hausen, H., L. (1996). A method for software evaluation with respect to quality standards. In S. Bologna and G. Bucci (Eds.), *Achieving Quality in Software* (pp. 381-399). London: Chapman & Hall.

6.2 Artiklar

Dahlgren, E. (2000). Falsk trygghet med stämpel. Rutiner som märks. ISO-certifiering ingen garanti för kvalitet, men det hjälper i marknadsföringen. *Dagens Nyheter*, 000105, B01.

Hedlund, T. (2000). Kvalitetsmärkt mjukvara håller längre. *Computer Sweden*, 29.

Foody, M., A. (1995). When is software ready for release?. *Unix Review*, 13, 35-41.

6.3 Internetkällor

EAGLES. (1995). ISO Terms and Guidelines, EAGLES Evaluation of Natural Language Processing Systems FINAL REPORT[WWW dokument]. 2000-02-01. URL <http://issco-www.unige.ch/ewg95/node54.html>



Appendix

A.1 Intervjumaterial

A.1.1 Frågemall

Person- och företagsinformation

1. Vill du beskriva vad du arbetar med och hur du arbetar med kvalitetsfrågor.
2. Arbetar ditt företag ute hos kund och/eller med egna projekt?
3. Lyder ni då under kundens kvalitetsmetod eller arbetar ni efter er egen metod?
4. Är skillnaden stor?
5. Vad är normalstorleken på era projekt? (tid, personalåtgång)

Strategiska mål för kvalitetsarbetet

6. Vad är företagets syn på kvalitet?
7. Vad har ni för övergripande mål med kvalitetsarbetet?
8. Varför är det viktigt? Konkurrensmedel, kundkrav, etc
9. Kan det vara så att kunden inte är beredd att betala för den eventuella mertid och merkostnad som ett rigoröst kvalitetsarbete innebär?

Kvalitet på projektnivå

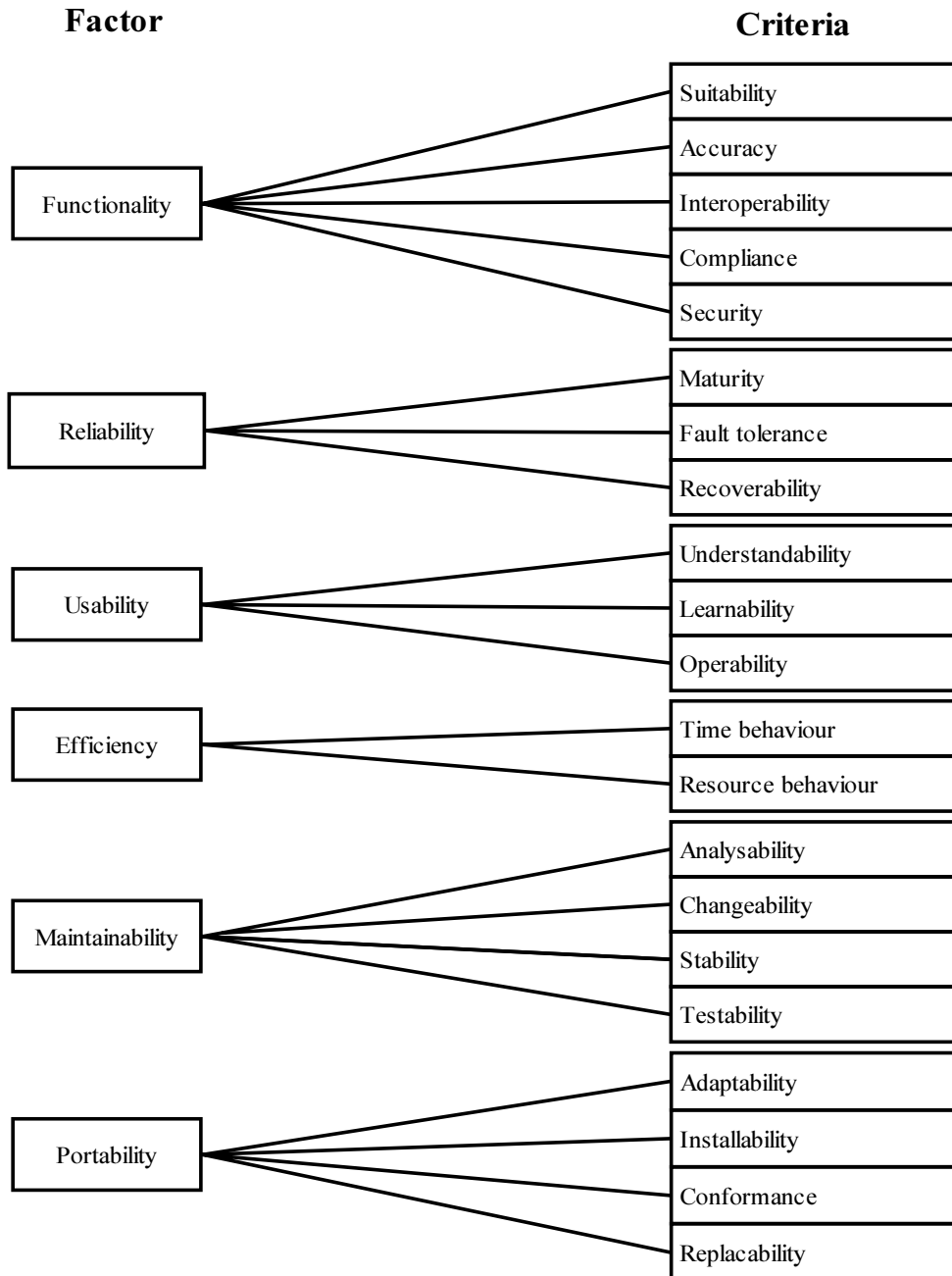
10. Inför ett nytt projekt kan du beskriva hur man planerar kvalitetsarbetet.
11. Följer ni en utstakad metod?
12. Har ni utvecklat er egen kvalitetsmetod?
13. Ser kvalitetsarbetet annorlunda ut vid olika typer av projekt? Blir olika faktorer/begrepp olika viktiga.
14. Vilket ansvar för kvaliteten har deltagarna i projektet?

Produktkvalitet (kvalitetsmodellen)

15. När under ett projektarbete förlöpande testar/kontrollerar man olika kvalitetsfaktorer?
16. Vilka begrepp anser ni är viktigast när man mäter kvalitet?
17. Hur går ni till väga för att mäta kvaliteten?
18. Vilka typer av tester använder man?
19. Är det något viktigt som vi inte fått med under intervjun?



A.1.2 Figur





A.2 Intervjuer

A.2.1 Intervjupersoner

Lars Andersson, Mandator. Besöksintervju 99-11-09.

Magnus Andersson, Sigma Systems. Besöksintervju 99-10-28.

Magnus Bergenkull, WM Data Caran Electronics. Besöksintervju 99-10-20.

Josef Bilek, KSg Systems. Besöksintervju 99-10-19.

Krister Johansson, Prevas. Besöksintervju 99-10-27.

Maria Jonsson-Kjellström, Enator Informationssystem Väst AB. Besöksintervju 99-10-21.

Göran Lindkvist, IBS. Telefonintervju 99-11-10.

Lennart Sahlberg, ICL. Besöksintervju 99-10-26.



A.2.2 Intervjusvar

1.

Jag ansvarar för kvalitet för södra Sverige. Jag arbetar då främst med våra modeller för kvalitet i projektarbete och quality assurance plan.

2.

Eftersom vi säljer en standardprodukt går våra projekt ut på att skraddarsy produkten för den enskilde kunden. Det är i den rollen vi finns ute hos kund. Inom företaget arbetar vi med förbättringar och nya versioner av vår produkt.

3.

I regel arbetar vi efter vår egna metod. 80% av våra kunder har inte så välutvecklade kvalitetssystem.

4.

Ja.

5.

150 - 1000 mandagar. Normalt 200 mandagar.

6 & 7.

Det vi levererar ska vara felfritt. Det ska stämma med kundens krav. Produkten ska täcka kundens behov. Nöjd kund.

8.

Kunden har krav på bra kvalitet på projekt och produkt. Kan man inte visa på ett bra kvalitetsarbete så är man inte ens intressant som leverantör hos vissa företag. Mest ett konkurrensmedel.

9.

Ja det märker vi hos mindre företag. De stora industriföretagen som själva har höga kvalitetskrav har en bättre förståelse.

10.

Vi följer vår metod som i stora drag går till enligt följande: Vi lägger mycket tid i början av projektet då vi ska utforma en kvalitetsplan, QAP. Detta är ett viktigt dokument som måste vara väl förankrat hos kunden. Planen innefattar mål för projektet, kravspecifikation., checklistor, vad som ska göras, hur, av vem och när. Sedan genomförs en review an QAP'n. Kodgenomgångar, mätningar av svarstider i systemet och tester sker löpande i projekten. Vi väljer också ut projekt som specialstuderas. Då går man in och kontrollerar vissa kritiska tidpunkter och milstolpar under projektets fortskridande. När projektet är avslutat går man igenom hela projektet för att kunna jämföra med erfarenheter från tidigare projekt.

11.

Ja.



12.

Egen utvecklad.

13.

Hos oss är kvalitetsarbetet ganska lika från projekt till projekt eftersom vi säljer en standardprodukt och även datamiljön är alltid samma, IBM's AS 400.

14.

Den enskilde programmeraren har en kvalitets mall för sitt arbete med tester etc. I övrigt har olika befattningshavare olika checklistor.

17.

För oss är functionality och reliability viktigast. Funktionaliteten kontrollerar man mot kravspecifikationen. Usability vad gäller gränssnitt så utformar vi det med kund och utbildning lägger vi stor vikt vid. I våra åtaganden läggs 1/3 av tiden på utbildning och assistans. Efficiency är inget vi studerar närmare. Reliability testar vi utförligt bl.a. så kör vi vårt nya system med uppgifter som finns i kundens gamla system för se att resultatet blir det samma. Vi kör även flödestest då vi testar minst 80% av systemets funktionalitet. Maintainability är en faktor som vi har i åtanke i kodningen. Vi har kodningsregler och vissa strukturer som ska följas. Portability är av mindre betydelse då vi alltid jobbar mot AS 400. Däremot släpper vi hela tiden nya versioner av olika moduler i vårt system och då måste de olika versionerna fungera ihop med övriga delar i systemet.

1.

Företagets fokus är att som Microsoft solution provider, inte bara implementera MS lösningar rent tekniskt utan även hjälpa till med total cost of ownership och return on investment som är viktigt för kunden. Vi erbjuder våra tjänster för hela processen från det tekniska till projektledning och helpdesk etc. Vi erbjuder även hjälp inför ISO certifieringar av företag. Certifieringar av IS/IT-organisationers rutiner, struktur och dokumentation.

2.

Renodlad konsultverksamhet ute hos kund.

3.

Beror på hur välutvecklad kunden är IT-mässigt. I Sverige överlag är IT-mognaden på en låg nivå. Många företag som det gått bra för har investerat i IT och sedan kört på och glömt kvaliteten och om det man gör är kostnadseffektivt. Det beror alltså på vilken kund vi har, i vissa fall tar dom in oss för att ge dom kvaliteten. Idag till exempel har vi slutcertifiering av en kund som vi har tagit fram ett kvalitetssystem åt. Till 80% är det vi som kommer med kvaliteten.

4. Är det stor skillnad mellan kundens och ert kvalitetssystem, poängterar man samma saker?

Det handlar om bra kvalitet på supporten, fungerande system, sänka downtime, nå projektets tidsmål. I detta projektet till exempel finns en stark IT-chef hos kunden med klara mål. Han har släppt IT projekt till oss och vi gav honom verktygen för att kvalitetssäkra med viss dokumentation som han accepterade sen körde vi på tillsammans. Kraven finns där, att projekten ska klaras av, målsättningarna skall uppnås sen kvalitetssäkra dom målen och det ska ske inom en viss kostnadsbegränsning. Man kan inte driva projekt utan att veta



att dom kostar och om dom försenas kostar dom mer. Kommer man överrens om detta så ligger kvalitetsmålsättningarna ganska lika. Vill dom ha en snabb lösning blir kvaliteten därefter.

Och det är kunderna medvetna om?

Vi är helt öppna mot våra kunder på det sättet. Vi vill att vårt arbete ska tala för sig långsiktigt.

5.

Allt från en enkel webblösning eller en enkel installation till en ISO certifiering som pågår i ett halvår. Integrationsprojektet här har pågått sen januari.

6.

Vi tror på att levererar vi resultat inom kostnadsramar med uppföljning och korrekt avlämning då slår vi kundens förväntningar. Vårt ledord är att slå kundens förväntningar på oss.

8.

Konkurrensmedel från vårans sida. En sund tanke från kunden måste vara att få nytta för pengarna.

9.

Det kan man klart säga. Det beror nog på att vissa kunder inte är mogna, kan bero på att dom har väldigt bråttom just nu eller en dålig IT ledning. Ibland måste vi in och bevisa oss. För en ny kund måste du alltid in och bevisa dig. Då försöker vi redan i första kontakten med kunden prata om hur vi kvalitetssäkrar hur vi driver projekt och att det här får ni med oss. Då hoppas vi att dom är villiga att betala för det.

10.

Vi arbetar efter en enkel metod som kallas för PDCA (Plan Do Check Act). Vi har vissa verktyg och procedurer för att planera projekt. Planering, själva implementationsfasen är viktig att den implementeras efter planeringens villkor. Vi kontrollerar vårt arbete, kvalitetssäkrar sen agerar vi efter eventuella missar eller saker som inte uppnåtts. Vi ser till så att när vi står vid punkten efter Act så har vi ett färdigt dokument med våra kunder där vi signar av med IT chefen eller den ansvarige för projektet ett dokument. Förklara proceduren... Företags hemligheter Vi har ett dokument för varje projekt som vi kallar scopedokument där definieras klara mål med projektet, kostnadsramar och sånt som kunden tycker är viktigt med projektet som vi ska uppnå. Detta dokument skriver både jag och kunden på, då vet vi precis vad vi har att göra och vi kan gå ut och implementera detta. Med basis av detta scopedokument skapas ett wrapupdokument. Där anger vi punkter för hur saker ska utföras vi kan sedan bevisa att punkterna i scope dokumentet har avklarats till din tillfredsställelse. Efter projektets slut går vi igenom dokumentet i punktform och skriver på med kunden.

Vet och kan kunden specificera vad dom vill och vilka krav dom har?

Vi hjälper till att definiera krav men oftast är kunden rätt kunnig. Vi vill att det ska bli ett scopedokument som motsvarar kundens förväntningar. Uppföljningen, handsoff processen är väldigt viktig.



Vi följer PDCA modellen, vi träffar kunden och pratar om det som ska göras och samtalet leder till ett scope dokument. Då vet vi vad det handlar om och efter det gör vi en projektplan. Vi använder då en standard, Microsoft project. Beroende på projektets natur gör vi en projektplan som speglar de olika faserna. Planering, implementering, testning, agera, dokumentation. I dom olika faserna lägger vi in kvalitetstestning. Vi ska arbeta på det sättet. Själva revisionen av projektplanen fungerar så att vi stämmer av arbetet, tid, deadlines etc så att vi kan se hur vi når våra mål. Vi träffas inom Företaget och ser över hur projekt går. Sen när projektet är klart signaleras projektet av. Vi har inte bara en projektplan utan det finns även andra dokument som syftar till att specificera upp mindre faser, tasklistor där man bockar av att saker är gjorda. Där kräver vi tillbakarapportering via e-mail eller veckomöten. Det ska drivas på under ordnade former.

13.

Inte för oss vi går efter denna modellen.

14.

Alla som är inblandade i projekten är inblandade i kvaliteten. Det är upp till projektledaren att bestämma hur mycket av resurserna som ska vara inblandade i kontrolleringen och vilka uppgifter dom har. Dom flesta har en tasklista som dom rapporterar tillbaka.

15.

Vi har obligatorisk buggtestning så långt vi kan enligt scopedokumentet. Funktionaliteten testas fullt ut med exempelvis genomgång av flowcharts där man funktionellt processmässigt följer en process i ett program. När man trycker på den knappen vad genomförs då, också följs det ända ner till roten och det kontrolleras (i en databas) att det genomförts. Så är man tvungen att göra med varje krav på produkten. Vid implementation av ett nätverk då får man se till klienterna och vad som behöver installeras, vad serverna behöver och hur kommunikationen ska ske. Det finns också många små små avvikelser som kan orsaka fel. T.ex. kan MS produkter ha buggar och då måste vi tala om för kunden att buggarna finns och så får man avgöra om man ska satsa på produkten i alla fall eller ej. Det är betydligt svårare att testa projekt som inte är ren programmering. Det kan vara så att det finns buggar i operativsystemet ute hos kunden något som vi inte råder över. Därför försöker vi klart definiera våra mål så att vi kan testa kvaliteten i det vi gjort för kunden resten ligger utanför vårt ansvar.

16.

Det som vi ser här är helt klart alla viktiga saker. Svårigheten i konsultbranschen är att sälja in detta att kunden ska vilja testa allt detta. Man måste ha att göra med en otroligt kompetent IT ansvarig som förstår vikten av allt detta. Det beror även på projektet, är det ett projekt med en viktig lösning, systemet får aldrig krascha det ska vara redundans och backup då måste man givetvis testa detta. Det beror på kundens förväntningar. Det är upp till projektledaren att se hur vi ska testa och verifiera målen i wrapupdokumentet.

Fler exempel på tester?

Det finns många olika användare på olika funktioner inom kundens organisation då måste de olika typerna av användare få testa applikationen eftersom olika användare har olika behov.

19.

Vi har berört det mesta.



1.

Personalansvarig. Arbetar med kvalitetsfrågor på så sätt att det delvis är en utbildningsfråga. Även marknadsansvarig på industrisystem där vi säljer våra stora fastprisprojekt och då säljer vi mycket på kvalitet och kvalitetstänkande. Vi säljer mycket på att skriva om gamla system som går i produktion vilket man inte kan göra om man inte står för en hög kvalitet.

2.

Här i Göteborg jobbar vi mycket med fastprisprojekt vilket innebär att vi sitter ute hos kund och förhandlar i affären och kräver kunden på relativt goda underlag i affärsuppgörelsen. Sen går vi in i en fas av detaljspecifikation och försöker göra en specifikation över systemet tillsammans med kunden. Sen sitter vi hemma i stort sätt fram till produktionsstarten. Då åker man sen ut till kund och installerar. Sen har vi även en viss konsultverksamhet där våra konsulter sitter ute hos exempelvis Ericsson och Volvo.

3.

När man går ut som konsult går man oftast in under kundens kvalitetssystem.

4.

Det är väldigt olika. Kvalitet har lätt för att bli teori om man pratar om ISO 9000 men kvalitet är väldigt mycket praktik. I metoder räknat är våra metoder ganska lika i och med att vi är industri- och tekniskakonsulter. Något att betona är att även om man tycker att en kvalitetsmetod är bra så ger den inte självklart bättre kvalitet. Jag har jobbat på företag där kvalitet sitter i väggarna och då spelar metoden ingen roll man gör rätt ändå. Sen finns andra som har väldigt strikta regler där allt är reglerat i detalj men det ger inte kvalitet trots allt.

Hur man väljer att styra präglar kvaliteten. Men i stort sett är skillnaderna mellan oss och kund marginella.

Man kan tänka säg att vissa kunder inte har en IT medvetenhet som gör att man inte kan ställa rätt krav?

Vi har haft lyckan att jobba med kunniga kunder inom industrin (Volvo, Saab) vilket gör att vi hjälper dom och dom hjälper oss att göra kvalitativsystem. Dom sämsta systemen är dom man gör för kunder som inte vet, för dom föreställer sig att våra system är något man bara sätter på och av.

5.

6-18 månader, 1-10 man. Normalt 12 mån 2-3 man.

6.

Den teoretiska delen är att få certifieringen som även till vissa delar hjälper oss marknadsmässigt. Vår strategi är att klara av att byta ut ett gammalt system i flygande drift utan att egentligen få några stopp. Nu sker många utbyten under semestrar men även då har det blivit tajtare (max två veckor). Det är inte ovanligt att vi byter ut hela styrsystem på en helg. Det vi fokuserar på är leveranstid. Det kostar oss mycket mindre i prestige, marknadsföring och pengar att lägga tid innan leveransdatum än vad det kostar oss att missa leveransen.

8.



Kundkrav givetvis men framför allt ett konkurrensmedel. Vi är ju ingen drake i sammanhanget så vi måste kunna visa att vi kan leva upp till industrins hårda krav på kvalitet. Den stora vinsten är marknadsföring.

9.

Det är absolut så att de utbildade kunderna är svårare att jobba med dels för att dom inte förstår vårt kvalitetsarbete och dom förstår inte varför vi ställer krav på dom. Det är t.o.m. så att vi har tackat nej till jobb för att vi anser att kunden inte klarar av att leva upp till våra krav och ett dåligt genomfört jobb ger oss ett dåligt ryckte. Sen är det så att vi som jobbar här har kommit in i ett sätt att jobba och då kan man inte plötsligt sänka sig för att vi tar en kund som inte ser kvaliteten så viktig.

10.

Man upprättar en tidsplan med olika aktiviteter och i den så ligger då olika typer av granskningar. Där vi lägger in mycket jobb och kraft är den första fasen då vi jobbar tätt med kunden för att specificera eller designa systemet. Tidsplanen får projektledaren även redovisa för annan personal. Vi vill ha granskningen till något positivt, att man kan få idéer och feedback i ett tidigt läge eftersom det är då det kostar lite pengar. Sen har vi kodgranskningar planerade i tidsplanen. Vi har även en tradition av att göra rätt omfattande testprotokoll som går igenom ihop med kunden i emulerad miljö (testmiljö) annars klarar man inte av att införa systemet under en helg. Man kör alltså hela systemet helt torrt. Det är i huvudsak aktiviteterna som finns i vår modell; granskningar under designfasen, granskningar under konstruktionsfas. Man granskar delar i den mån det går. I designfasen granskas helheten och delarna tillsammans.

11 & 12.

Ja. Det är en egenutvecklad modell som baseras på den erfarenhet som de två grundarna hade med sig. Sen i början på 90-talet när man började tala om ISO så fick man dokumentera ner metoden lite noggrannare. Så det är denna som ligger i botten även om den vidareutvecklats en hel del.

13.

Kvalitetsarbetet blir olika beroende på projektledaren men ska egentligen bli identiskt, även om det finns punkter som regleras av modellen.

Kan det vara skillnader beroende på systemet? Säkerhet framför snabbhet?

Där är det marknadsområdet som avgör, som ansvarig för industrisystem kan jag säga att det är leveranstillfället som är det viktiga. Där tycker jag att vi lyckats väldigt bra, vi levererar kvalitet trots att det är en deadline som är viktig. Inom medicin och läkemedel är det säkerhet som är viktigt. Inom CQR, IT säkerhet till affärsnytta, är kvaliteten säkerhets- och sårbarhetsanalyser. Där är det marknadssegmentet som gör att vi får olika fokus.

14.

Varje enskild medarbetare är ansvarig för det arbete han utför. Projektledaren är ansvarig inför projektägandechef, han i sin tur är ansvarig för regionchefen. Det blir en trappa och det är därför det är så viktigt att ha "kvalitet i väggarna" annars kommer inte medarbetaren som utför jobbet att tänka i kvalitet. Kvalitet ska komma underifrån. När kravet på oss att öka kvaliteten kommer underifrån då har vi kommit långt.

15.

Fortlöpande.



16.

När det gäller industrisystem kan man säga att leveranstid och tillgänglighet är A och O. Det ska vara klart i tid och industrin accepterar inte 98% tillgänglighet, de vill ha 100%-ig tillgänglighet. I industrin är portability inte så viktig. För ett industrisystem kan man säga att reliability är skitviktigt men portability är inte så viktigt, att det går att flytta till en annan plattform är mer en vinst för oss än för kunden. Att något vi utvecklat för VMS kan vi använda för NT, LINUX etc. Funktionalitet är det viktiga speciellt i vårt läkemedelssystem där begreppet spårbarhet ingår så att man kan härleda var, när, hur och varför en etikett skrivits ut. Vidare är usability viktigt, att det är lätt att använda så att man inte kan göra fel. Inom CQR har jag svårt att säga vad som är viktigt av dom här. Säkerhet saknas i figuren det blir nästan en egen faktor.

17.

Tillgängligheten mäts hos oss med hur många minuter systemet inte fungerar och då är det ofta så att när något inte fungerar då står allt still. Detta beror mycket på hur flödena ser ut, det kan vara så att ett fel slår ut t.ex. en kran men om det finns flera kranar så innebär detta att en viss procent är utslagna men det finns andra som fortfarande går. Det har vi inga problem med att mäta däremot kan det vara svårt att komma överrens med kunden om procentsatsen. 99.5%, 10 min/dygn detta mäts under en månad.

Luddigare saker som att strukturera kod, har ni riktlinjer för det?

Ja, det går inte att mäta och det finns säkert lika många bra sätt som det finns programmerare. Det som vi poängterar med strukturerad kod är maintainability, vem som helst ska kunna läsa en kod och förstå den. Man kommenterar på ett sätt, man strukturerar IF-satser på ett sätt. Enkelkod, approgrammering.

18.

Kommunikationstester mot överliggande och underliggande datorer och detta kan nästan helt göras i torr miljö. Vi har även tillfällen då operatörerna får komma hit och göra s.k. bänktester i emulerad miljö. Vi bygger upp ett program i bakgrunden som simulerar rörelser i det fysiska systemet på detta läggs vårt system och så får operatören testa allt från skärmdialoger. I designfasen ritar man GUI på papper sen en till två månader innan implementation sitter man ner med kunden och går igenom allt.

19.

Vi är duktiga på att mäta vårt arbete på ett sätt som inga andra IT-konsultbolag. Vi har som mål att vi ska ha mindre än 2% garantiarbete och det är ju ett mått på hur bra system vi gör. Får vi göra mycket garantiarbete så har vi ju inte gjort ett bra system. 2% är en låg siffra och de senaste fyra åren har vi legat under det på 1.6%. För varje projekt sätts det upp en projektkod till den projektkoden sätter man upp en massa underkoder beroende på vad vi jobbar med för att kunna jämföra och se att då tog det si och så långtid.

1.

Jag ansvarar för en generell systemutvecklingsgrupp som består av c:a 20 konsulter. Jag ansvarar för försäljning, personal och uppdrag.

2.



Huvuddelen av våra konsulter sitter ute hos kund men vi har även ett antal projekt i huset. C:a 80% sitter ute hos kund i olika roller allt från nytexaminerade programmerare till erfarna projektledare.

3.

Det kan man säga. I de fall där kunden har en metod eller ett arbetssätt som de vill att våra konsulter ska använda så anammar vi det sättet. Däremot om det inte finns så går vi efter vårt eget verksamhetsstyrningssätt.

4.

De kunder som verkligen är måna om att jobba med kvalitet har i grunden ett liknande system som vårt, det är snarare graden av tillämpning som skiljer. Generellt är nog konsulter mer drivna i detta än vad folk ute på IT-avdelningarna är. Det sitter mera i ryggmärgen att man ska jobba strukturerat och med specifikationer etc. Även om vi inte mäter kvalitet och jobbar med ISO så mycket i den här branschen.

5.

3-5 deltagare under 6 mån.

6.

Vi försöker att utbilda våra konsulter så mycket som möjligt att jobba efter vårt verksamhetssystem och däri ligger ett antal kvalitetsbitar invävda. Vi har idag ingen riktig QA funktion men det är något vi kommer att skaffa framöver. Mycket p.g.a. att vi får fler åtaganden där vi själva driver projekten och detta ställer högre krav på vår inre organisation.

7.

Målet är att kunna driva projekt med kvalitet både vad gäller innehåll, kostnad och tidsplan och att dokumentera detta väl innan man lämnar något ifrån sig.

8.

Ett kundkrav delvis men framförallt ett sätt för oss att inte få skit i efterhand. Har det varit en konsult från Företaget på plats så ska någon kunna komma efter och ta vid och det ska fungera. Ofta är det det som är problemet att om man inte jobbar efter någon metod eller modell och dokumenterar det du gör så är det väldigt svårt för någon annan att ta vid, t.ex. om förvaltningen av systemet tas över eller om konsulten byts ut av någon anledning. Det ska ändå vara relativt enkelt för den som tar vid. Där finns ju då poängen med att jobba strukturerat och dokumentera och ha ett kvalitetstänkande i det man gör.

9.

Det som är svårast är att få kunden att förstå att det är en initial kostnad att driva igång ett projekt enligt någon form av kvalitetsmodell. Det kanske ser dyrt ut vid en första anblick och kunden frågar sig varför man ska behöva hålla på med sånt men det gäller att se helheten och gör du inte rätt från början så kostar det ännu mer men det syns inte inom projektets ramar utan det kommer som en förvaltningskostnad senare. Där är kunderna idag dåliga på.

Är det en skillnad mellan olika kunder?

Ja, det är olika mognadsgrad hos kunderna. De större företagen som Volvo där följer man sin egen kvalitetsmodell där har man insett att man måste jobba strukturerat. Dom har ju i



många fall hela avdelningar som bara jobbar med kvalitet. Sen kan det vara så och så med hur man efterlever detta. I vissa fall får våra konsulter väldigt lite dokumentation och direktiv och får man luddiga direktiv är det svårt att mäta hur bra man utfört arbetet.

10.

Det är lite olika beroende på om det är ett projekt eller ett enskilt uppdrag. När det gäller enskilda uppdrag är kundens åsikter viktiga, vilka kvalitetskriterier som han tycker är viktiga. Vad är viktigt för honom för att han ska tycka att det här är kvalitet. Det ligger sedan till grund för vad vi mäter.

11.

Ja.

12.

Den är framtagen internt på Företaget men naturligtvis är det en produkt av samlade erfarenheter.

13.

Projektmålen visar vart tyngdpunkten i kvalitetsarbetet hamnar. Om projektets mål är att bli klart i tid så styr det, om det är kostnad så styr det. Är det innehållet t.ex. hög kvalitet på gränssnittet så får man gå på det. Det nya med webbsystemen är att det är mycket mer komplex miljö runt omkring vad gäller infrastruktur och allt sånt så kvalitetsbiten blir allt viktigare framförallt testning. Och gäller det system som kan innebära personsador så blir kvaliteten ännu viktigare i både processerna och testfasen. Om man talar om branschen i allmänhet så har vi kommit långt med kvaliteten men borde kanske kommit längre men det beror till viss del på att det blivit betydligt svårare att testa systemen p.g.a. komplexiteten.

Så även om medvetenheten om behovet av kvalitet kommit ikapp så har tekniken sprungit ifrån?

Ja, lite så är det plus att under början av 90-talet under lågkonjunkturen så kom metoder och kvalitet i skymundan i kombination med teknikskiftet som gjort allt komplexare. Att skriva kodningsregler kunde man göra när det bara fanns COBOL men nu finns så många världar så det gäller att kodaren har en viss grundkompetens.

14.

Kvalitetssäkringen ligger i första hand hos projektledningen.

15.

Trenden är att man kör mer och mer inkrementellt bit för bit och släpper en version som du testat innan du bygger nästa. Du gör inte hela systemet och sen testat utan det finns naturligtvis ett antal steg i testningen. Programtest, systemtestning, integrationstestning och till slut acceptanstest. Det talas egentligen för lite om tester och testmetodik i branschen men vissa företag typ Ericsson ligger långt framme där.

16.

Vi litar på att individen gör ett bra jobb och att vi tar in rätt folk från början. Projektledaren drar upp riktlinjer. Gränssnitt t.ex. där arbetar vi med användaren i workshops. Det är ju viktigt att gränssnittet upplevs positivt.

Det finns metoder som att mäta uppkomna fel under 1000 timmars körning, använder ni sånt?



Tja, i förvaltningsåtaganden så loggar man ju allt som händer och då kan vi gå in och mäta.

19.

Det kräver rätt ansvarskänsla hos individen. Oavsett hur bra metoder du har så kräver det rätt inställning hos individen.

Vi garanterar våra kunder viss inställelse eller åtgärdstid och den ska vi uppfylla till en viss procent och det kan vi mäta. Vi har ett antal andra sådana kvalitetstal som vi kan mäta upp varje månad. Dom kvalitetstalen speglar då till viss del hur bra vi utför en tjänst. Vi har flödesscheman efter de processer som vårt arbete följer och vissa punkter går då att mäta m.h.a. våra rapporteringssystem och det är grunden i vårt kvalitetssystem.

1.

Jobbar som regionchef för systemservice och vi jobbar med service, installation och tjänster av system. Ansvarig för södra Sverige. Som ansvarig är jag även ansvarig för kvaliteten. Vi ska uppfylla vissa ISO kontroller och rutiner. Vi har krav på oss att vi ska kontrollera våra instrument så att de mäter rätt. Att saker vi får in för reparation märks upp så att det alltid går att se var i processen de befinner sig. Det är det dagliga arbetet. Sen har vi många interna processer protokoll för möten, åtgärdslistor.

2.

90% är ute hos kund. Antingen resurser på längre projekt eller att vi åker ut på uppdrag som vi får in samma dag.

Gäller det både mjukvara och hårdvara?

Ja.

3.

Man jobbar under dom kvalitetskrav som vi har på oss. Det finns ju företag som inte är ISO certifierade till exempel men vi arbetar alltid efter våra egna krav.

5.

Några veckor. Mer parten av våra projekt är att vi åker ut till kunden och lagar det som gått sönder. Installationer kan ta en till två veckor.

6.

Vi har en väldigt fin kvalitetspolicy där målet är nollfel. Inom företaget har vi en väldigt hög kvalitetspolicy vi har jobbat länge med ISO. Vi var det första IT företaget i världen som certifierades.

7.

Att man ska uppfattas på marknaden som ett seriöst företag med hög kvalitet på en levererad tjänst.

8.

Framför allt ett konkurrensmedel. Gör vi ett jobb med bra kvalitet i de uppdrag vi har så slipper man gå tillbaka en gång till och kanske göra om vissa bitar, det är effektivt. I våra serviceavtal är det ju så att ju bättre vi gör det första gången desto effektivare är vi både för oss själva och för kunden.



9.

Kunden vill alltid ha det snabbt men inte till sämre kvalitet.

Har kunden en dålig förståelse för ert kvalitetsarbete?

Nej, det tycker inte jag. Även om dom själva inte är ISO certifierade ifrågasätter dom aldrig sättet vi arbetar på, dom upplever själva att det är viktigt att nå upp till kvalitetskraven.

10.

Vi har rutiner för hur våra projekt ska drivas allt ifrån försäljning och projektstart ner till systemleverans. Det finns beskrivet i våra rutiner t.ex. hur man avrapporterar. Det är ofta något man kommer överrens om med kunden. Kunden sitter ju ofta med på något sätt. Ibland i stora projekt arbetar man ihop med andra leverantörer det kan även vara externa konsulter som driver projektet.

Man stämmer alltså av under tidens gång?

Ja.

12.

Ärligt talat vet jag inte.

Hur fungerar det? Är det mycket dokument och checklistor etc?

Ja visst det är det. I t.ex. ett installationsprojekt där man installerar ett system där finns det beskrivet hur man gör en förkontroll, vilka dokument man använder och hur man rapporterar till andra system och vilka flöden vi har.

Är det viktigt att man stämmer av med kunden vad som ska göras från början? Att man har en kravspecifikation som båda är överrens om så att det inte blir diskussioner i efterhand.

Ja, det är oerhört viktigt att man dokumenterar vad man kommer överrens om redan i upphandlingen om hur systemet är tänkt att fungera så kunden inte i ett senare läge kan säga att så här ville jag egentligen inte ha det. Det är svårt för det är många gånger en tolkningsfråga. Där kan uppstå problem ibland. Men köper man t.ex. en färdigspecifierad tjänst som vi har i vår tjänstekatalog då kan vi få en exakt specifikation redan i första kontakten med kunden. Ibland om kunden sitter med många system erbjuder vi även en förstudie och utifrån det bestämmer man vad som ska göras. Gamla system ska konverteras till nya etc.

13.

I sådana här sammanhang styr kunden i mångt o mycket. Vad vill man ha för säkerhet i sin brandvägg exempelvis.

14.

Alla har kvalitetsansvar i alla lägen. Projekt ledaren har inte ett övergripande kvalitetsansvar, utan var och en ansvarar för sina arbetsuppgifter.

15.

Hård och mjukvaran ska vara testad i ett tidigare skede men i våra projekt testar vi först i simulerad miljö sen gör vi en eller flera pilotinstallationer. Detta får då tuffa och gå så att vi ser hur det fungerar tillsammans med andra system som finns på plats. Detta utvärderar oftast kunden för det är han som vet hur dom vill att det ska funka. Med den utvärderingen som underlag får vi eventuellt gör applikationsändringar.



Har ni då gränser för hur många fel som får uppstå eller liknande normer?

Det beror på vad det är för fel som uppstår. Det kanske bara finns ett fel men det kan man absolut inte leva med. I mjukvaran ställs man alltid inför dilemmat: Hur allvarligt är felet? Står det i relation till kostnaden att rätta till det. Det får man överväga i varje enskilt fall. Hårdvaran går i allmänhet sönder inom en vecka om det är något fel på den.

16.

Vi har ingen egen utveckling utan vi plockar ihop Microsoft program. Vet ej hur man jobbar med detta.

18.

Det vi testat är för att hitta unika fel som uppstår hos en enskild kund. Då mäter vi på nätverket med en så kallad "Sniffer". Sniffen får stå på t.ex. ett dygn och sen analyserar vi den datakommunikation som skett. På så vis kan vi se om det förekommer annan kommunikation som stör vårt system.

19.

Spårbarhet.

1.

Jobbar som ansvarig för Test och Metodik på Företaget. Tanken är att jag ska ansvara för utveckla kompetensen inom test och verifiering samt testmetodik på dom bitarna som ingår i vår kvalitetssäkring. Jag har även varit med och tagit fram det kvalitetssystemet vi har.

2.

Både och.

3.

Ja, inom de delar som innefattar själva utvecklingen men när det gäller försäljning och avslut av själva uppdraget då är det vårt eget som gäller. Man lämnar över ansvaret i och med att konsulten börjar arbeta för kunden men sen har vi uppföljningar och riktlinjer för hantering av affären.

4.

Ibland, ibland inte. Vissa kunder har långt utvecklade rutiner i programvaruutveckling och elektronik, hur man testat och vilken typ av dokumentation man tar fram andra har ingenting i den vägen.

Kan det bero på att det just är IT produkter? Kunden vet kanske inte hur man ska kvalitetsbedöma IT?

Kanske.

5.

1 pers, en vecka till 20 pers i ett år. 3-4 på ett år.

6 & 7.

Att vi ska ha nöjda kunder, att Företaget ska vara en bra arbetsplats, vi ska ha få fel i det vi levererar och finns det fel så ska vi peka ut dom för kunden, vi ska leverera rätt kompetens och vi ska leverera i tid.



8.

Vissa kunder har krav på att vi har ett fungerande kvalitetssystem. Egentligen inget konkurrensmedel i sig men stämpeln är det. Men för oss har det även vuxit fram internt att vi haft ett behov av ett kvalitetssystem. Framst har det varit internt, att vi känt att i och med att systemen vi levererar blivit större och att vi som företag har vuxit så behöver man ett kvalitetssystem i ryggen så att vi sätter vår stämpel på vårt arbete.

9.

Ja absolut. Vi är ett företag som satsar mycket på kvalitet och att personalen har hög kompetens (vilket kostar) och det är inte alltid kunderna förstår varför vi är dyrare än källarfirmen. Det kan till och med vara så att dom vill skippa vissa tester bara vi levererar i tid. De kunder som själva har börjat fundera kring kvalitet förstår att det kostar men att det kostar ännu mer att inte ha kvalitet.

10.

Först får vi en förfrågan och då tillsätter vi en förstudie för att ta reda på vilka dom egentliga kraven är. Detta ska syfta till att ta fram en kravspecifikation, förutom kravspecifikationen tar man även fram en grov projekt- och tidsplan för att bedöma personal- och tidåtgång. Efter det kan man gå in i offereringsfas. Efter det kommer själva projektarbetet igång då man bryter ned kraven i delsystem och delenheter och då fastlägger man projektplanen.

Rapporterar man kontinuerligt och bockar av checklistor etc?

Det man gör är att man använder en konstruktionschecklista där man vartefter man granskar konstruktionen bockar av. Samtidigt har man en projektmodell med ett antal milstolpar som måste ha nåtts för att man ska få gå vidare till nästa fas. De här milstolparna gås igenom med representanter från ledningen.

11.

Ja. Startmöte där projektledaren presenterar projektplanen. Kravspecificeringsfas. Kravspecifikationen granskas. Systemkonstruktion som delar upp systemet i mindre delar, systemspecifikation. Konstruktionsspecifikation. Därefter själva detaljkonstruktionen. Tidigt påbörjas ett verifieringsarbete som avser hur man ska verifiera tester och testsystem. Arbetet fortgår genom en iterativ process då man konstruerar och verifierar vartefter man kommer framåt mot slutmålet, slutverifiering och leverans.

12.

Företagets egen som naturligtvis är påverkad av våra erfarenheter från olika håll.

13.

Det man kan säga är att vår modell är generisk och den kan vi applicera på vilket projekt som helst. Det är specifikationerna och checklisterna som förändras. Är det t.ex. kodning i C så måste vi ha kodningsregler för C. Faserna i ett utvecklingsprojekt är alltid samma men dokumenten i respektive fas är olika p.g.a. projektets art.

14.

Projektledaren samordnar och håller koll, men i och med att teamen består av 3-5 man så är det inte så att Pelle gör en del som han testat och sedan har han gjort sitt utan när helheten testas är alla med tillsammans med de andra. Vi har inte så strikta ansvarsgränser utan det



är mera dynamiskt. Det ansvaret man har som konstruktör är att man levererar alltid en testad delprodukt.

15.

Testerna sker fortlöpande på många olika nivåer. De system som vi levererar utvecklar vi i en värdmiljö som vi sedan laddar ner på en målmiljö. När man sitter i värdmiljön är det lättare att simulera testdata, här gör man alltså en typ av test. Sen lägger man ner programvaran på annan hårdvara och då går vi in i en integrationsfas som går ut på att få hård- och mjukvara att fungera ihop. Efter det går vi in i en verifieringsfas då man testat enligt en specifikation. Vi jobbar med inkrementella integrationsdrivna projekt vilket fungerar så att vi försöker hitta minsta gemensamma nämnare. Vilken är den minsta funktionaliteten som måste finnas i de olika delsystemen för att få ett system som kan snurra lite? Man försöker få en låg funktionalitet som fungerar för att få en plattform där man kan få en uppfattning om det går att koppla ihop systemet som man tänkt sig. Därefter lägger man till funktioner på de olika delsystemen som man testat och vartefter bygger man vidare tills man får ett fullständigt system.

16.

Det beror på för vi arbetar med många olika typer av system. Vi utvecklar en del testsystem och i dom fallen är usability viktigt så att även andra än tekniker kan förstå systemet. Vi har t.ex. regler för hur man bygger upp ett GUI, vilka färger etc. Man diskuterar med användaren hur GUI't ska utformas. Efficiency kriterierna testat där det är viktigt. I vissa sammanhang vet vi att vi har ett väl tilltaget standardminnesutrymme och då behöver vi inte göra minneseffektiv kod, då är det helt onödigt att lägga tid på att optimera koden. För oss är det kunden som styr de här faktorerna. Säger kunden att det ska vara flyttbart till 15 olika plattformar så får vi göra det!! Vad det gäller maintainability så har vi kodningsregler för hur programmet ska se ut men om vi gör ett jobb för Ericsson så kodar vi efter deras regler för det är dom som tar över produkten och det är dom som måste kunna underhålla systemet.

Vi gör alltså en krav- eller verifieringsmatris där alla krav radas upp och för varje krav finns ett testfall och för varje testfall skriver vi en testspecifikation som beskriver varför, vilket är kravet, hur testat vi. Under testen fyller man i ett testprotokoll med vilken programvara, hårdvara, när, vem, godkänt ja/nej.

19.

Vi har börjat med att titta på mått på hur bra själva kvalitetsmetoden är. T.ex. hur många fel hade vi innan vi testade och hur många fel har vi hittat m.h.a. testet.

1.

Regionansvarig för Företaget. Vi sysslar med programutveckling och elektronikutveckling, realtidssystem och inbäddadesystem och det faller på mitt ansvar att vår avdelning producerar resultat med kvalitet. Till stöd för detta har vi ett regelverk.

2.

En stor del av vår personal jobbar som resursförstärkning ute hos kund och lyder då under deras kvalitetssystem. Dom ska då ha med sig ett papper ut med vissa punkter att följa men i övrigt kundens system som gäller. I den egna utvecklingen har vi ansvarsprojekt där vi implementerar vårt kvalitetssystem.



4.

Andemeningen är ofta den samma. Man kan ställa olika krav på dokumentation, hur saker och ting ska göras eller i vilken ordning.

Kan det vara så att kunderna inte är så IT medvetna att dom inte kan ställa rätt krav?

Det kan hända men det beror på varför dom har tagit in oss. Går vi in som resursförstärkning i en befintlig organisation som dessutom kanske är ganska mogen (Ericsson, Volvo) då har dom en god medvetenhet. Men går vi in som kompetensförstärkning i ett företag som inte har detta som kärnverksamhet då vet dom inte heller vad som behövs för att kvalitetssäkra en IT-tjänst.

5.

Inhouseprojekt sträcker sig från 1 till 2 manmånader till 3manår.

6.

Företagets uttalade policy är att kvalitetstänkandet är väldigt viktigt. Det ska genomsyra verksamheten och vi har en kvalitetschef i koncernen. Vi har internrevisioner och vi är ISO-certifierade.

7.

Det kan finnas flera mål, det är inte alltid uppenbart, många företag drog igång sitt kvalitetsarbete efter kundkrav, man har med andra ord inte riktigt begripit riktigt varför man ska göra det men kunden har sagt att ni får inte vara med och offerera det här om ni inte kan visa att ni har ett fungerande kvalitetssystem (helst ISO cert). Det är väldigt viktigt för i slutändan kommer man att tjäna pengar på att leverera rätt produkter från början.

8.

Det är självklart även ett konkurrensmedel. Sitter man med likvärdiga anbud från olika leverantörer så slår ju den som kan visa på bättre dokumenterad kvalitet ut den andre det är jag övertygad om. Dels det och dels så gör ett bra kvalitetssystem att man lyckas bättre, det bör vara så men det är inte säkert. Det gäller att ha kvalitetssystem som stöder verksamheten och inte motarbetar den vilket lätt kan bli fallet. Du får ett tungrott kvalitetssystem som tynger ner verksamheten som gör att det blir dyrare att producera saker som i slut ändan inte ger ett bättre resultat.

9.

Man kan inte säga till en kund att med vårt kvalitetsarbete vill vi ha mer betalt. Däremot så inser alla att det arvodet man tar inberäknar alla kostnader man har.

Är det så att ni har högre kvalitetsambitioner än kunden?

Arbetar man ute hos kund är det inga problem. Om kunden vill belägga konsulten med kvalitetsarbete är det hans ensak. Och inhouse projekt har kvalitetsarbetet inbakat i priset.

Det ska gå fort och som IT leverantör ges man kanske inte tiden som behövs?

Ja, det är nästan alltid på det viset. Man har ett slutdatum men förhandlingar drar ut på tiden så startdatumet skjuts fram men ej slutdatumet. Det är klart att börjar man skära så är det risk för att dokumentationen blir lidande. Men gör man rätt från början så kräver kvalitetsarbetet inte mer tid.

10.



Det finns olika faser. Vi har försäljningsprocessen som mynnar ut i en ordermottagningsprocess, den ska dokumenteras enligt en instruktion. Man gör kalkyl, projektplan vilket resulterar i offerten. Acceptans från kund. Projektet startar, projektledare utses. Gör en detaljerad projekt plan. Projektet har ett antal faser och det finns beskrivet vilka in och ut dokument som finns vid de olika faserna innan man får gå vidare.

11, 12.

Vi har ett antal arbetsinstruktioner som finns beskrivna i processbeskrivningar. Sen finns det även processtyrnings metoder. Företaget har en projektstyrningsmodell som kallas Ratten som kommer att verka i två till tre nivåer beroende på projektstorlek. Ratten togs fram av Företaget för att få ett enhetligt sätt att styra projekten.

13.

Beroende på uppgiften är det naturligtvis olika dokument som ska fram. Några basdokument behandlas i alla projekt dom pekar sen i sin tur på specifika dokument som tas fram för dom enskilda projekten.

Kan det vara olika kvalitetsbegrepp som blir olika viktiga så man mäter olika saker?

Inför varje projekt så skriver du en verifieringsspecifikation i kravspecifikationen och i den kravspecifikationen finns ett antal punkter som kunden sagt att han vill ha testade. Utifrån verifieringsspecifikationen ska man prova vad man gjort. Har du uppnått rätt kvalitet ska du kunna bocka för dom här sakerna.

14.

I ett hyfsat stort projekt har man definierat olika roller. Det är projektledaren, projektadministratör, kvalitetsansvarig, ansvariga för olika delar som programvara, hårdvara etc. Då är man ansvarig för sin del och har rapporteringsskyldighet.

15.

Fortlöpande har man kodgenomgångar. När man gjort klart en modul lämnar man över kodsnutten till någon annan som går igenom koden. Så gör man på kodavsnittet. När man har hela paketet ska man göra en formell kodgranskning. Tidigt i projektet så sker granskningar när man sedan är färdig med projektet skriver man en testplan. Testplanen är detaljerad och beskriver vilka uppkopplingar man ska göra och vilka instrument som ska användas.

16.

Det vi har minst krav på här är portability. Mjukvara skrivet för ett inbäddat system är inte portabelt. Dom övriga känner jag igen. Funktionalitet det måste man alltid uppfylla det mäts på så vis att man ser till kravuppfyllelse. I den flightrecorder som vi byggt testar man att kommunikationen fungerar och att rätt data hämtas in.

19.

Erfarenhetsåterföring. Kunden ska vara väl insatt i förutsättningarna innan man går in i projektet. Har kunden ingen kravspecifikation får vi hjälpa till att ta fram en så att man kan var överrens när projektet är avslutat. Kunden sitter med i styrgruppen som beslutar under projektets gång.

1.



Jag jobbar med verksamhetsutveckling, hjälper företag att ta fram förbättringsprogram och strategier för hur företag ska kunna förbättra sina programvaruprodukter. Tittar på utvecklingsmodeller för hur man tar fram programvara (produktnära). Projektstyrningsmodeller, affärsmodeller.

2.

Både och.

3.

När vi tar hem ett projekt och levererar en lösning då kör vi med våra egna metoder och modeller. Det är vår argumentation mot kunden så dom vet att dom får en kvalitetssäkrad produkt. När vi deltar i projekt ute hos kund så får vi använda dom processer som kunden har men ser vi att det brister så föreslår vi förändringar.

4.

Det beror på vad man gör åt kunden men så kan det vara, ja.

5.

Vi har haft två tunga projekt i huset. 10-15 pers på två års tid. Mindre projekt på under ett halvår är oftast mer specifika eller delar av system.

6.

Tre nivåer (se papper) verksamheten, projektet och produkten. Då säger vi att det krävs kvalitetsrutiner i alla de här nivåerna för att produkten, programvaran ska bli kvalitetssäkrad. Det räcker inte att bara ha en utvecklingsmodell utan du måste ha en projektstyrningsmodell som fungerar och du måste ha roller, ansvars befogenheter och beskrivningar i ett verksamhetssystem.

7.

Nöjd kund, att lösningen ska ge effekter hos kundens verksamhet.

8.

Framförallt konkurrensmedel. Kunder kan ha en diffus uppfattning om att IT kan ha effekter på andra delar av deras verksamhet. När man inför IT lösningar kanske man måste förändra arbetssätten för att få ut den effekten.

9.

Mja! Det går till så här att vi får någon form av förfrågan som vi går vidare med och analyserar mer ingående vilka funktionella och tekniska krav dom har. Sen när vi gör presentationen så talar vi om vad vi tycker är kvalitet och varför vi tycker att dom ska göra på ett visst sätt. Vissa kunder köper inte det men i och med att vi talar om det och att vi bygger in det i våra offerter och åtaganden så beskriver vi exakt vad det är kunden vill ha ut. Ofta kan det vara så att man hålls ansvarig för något som inte var med i kalkylen från början. Det är viktigt att redan i offerten reda ut detta.

10.

Först tittar vi på kravspecifikationen från kunden och ser om dom täcker in allt, om dom är tillräckligt tydliga det är ett sätt att säkra kvaliteten att vi är överrens om vad som tekniskt sett ska göras. Sen tittar vi på vilken typ av kompetenser som vi behöver för just den här lösningen och gör en gap analys mot vår befintliga kompetens. Sen tittar vi på



utvecklingsmodellen, vad är det för typ av lösning, är det känd teknik eller är det ny teknik, har vi gjort det innan eller aldrig då får vi anpassa utvecklingsmodellen efter detta. Sen har vi vår projektstyrningsmodell som vi väver in med beslutspunkter i detta.

Löper kvalitetsarbetet integrerat med projektet?

Ja, och vi ser helst att även verksamhetsdelarna löper parallellt så att de som ska använda systemet är med på spåret. Vi vill att kunden är med och fattar beslut under projektets gång.

11.

Ja, PBS för projektstyrning och RUP (Rational Unified Process) på själva systemutvecklingsmodellen.

12.

PBS är helt igenom Företagets egen. RUP kommer från Rational och är en kommersiell produkt för inkrementell och iterativ utveckling.

13.

Ja för en databas, till exempel, blir det mera test, med fokus på det systemtestdelar. Läger du på ett användargränssnitt då användartestas och usabilitytestas detta. Vi utveckling av intranät är usability ännu viktigare plus kanske säkerhetsaspekten. Det kan vara olika ja. Den första analysen, att ta reda på vad det är för lösning, får styra upplägget.

14.

Projektledarens roll är att tillgodose kravspecifikationen. Vi har personer som ansvarar för testarbetet. Sen har vi dom som gör designen som ansvarar för sin del. Utöver det lägger vi till granskningstillfällen. Huvudsaken är att det inte blir vattenfallsmetoden där en grupp utvärderar kravspecifikationen och lämnar över till designgruppen som gör sitt som lämnar över till konstruktionsgruppen etc.

15.

Fortlöpande hela tiden. Det kan vara på en hög abstraktionsnivå i RUP. Det kan vara "test" där man stämmer av exempelvis att det är den och den användaren som matar in dom värdena, och utför dom transaktionerna.

16.

Konstruktionsnära har vi reliability och portability. Vi försöker täcka in alla dom här i kravspecifikationen så det här ingår i kravanalysen. Så här gör vi: kravunderlaget från kunden är i många fall uruselt därför försöker vi sälja in en förstudie för att ta reda på vad kunden vill. Då täcker vi in dom här områdena. Vi hittar kravställare hos kunden som kan tänkas ha krav inom exempelvis maintainability. Finns det en förvaltningsfunktion hos kunden som ska ta hand om det levererade systemet så finns kanske där någon som kan sätta upp kraven för maintainability. Det som kunden oftast formulerat i kravspecifikationen är funktionaliteten.

Det är alltså där dom sett sitt behov?

Japp. Då är det upp till oss att påpeka att även de andra faktorerna är viktiga.

17.

Specifika mätdata tar vi bara fram på vissa av faktorerna/kriterierna. Det beror på vilka lösningar vi sysslar med. Ex: hade vi gjort ett realtidssystem så hade vi fokuserat mer på



vissa saker, här vi gör inte det utan det är mer antal användare systemet ska klara av, svarstider.

Det blir väldigt systemlösningsspecifikt vad man trycker på?

Det är helt rätt att de här faktorerna måste man titta på när man gör sin kravanalys. Man måste hitta kravställare inom varje område så att man inte glömmer bort det. När det gäller usability är det väldigt viktigt.

Exempel på mått?

Användargränssnitt testas på användaren. Ett mått kan vara hur många knapptryckningar eller tabbar hon måste göra för att få sitt svar. Hur många inmatningar på hur många ställen. Ska ta hänsyn till erfarenheten hos användaren (snabbfunktioner)? Användaren ska inte behöva komma ihåg data från en skärm till en annan. Den typen av mått kan man sätta en siffra på. Resource och time efficiency kan man också ta fram mått på ganska lätt. Portability har med designen att göra och är inte lika lätt att mäta särskilt inte nu när man bygger mycket kring färdiga windows. Det kan ligga något inbyggt fel som gör att man kan köra på PC men inte Unix. Kunden är inte så intresserad av uppfyllande av mätdata han tittar mer på funktionalitetsdelen därför har inte vi så mycket mätdata. Men på Ericsson där jag är är man väldigt fokuserad på mätdata.

Reliability har det forskats mycket kring.

Vi gör i huvudsak affärslösningar så för oss är reliability t.ex. på en bank transaktionssäkerheten. Man får anpassa dom här måtten beroende på vilken sorts lösning man gör. Feltolerans, användaren kommer att göra fel så hantering av sånt är viktigt. På Volvo har man styrsystem där fel kan skapa personskada.

19.

Min erfarenhet är att man på företag tror att det räcker att man har en utvecklingsprocess. Att man säkrar krav, analys och design och att man kör tester. Då glömmer man att man måste ha en nivå närmast ovanför projektet. Att projektet måste styras i enlighet med den här modellen att det finns en projektstyrningsmodell och sen i sin tur så påverkas projektet av en affärsprocess som hör till verksamhetsdelen. Ericsson t.ex.: Dom får en order på ett stort operatörsystem, dom får in kravspecifikationen, gör en analys och presenterar tid och kostnad. Kunden vill att det ska gå på halva tiden. Då är det stor risk att man går i fällan och säger OK om det från ledningshåll finns brister i förståelsen för allt som påverkar utvecklingsprocessen. Då får man kvalitetsproblem. Det finns alltså processer i omgivningen som påverkar utfallets kvalitet. Det är viktigt att företaget har den helhetssynen och förståelsen. Det kan ju vara så att kunden inte köper vår projektstyrningsmodell med avstämningar och granskningar. Då är det viktigt att det finns med i åtagandet av projektet. Det kan vara lite si och så med förståelsen hos kunden.

ISO är mycket att men inte hur man ska gå tillväga?

Vi som arbetar med programvarukvalitet tittar inte så mycket på ISO. CMM och SPICE som är mer kopplad till programvaran och dess egenskaper. Vi har något vi kallar för integrerade kvalitetssystem där vi säger att det räcker inte att titta på X utan vi måste bygga hela vägen ner för att ha en helhetssyn där olika personer fokuserar på olika delar och där jag är utvecklingsnära, projektet och produkten.