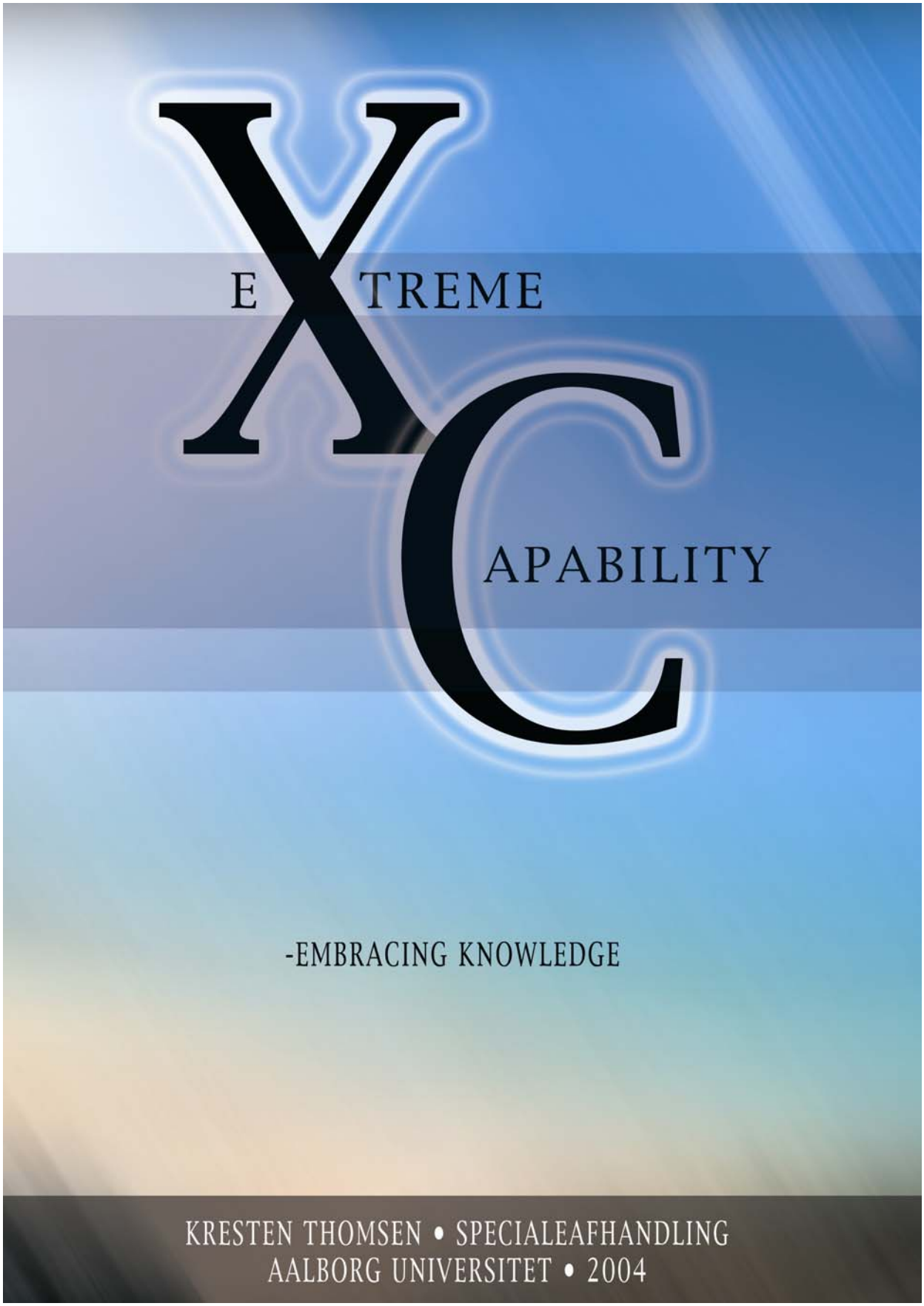

EXTREME
CAPABILITY
-EMBRACING KNOWLEDGE
KRESTEN THOMSEN • SPECIALEAFHANDLING
AALBORG UNIVERSITET • 2004

Synopsis

Indeværende speciale diskuterer både teoretisk og praktisk, i hvilken udstrækning softwareudviklingsmetodikken *Extreme Programming* (XP) kan anvendes til at supportere de organisatoriske beslutninger, der er medvirkende til at konstituere softwareorganisationen på et givent niveau i *The Capability Maturity Model* (CMM modellen). På baggrund af den teoretiske og praktiske analyse er konklusionen, at organisationer baseret på Extreme Programming er i stand til at opnå en certificering på CMM niveau 3, men at det ikke er muligt for organisationen at blive certificeret på højere niveauer uden at kompromittere de grundlæggende værdier i Extreme Programming. Specialet diskuterer videre om intensionen med CMM modellen kan efterleves med XP som organisatorisk redskab, og konklusionen er, at XP i kraft af sin eksplicitte videndeling og anvendelse af tillid til medarbejderne er i stand til at opfylde intensionen med CMM niveau 5 uden dog at kunne opnå en certificering på dette niveau. Det konkluderes endvidere, at organisationen gennem koblingen af anvendelsen af XP i forhold til CMM modellen kan opnå *Extreme Capabililty*, en konstrueret betegnelse for organisationer, der kommunikativt udnytter viden og omskiftelighed både i softwareudviklingen og på et organisatorisk niveau.

Takketale

I forbindelse med udarbejdelsen af dette speciale skal der først og fremmest lyde en hjertelig tak til min vejleder Per Hasle, som i den grad har bidraget til særdeles berigende diskussioner, som jeg har nydt meget og kommer til at savne. Du har i den grad været uvurderlig. En stor tak skal også lyde til min kæreste Marie Bjerring, som har overlevet disse specialemåneder og stadig nærer en dyb og hjertelig kærlighed til mig. Du skal vide, at det er gengældt på højeste niveau. Mine kolleger i Dreamhouse skal vide, at jeres bidrag til både speciale og forretning har gjort mig mere viis og især Bjørn Nedergård Nielsen har været en særdeles givende sparringspartner omkring forretningsdelen (og alt det andet vi har diskuteret). En tak skal også lyde til den projektgruppe, som har bidraget til de eksperimenter jeg har fundet nødvendige for at kunne udtale mig praktisk i dette speciale. Det drejer sig om Dennis Nørgaard, Jonas Dinesen, Lars Byrialsen, Nikolaj Dam Østergaard og Nina Lilholt, alle fra 6. semester humanistisk datalogi på Aalborg universitet.

Aalborg den 22.juni 2004

_____

Kresten Thomsen

Specialet er sat med Garamond str. 11 og fylder 234.343 tegn, svarende til 97, 64 normalsider eksklusiv bilag mm.

English Summary

When one door closes another door opens;
but we so often look so long and so regretfully upon the
closed door, that we do not see the ones which are open for us.
- Alexander Graham Bell

The choice of software development methodology depends largely on the organizational foundation in any software company around the world. The choice of methodology reflects the organizational focus, which can be traced back to the choices of the individual managers in the organization. Some software organizations use the Capability Maturity Model (CMM model) as an effective certification tool to ensure a stable software development environment, and this certification is used to create an advantage and a distance to other software development organizations, and thereby create a greater chance of success in contract negotiations. The CMM model is basically divided into 5 levels which are based on an inclusion principle where the lower levels are included in the upper levels. CMM level 3 therefore includes CMM level 2 and CMM level 1.

The focus of the master's thesis is a reflection on the CMM model in the light of the software development methodology Extreme Programming (XP). The thesis focuses on the possibility of using XP as an organizational tool to support the organizational decisions, which appoint the organization to a certain CMM level. During the theoretical analysis I conclude that XP can support CMM level 3 requirements except from two Key Process Areas (KPA), which are not included as a focused area in XP. In the practical experiment I address these two areas, and based on the empirical data I collected during the cooperation with the 5 students from Aalborg University, I conclude that it is possible to contain these two KPA without compromising the fundamental values in XP. This means that it is indeed possible to certify an organization based on XP at CMM level 3.

But the master's thesis goes deeper than CMM level 3, and the theoretical analysis concludes that an organization based on XP has no chance of getting certified on CMM level 4 or level 5, because the CMM model implies that the organization applies quantifiable tools in order to quantitatively control both developers and the software developed. This conflicts with the fundamental values in XP, and therefore the master's thesis discusses the strategic role of the CMM model further to ensure that no visible stone is unturned.

The discussion in the master's thesis views the CMM model from two angles: 1. as a certifying tool to contain all KPA at all levels and thereby fulfil the certifying intention of the CMM model. 2. as an organization development tool to ensure that the organization provides stable and quality software, but also to gear the organization to an ever-changing world with shifting standards and demands and thereby fulfilling the organizational intention of the CMM model. During the discussion it is made clear that the certifying intention in the CMM model transfers the responsibility and the trust from the employers by using the quantitative tools in the CMM model. The transfer is very problematic to the humanistic perspective of system development, but the quantitative tools ensure measurable parameters that enable the management to set priorities and fulfill

these priorities. The downside is that the personal dedication from the employers most likely is missing, and this being a fact does not necessarily create quality software or an adaptive organization as intended in the CMM model.

The discussion concludes that XP can be used to fulfil the organizational intention of the CMM model without using quantitative tools. The CMM certification is of course impossible without quantitative tools, but the master's thesis concludes that by using qualitative tools XP ensures that the intention of the KPA at level 5 is fulfilled.

The practical experiment provides the evidence that an organization can fulfil all KPA at level 3, but this evidence is based entirely on the preliminary study of a software development process and not on an entire development process, which would be preferred. Simply by using XP in the context of the CMM model it is most likely for an organization to achieve a certification at CMM level 3 and at the same time fulfil the organizational intention at CMM level 5. Of course the organization needs to develop new communicative tools to ensure verifiable decisions at an organizational level, but this is an area that the master's thesis does not address. This however being a highly interesting area.

Indholdsfortegnelse

Indledning

"I have a cunning plan."
Baldrick, Blackadder

Softwareudvikling som organisatorisk disciplin er et stærk omdiskuteret område, specielt indenfor indeværende uddannelsesretning; humanistisk datalogi. Området er specielt, idet litteraturen på området som oftest enten beskæftiger sig med organisationen eller med softwareudvikling og derfor kun sporadisk kobler de to discipliner. Der er dog discipliner i dette område, som ser softwareudvikling som organisationsudvikling og der er koblinger i dette område, som kan minde om specialets fokus, men litteraturen er særdeles begrænset, hvorfor specialet retter et eklektisk blik på softwareudvikling i et organisatorisk perspektiv, men samtidig forsøger at koble softwareudviklingsteorier og organisationsudviklingsteorier i et praktisk orienteret perspektiv. Specialet kan derfor læses af de, der har en interesse for softwareudvikling og samtidig for de, der har en interesse for den organisation, som opstiller rammerne for softwareudviklingen; den kontekst i hvilken softwareudviklingen agerer.

Et af målene med at udvikle software er at tilgodese brugernes behov og gerne skabe profit ved at tilfredsstille disse behov. Indgangsvinklen til softwareudvikling kan tage mange former, såsom autodidakt og selvledende, regelstyret og ligefrem, tilfældig men sammenhængende og til sidst smidig udvikling, som dette speciale vil rette fokus imod. Disse softwareudviklingsmetoder kan spores tilbage til et metodisk grundlag med fundament i lærebøger eller kulturer på de enkelte områder, og alle indeholder både mange fordele og ulemper, som organisationen bør forholde sig til før det endelige organisatoriske valg af softwareudviklingsmetode. Et af målene har alle metoder dog til fælles: at udvikle (funktionel) software til brugere.

Men ud over valget af softwareudviklingsmetode ligger der også et organisatorisk perspektiv bag, som skal håndtere de processer, der konstituerer og institutionaliserer en del af softwareudviklingen. Det er bl.a. kundekontakt, markedsføring, tilføjelser til software, videreudvikling mm. Derfor er der et interessant perspektiv i at overveje sondringen mellem organisationen og valget af softwareudviklingsmetode, da de hænger uløseligt sammen; et perspektiv som indeværende speciale vil forfølge.

Der er imidlertid stor forskel på traditionelle organisationer og softwareorganisationer, da softwareudviklingsorganisationer er kendetegnet ved en meget begrænset masse med forholdsvis begrænset erfaring i forhold til traditionelle organisationer. Softwareindustrien har jo kun eksisteret i få årtier (eksempelvis IBM) og litteraturen omkring softwareorganisationer har kun eksisteret i en begrænset årrække i modsætning til traditionel organisationslitteratur. Derfor vil dette speciale primært omhandle softwareorganisationer ud fra et operationelt perspektiv, frem for et teoretisk og generaliserende organisatorisk perspektiv.

Specialet vil grundlæggende have et operationelt sigte til brug i både specialet og i forbindelse med opstart af egen virksomhed. Dermed er der to fokuser i specialet; et afklarende fokus omkring specialets teoretiske indhold og muligheder, hvormed der kan opstilles et grundlæggende analyseapparat, samt et operationelt fokus,

som tillader direkte anvendelse af teoriens pointer i en reel softwareudviklingsproces og en reel organisationsopbygning, som ekspliciteres yderligere i kapitlet omkring det praktiske eksperiment på side 73.

Men når man taler softwareorganisationer, så taler man som regel også kvalitet i den udviklede software. I lyset af denne problemstilling indførte det amerikanske militær en model, hvormed man kvantitativt kunne validere og måle kvaliteten i en softwareorganisations produktlinje og dermed sikre sig, at forsvaret kun skrev kontrakter med softwareorganisationer placeret på bestemte niveauer i denne model og dermed kendetegnede en stabil udviklingsproces med den efterspurgte kvalitet. Modellen blev videreudviklet af Software Engineering Institute (SEI) og betegnet *Capability Maturity Model* (herefter CMM modellen), og vil i dette speciale blive anvendt til at forholde kvaliteten i den organisatoriske del af softwareorganisationen til valget af softwareudviklingsmetode. CMM modellen vil være den guideline, som muliggør en videre analyse.

CMM modellen er kort fortalt inddelt i fem niveauer, hvor niveau 1 er det laveste niveau og niveau 5 det højeste. Hvert niveau indeholder en række *Key Proces Areas* (herefter KPA), der skal opfyldes og verificeres, før organisationen kan certificeres på et bestemt niveau. Niveauerne i CMM modellen er opbygget som et inklusionssystem, hvor nedenstående niveauers KPA er fuldstændigt indeholdt i ovenstående niveauer. De fem niveauers samlede KPA er vedlagt i bilag 2, og indeværende speciale vil analysere alle disse KPA relateret til de forskellige niveauer dels for at blive fortrolig med CMM modellen, dels for at skabe et udførligt grundlag til specialets endelige konklusion.

Der er altså en interessant kobling mellem softwareorganisationen, CMM modellen og softwareudviklingsmetoden. For at iagttage disse sammenhænge, er det fordelagtigt at observere en organisation, som arbejder med disse interessante emner og samtidig er indstillet på et indgående samarbejde, hvor metoder kan afprøves og udviklingskulturen sættes på prøve. Særdeles få organisationer er villige til et så eksperimenterende samarbejde, hvorfor jeg har konstateret, at det har været nødvendigt at oprette en organisation til netop dette formål, hvormed det vil være muligt at afprøve teoriens pointer i praksis. Intensionen med oprettelsen af organisationen har været at efterprøve en softwareudviklingsmetode i lyset af CMM modellen i form af et praktisk eksperiment. Imidlertid har dette eksperiment været afhængigt af udviklingskapital, som desværre har været væsentligt sværere at tilvejebringe end først antaget. Derfor er organisationen en realitet, mens den reelle udvikling ikke har været mulig at igangsætte, idet det ikke har været muligt at indhente kapital, som kunne betale lønninger, udstyr og administration. Derfor er eksperimentet begrænset til forundersøgelser og organisationsopbygning, hvor det ønskelige eksperiment om reel udvikling ikke vil være til stede i den udstrækning, som det har været intenderet fra begyndelsen.

Sammenhængen mellem udviklingsmetoden og organisationen vil blive belyst i indeværende speciale, hvor softwareudviklingsmetoden *Extreme Programming* (herefter XP) vil anvendes i en reel organisation, som blev etableret 5. januar 2004. Jeg har observeret, at systemudviklingsmetodikken XP er en agile systemudviklingsmetodik, hvilket betyder, at den ændres, udbygges og vedligeholdes i forhold til det handlingsrum, i hvilken den udfoldes/anvendes. XP er valgt som udviklingsmetode, da de indledende observationer tyder på, at der er en sammenhæng mellem netop denne agile udviklingsmetode og de niveauer, som CMM modellen inddeler

softwareorganisationer i. CMM modellen er valgt på baggrund af dens universelle måleegenskaber, som er vidt udbredt i softwareorganisationer og fordi modellen er bredt dokumenteret og verificeret.

Specialet vil i modsætning til andre specialer indeholde en todelt konklusion; en teoretisk konklusion, som udelukkende vurderer de teoretiske muligheder og en praktisk konklusion, som både diskuterer og anvender teoriens antagelser i en reel organisation i den udstrækning det har været muligt indenfor specialets tidsrum og indenfor de rammer som den manglende udviklingskapital har medført. Dermed er forhåbningen med dette speciale at skabe grobund for en anvendelig organisatorisk metodik baseret på Extreme Programming; en metodik, der forhåbentlig kan anvendes i virksomheder til at muliggøre en relativ hurtig certificering iht. CMM modellen. Hvorvidt dette er muligt vil afsløres i konklusionen.

Anvendelsen af CMM modellen som organisatorisk referenceramme virker ligeledes kompleksitetsreducerende på flere niveauer. For det første gør anvendelsen af CMM modellen det muligt at operationalisere de institutionelle funktioner og områder, som ifølge CMM modellen kvalificerer organisationer til at eksistere på bestemte niveauer. Derudover tjener CMM modellen som generel operationalisering af den teoretiske organisatoriske beskrivelse af organisationer og beslutninger. Der er kort sagt noget konkret at forholde sig til, så specialet i sidste ende får mulighed for reelt at kunne konkludere både teoretisk og operationelt.

Problemformuleringen bliver derfor:

> *I hvilken udstrækning kan grundideen i Extreme Programming teoretisk og operationelt supportere de organisatoriske beslutninger, der er medvirkende til at konstituere softwareorganisationen på et givent CMM niveau, og i hvilken udstrækning kan Extreme Programming i denne funktion anvendes som organisatorisk værktøj?*

Overvejelserne igennem specialet vil primært være selvreflekterende, forstået på den måde, at der vil være et anvendelsesorienteret underliggende perspektiv, som vil dreje de teoretiske antagelser mod overvejelser omkring den direkte implementering og anvendelse i den etablerede organisation. Det betyder, at der i dette speciale vil være en forholdsvis høj vægtning af teori i forhold til analyse, idet der på kort tid skal kortlægges et komplekst genstandsfelt for dels at blive fortrolig med emnet, dels en tvingende nødvendighed for overhovedet at kunne diskutere og konkludere. Ydermere har opstarten af en reel virksomhed afkrævet særdeles meget arbejde uden om specialets fokus, som ikke direkte kan indeholdes i specialet, men som er en forudsætning for overhovedet at opstarte en virksomhed.

Softwareorganisationen

Foundations do not keep the rain out, but they offer
some small basis for imagining that a roof is possible.
[James G. March, 1994, s. 271]

Dette kapitels primære formål er at redegøre for softwareorganisationens generelle opbygning med henblik på at sætte prædikat på de dele af softwareorganisationen, som anvendes i CMM modellens *Key Process Areas* (KPA), for i analysen at perspektivere de beslutninger, som konstituerer organisationen på et givent CMM niveau, med XP's iboende redskaber og værdier. Der er to distinktioner i den organisationsteoretiske tanke, som er værd at hæfte sig ved, inden den direkte beskrivelse af CMM modellen. Der vil gennem specialet være en klar distinktion mellem proces og produkt, som nødvendigvis er dialektisk forbundet, da produkt forudsætter proces, og proces forudsætter produkt[1]. Denne distinktion vil senere anvendes til at sammenligne XP og CMM modellen med beslutninger som fokusområde, for i den endelige konklusion at drage nytte af dette samspil i relation til organisationen.

For at undersøge, hvorvidt XP kan supportere de organisatoriske beslutninger, der er medvirkende til at konstituere softwareorganisationen, er det særdeles hensigtsmæssigt at iagttage selve softwareorganisationens konstitution, hvorfor specialet kort vil dvæle ved den organisatoriske grundtanke symboliseret ved Niklas Luhmanns systemteori og senere James G. Marchs beslutningsteori.

Organisationen

Luhmanns systemteori er en særdeles omfattende og vidtrækkende teori, der i sin fulde form ville være uanvendelig i dette speciale. Derfor vil dette kapitel kort beskæftige sig med de elementer, som vurderes nødvendige for at konstruere et begrebsapparat til den videre analyse. CMM modellen bygger på et kvantitativt ledelsesmæssigt perspektiv, hvorfor de organisatoriske mindstedele ikke nødvendigvis er indeholdt i denne begrænsede organisatoriske beskrivelse.

Organisationer er i systemteorien defineret ud fra et medlemskab. Enten er man medlem af organisationen, eller også er man ikke. Organisationer må derfor siges at være determineret af beslutninger, idet det besluttes, om nogen er medlem eller ikke. Denne beslutning om medlemskab ligger dog ikke eksplicit hos én person men er mere en organisatorisk konstatering; enten er man medlem, eller også er man ikke [eks. Luhmann, 2000, s. 240]. Organisationer er ikke kun etableret af et medlemskab men også i forhold til funktioner i systemet. Selve funktionssystemet kan repræsenteres ved de forskellige områder, som softwareudvikling generaliseret indeholder, eksempelvis risikohåndtering, kravspecificering, ledelse, videreudvikling o.l. Disse enkeltstående funktionssystemer konstituerer i sig selv ikke organisationen som helhed, men konstituerer det område af organisationen, i hvilken de både eksplicit og implicit er indeholdt; eksempelvis softwareafdelingen. For

[1] Denne sondring har bl.a. Sokrates problematiseret, idet idé og fænomen kan indtage hinandens pladser. Kom ideen om kniven før selve kniven eller blev kniven opfundet før dens anvendelse var tænkt?

softwareorganisationer er de forskellige områder ikke repræsenteret ved en eller flere personer, da personer kan repræsentere flere områder, og derfor anvendes betegnelsen funktionssystemer i stedet for personer i den videre teori og analyse.

Organisationer kan gennem beslutninger skelne medlemmer fra ikke-medlemmer, og organisationer er så at sige konstitueret af kommunikation i form af beslutninger. Kommunikation og dermed beslutninger styres af forholdet mellem system og omverden. Kun fordi systemet er lukket mod sin omverden, kan organisationen forholde sig til den. [March, 1995, s.32] Denne konklusion fremkommer, idet organisationers kommunikation kun kan iagttages udefra gennem beslutninger, og denne beslutningskommunikation producerer og vedligeholder samtlige elementer i en organisation, herunder funktionssystemer.

Men organisationer er ikke alene kausale relationer af kommunikation mellem system og omverden. Organisationer er foranderlige størrelser og er billedligt talt lige så forskellige som snefnug i haven. Nogle gange ignorerer organisationer klare retningslinjer; andre gange efterlever de dem mere stringent end hensigten. Nogle gange skåner de beslutningstagere for de ubehagelige konsekvenser af mere eller mindre tåbelige politikker; andre gange gør de det ikke. Nogle gange står de stille, når det er ønskeligt, at de skal flytte sig; andre gange flytter de sig, når det er ønskeligt, at de skal stå stille [eks. March, 1995, s.31]. Organisationers foranderlighed gør dem i teorien i stand til at tilpasse sig deres omgivelser, og effektiviteten i denne tilpasning afspejles ofte i organisationens ledelse, ganske som i en almindelig softwareudviklingsproces. Men for at iagttage og vurdere denne organisatoriske foranderlighed og organisationens grundlæggende elementer, er det essentielt at opstille kriterier for iagttagelse, hvilket i indeværende speciale er de kriterier, som CMM modellen opstiller. Derfor vil næste afsnit give et kort resume af CMM modellens form, funktion og niveauer, baseret på Pankaj Jalotes *CMM in Practice*.[2]

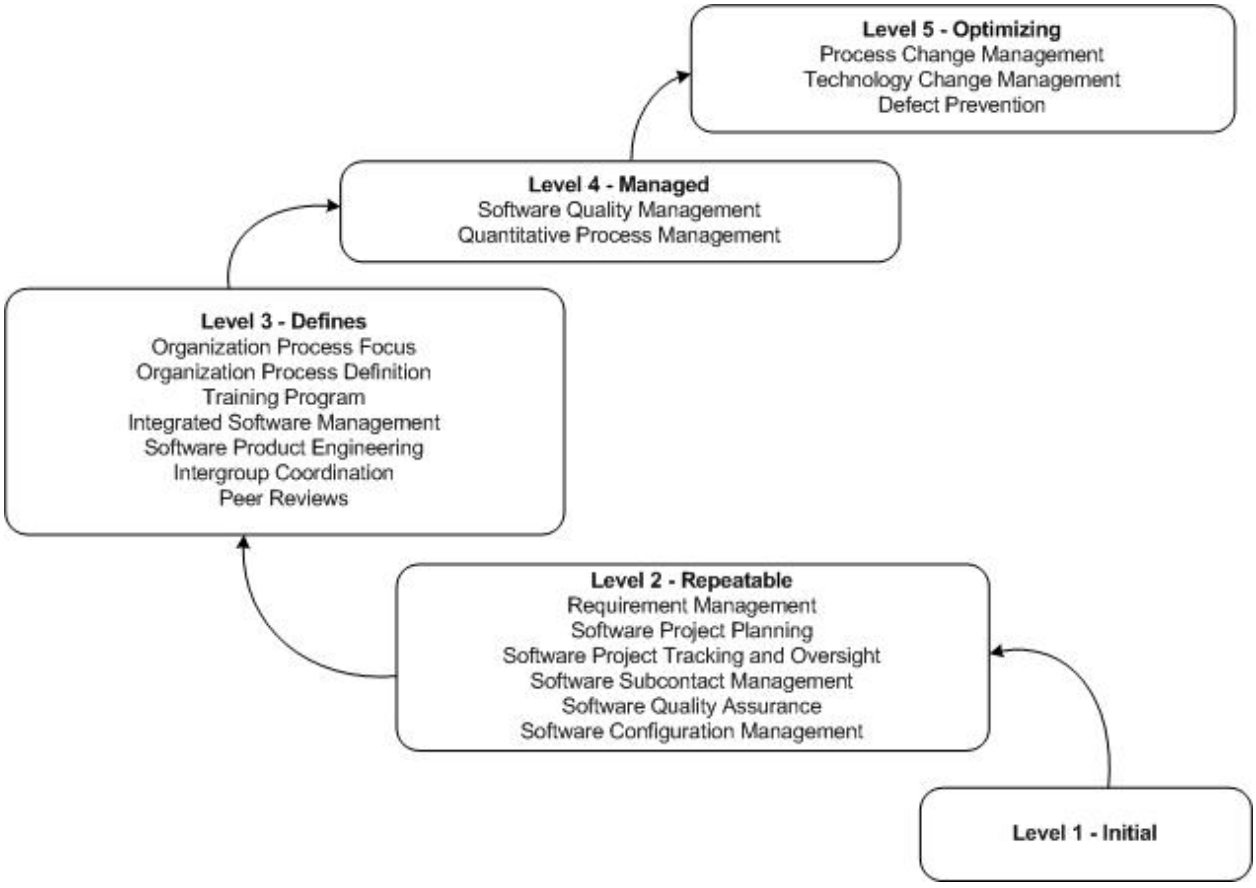
CMM Modellen

CMM modellen er kort fortalt en handlingsramme, hvor softwareorganisationer kvantitativt kan måles og vejes i forhold til opstillede kriterier. Disse kriterier er meget generelle og kan derfor være vanskelige for softwareorganisationer at implementere uden en vis modificering, hvorfor CMM modellen primært fokuserer på to områder indenfor softwareudvikling, hvor resultatet har indflydelse på den økonomiske bundlinje; produkt og produktivitet [Jalote, s.1, McConnell, 1999, s. 69]. Produktområdet fokuserer på kvalitet, modificering og videreudvikling, hvor produktivitetsområdet fokuserer på de organisatoriske elementer som eksempelvis planlægning, management og organisatorisk evne til at omstille processerne.

Udgangspunktet for CMM modellen er, at alle softwareorganisationer har et iboende ledelsesmæssigt krav om måling af både produkt og produktivitet, og CMM modellen tilbyder derfor målbare krav og metoder med

[2] Anvendelsen af Jalotes bog indeholder en personlig anekdote, da bogen blev bestilt gennem Amazon.co.uk og derefter blev sendt fra Kina. Indholdsmæssigt er bogen identisk med den oprindelige bog og de kinesiske myndigheder har selv trykt bogen (Higher Education Press), men har af uvisse årsager undladt årstallet. Når der henvises til Jalote anvendes der således kun navn og sidenummer, hvilket er en venlig hilsen til både anekdoten og Higher Edication Press, såfremt de måtte læse dette.

henblik på senere implementering i organisationen. CMM modellen er inddelt i 5 niveauer, hvor det første niveau indeholder alle softwareproducerende organisationer, da der ikke er specifikke krav til at eksistere på dette niveau, og niveau 5 indeholder ganske få softwareorganisationer på verdensplan, da kravene hertil nærmest kræver en organisation i organisationen for at håndtere denne større udviklingsmæssige kompleksitet, som CMM modellen medfører. Niveauerne i CMM modellen er, som nævnt i indledningen, bygget på et inklusionsprincip, som illustreret med figur 2, hvor nedenstående niveauer indeholdes i ovenstående niveauer. Organisationer på CMM niveau 4 indeholder dermed alle krav for niveau 1, 2, 3 og 4, hvilket er kravet for at kunne certificeres som værende på CMM niveau 4. Spændet mellem de 5 niveauer illustreres i figur 1.

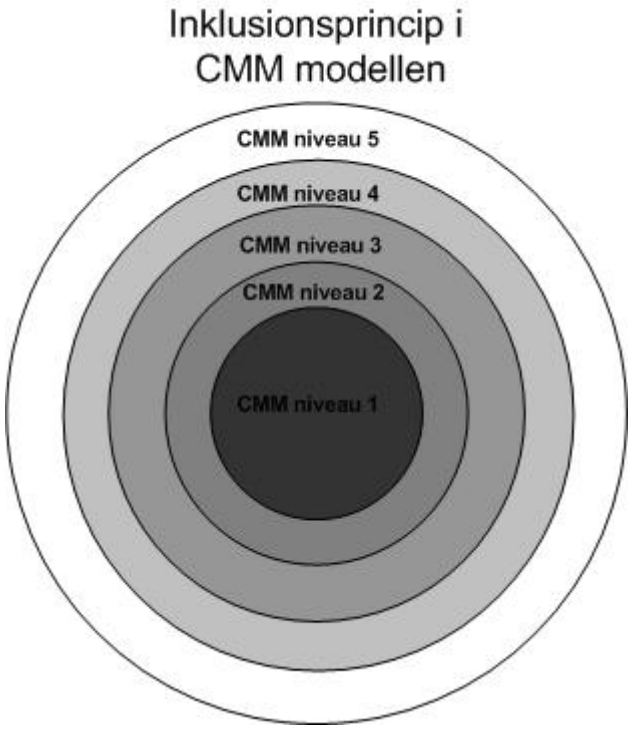


Figur 1 – CMM modellen [Jalote, s. 7]

Kravene til niveau 1 (initial) er udelukkende, at organisationen udvikler software, og måden der udvikles på har ingen indflydelse på vurderingen. Allerede på niveau 2 (repeatable) er der krav om stabile og veldefinerede processer, der håndteres ens gennem hele organisationen, og områderne omhandler primært projektstyring. Skridtet til niveau 3 (defined) er stort, da der på niveau 3 er krav om specifik og tilgængelig viden fra tidligere projekter, som anvendes til at forbedre processerne i hele organisationen, hvormed processerne institutionaliseres, og skridtet til niveau 3 indebærer da også, at organisationen lærer at lære (jf. Bateson), og denne distinktion er særdeles interessant i forhold til traditionel softwareudvikling. Eksempelvis indebærer vandfaldsmodellen (som er den mest udbredte) ikke et iboende krav om, at man lærer af tidligere erfaringer, og selv om organisationen er lille, er softwareudviklingsmetoden med andre ord begrænsningen for, hvilket CMM niveau en softwareudviklingsorganisation kan befinde sig på. Skridtet til niveau 3 indebærer derfor en metodisk æn-

dring, som indeholder redskaber, som kommunikativt kan distribuere viden og erfaringer fra tidligere projekter, og organisationen skal derfor indstille sig på at dele viden, som igen medfører en nedgraduering af eksisterende organisatoriske styringsmekanismer.

På niveau 4 (Managed) er der et iboende krav om kvantitative vurderingsredskaber, som muliggør en kvantitativ vurdering af processernes resultater, som kan spille sammen med en række udbredte kvantitative ledelsesmæssige redskaber som eksempelvis *Balanced Score Card* (BSC), men niveauet indeholder også kvantitative redskaber til produktforbedring. Så snart disse redskaber eksisterer på et organisatorisk niveau, er det muligt at kontrollere og forbedre processerne kvantitativt, og disse forbedringer kan ligeledes vurderes kvantitativt. På niveau 5 (optimizing) forbedres disse processer og redskaber kontinuerligt på et organisatorisk niveau, og der eksisterer på niveau 5 mekanismer, som kvantitativt evaluerer effektiviteten af de procesforbedrende redskaber. [Jalote, s. 8-10] Indeværende speciale vil som tidlige nævnt analysere alle niveauerne i CMM modellen i forhold til XP således, at der skabes et solidt grundlag for vurderingen af XP som organisatorisk redskab, som er intensionen men dette speciale. Det direkte inklusionsprincip illustreres i figur 2.



Figur 2 – CMM modellens inklusionsprincip

Hvert niveau i CMM modellen indeholder en række *key process areas* (KPA), som er vedlagt i bilag 2. Hvert enkelt KPA er vurderet ud fra faser, som igennem en årrække er observeret i organisationer i takt med at organisationerne forbedrer sig og videreudvikles. Hvert niveau repræsenterer en væsentlig forbedring fra det foregående niveau, og niveauerne indeholder guidelinjer til procesforbedring, når organisationens niveau er blevet fastslået. Som vurderingsredskaber anvendes både spørgeskemaer, softwaredokumentation og interviews i forbindelse med CMM modellen, og fokus er næsten udelukkende rettet mod ledende personer og er dermed i den initierende fase underlagt den traditionelle softwareudviklingstraditions grundtanke; hierarkisk

opdeling af funktionssystemer, hvilket problematiseres senere. Denne praksis ændres radikalt, hvis der er tegn på, at organisationen befinder sig på et højere niveau end CMM niveau 2, da dette skridt som nævnt tidligere medfører et skift fra styring til ledelse, hvormed fokus kan rettes mod medarbejderne og i særdeleshed mod de kommunikative aspekter af organisationen. Dette fokus ændres dog igen på niveau 4 og 5, hvor styringen igennem de kvantitative vurderingsredskaber tager over, og ledelsen igen er styrende. Dette problematiseres yderligere i diskussionen s. 85.

Iagttages organisationens ledelse i en refleksion over CMM modellen, er der en klar sammenhæng mellem organisationens evne til omstilling og ledelsens evne til at effektivisere denne omstilling. Skal organisationen forandres for at tilpasse sig markedskrav (som CMM modellen lægger op til), er det ikke en medarbejders opgave at føre denne forandring ud i livet, men en ledelsesmæssig opgave; en beslutning [March, 1995, s.33]. Organisationer består jo ifølge systemteorien ikke af mennesker og objekter men udelukkende af kommunikation. Dette begrundes overordnet med, at objekter kan udskiftes efter behov, mennesker kan erstattes med alderen osv. Der må være noget mere grundlæggende, den mindste fællesnævner om man vil; og dette er ifølge Luhmann, Qvortrup og Thyssen kommunikation. Beslutningerne er med andre ord den omstændighed, som bestemmer, om organisationens handling skal gøres til præmis for egen handling, som Luhmann så poetisk udtrykker det, og denne handling lægges så ud til de forskellige funktionssystemer - i dette tilfælde ledelsen. Eksempelvis kan en ledelse på organisationens vegne beslutte, at der skal indføres overordnede kvantitative redskaber for at vurdere softwarekvaliteten i forhold til intensionen, og hvis medarbejderne, som skal anvende disse redskaber er enige heri, gør medarbejderne organisationens handling til præmis for egen handling. Modsat kan medarbejderne også aktivt vælge ikke at gøre organisationens handling til præmis for egen handling, hvormed organisationen enten må revidere beslutningen eller udskifte de medarbejdere, som ikke efterlever organisationens beslutning. Kort sagt er beslutninger i al sin enkelhed den kommunikation, der konstituerer organisationen, hvilket fører direkte over i ansvarshaverne for beslutningerne; ledelsen.

Ledelse

En hurtig specifik definition er, at ledelse ikke er en person, men beslutninger taget fra et bestemt perspektiv (funktionstilforordning). I systemteoretiske begreber er der fire fundamentale forhold, som grundlæggende karakteriserer en organisation:

- **Medlemskab og grænse** - en organisation beslutter sin grænse ved at afgøre, hvem beslutninger gælder, og hvad medlemskab vil sige.
- **Programmer** - organisationer har altid definerede mål, som sætter grænser for kommunikation og leder den i retning af, hvad der skal besluttes omkring.
- **Differentiering** - for at realisere deres programmer har organisationen personer i bestemte stillinger, som udgør den sociale beslutningspræmis.
- **Beslutninger** - en organisation fungerer og opretholder sin funktion ved at foretage, formidle og implementere beslutninger.

Den sidstnævnte - beslutninger - knytter de tre andre forhold og sig selv til sig, idet spørgsmål om medlemskab, programmer, steder, ansatte og beslutninger alle fremkommer som besluttede forhold - beslutninger

viser tilbage til tidligere beslutninger og giver anledning til fremtidige beslutninger. Beslutningsprogrammer og mål aftegner på denne måde begrænsninger for kommunikation. Et beslutningsprogram er ikke at sammenligne med f.eks. en regel, hvor der kun er en mulighed, men som et råderum indenfor hvilket der er store variationsmuligheder. Hermed fremstår koblingen mellem kommunikation, beslutninger og organisationen, idet beslutninger som en del af kommunikation konstituerer organisationen.

I systemteoriens organisationsforståelse har ledelsesfunktionen en central plads. Ledelsen kan betegnes som adressat for kommunikation til organisationen, både fra omverdenen, som omgiver organisationer, og som organisationen skal forholde sig til, og fra interne organisatoriske interessenter. Ledelse kan derfor defineres som repræsentationen af organisationens helhed i organisationen. [Åkerstrøm, 2001, s. 23]

> *Ledelse skal foretage den paradoksale operation at være en del, som indfører systemets helhed i hele systemet og altså repræsenterer systemet i systemet. Den skal træffe beslutninger, som forpligter hele systemet. De træffes i organisationen og drejer sig om organisationen.* [Thyssen; 1997, s. 78]

En kendt kompleksitetsreduktionsformel i moderne organisationer er delegering af ledelse, ansvar eller beslutningskompetence på en række områder af den organisatoriske kommunikation (temaer for kommunikation). Ledelse er på den måde noget, som foregår simultant og overalt i organisationen. Ledelse foregår derfor ikke kun i den øverste del af en organisation, men er noget, som involverer mange personer (ikke kun dem som benævnes ledere) og foregår mange steder, og som noget, der via beslutninger er relateret til den overordnede ledelses centrale funktion som beslutningstageren par excellence i organisationen - den øverste ledelse har det overordnede ansvar for organisationens funktionsudfoldelse. (Hvorfor ethvert organisatorisk forhold kan henføres til den overordnede ledelse og/eller dens beslutninger, også på trods af delegering af beslutningskompetencen). Men i det daglige vil langt størstedelen af organisationens ledelse dvs. beslutninger blive taget af dem, som er tilforordnet beslutningskompetencen i relation til bestemte organisatoriske temaer.

Ledelsesfunktionen er, ligesom alt andet i en organisation, under indflydelse af de beslutningsprogrammer og mål, organisationen har besluttet at arbejde efter og stræbe imod. En ledelse kan derfor ikke beslutte hvad som helst, men må agere ledelse indenfor de rammer, som beslutningsprogrammerne og målene giver. Ledelsens funktion er at lede organisationen til realisering af sine mål, og dette gøres vha. de beslutningsprogrammer, organisationen anvender som en form for egenkonstituering, hvilket nødvendiggør et over blik over selve beslutningsteorien, som skal danne grundlag for den teoretiske analyse af XP og CMM modellen.

Opsummering

Den grundlæggende antagelse i den anvendte organisationsteori baseres på antagelsen om, at organisationer grundlæggende består af kommunikation og ikke af personer, og at kommunikation konstituerer organisationen gennem beslutninger. Konkret er disse strukturer svært iagttagelige med almindelige kommunikative redskaber, hvorfor CMM modellen anvendes for at konkretisere disse konstituerende beslutningselementer yderligere. CMM modellen tager udgangspunkt i en hierarkisk opdeling af softwareorganisationen, hvorfor selve ledelsesperspektivet gøres interessant på et analytisk niveau, da ledere i både teoretisk og praktisk for-

stand repræsenterer organisationen og dermed kulturen i organisationen. Derfor er det vigtigt for specialet i næste kapitel at belyse beslutninger (af ledelsen) og den måde, hvorpå beslutninger repræsenterer viden.

Beslutninger

Out of intense complexities intense simplicities emerge
- Winston Churchill

Anvendelsen af beslutninger som organisationens konstituerende element indebærer udover specificeringen af beslutninger også en teori om videnrepræsentation og en teori om præferencer i og med, at specialet intentionelt er operationelt anlagt. Ofte er ledelsesmæssige beslutninger baseret på uvished og delvist uklare mål, men for at beslutninger kan anvendes som organisatorisk grundelement, skal den tilgængelige viden repræsenteres formålstjenligt, hvilket leder til et blik på videnrepræsentation som medie for beslutninger. Derfor vil dette kapitel primært omhandle disse to områder, hvor vigtigheden af beslutninger i konteksten søges eksplicitet i et operationelt perspektiv til den senere analyse. Hertil er den primære anvendelse James G. March, som en mere nutidig pendant til Herbert Simons beslutningsteori, som March tager afsæt i. Luhmanns beslutningsteori er fravalgt, ikke fordi den ikke kan udtale sig om området, men fordi den er svært operationaliserbar i modsætning til March. Beslutninger ses i dette speciale som en del af den organisatoriske kommunikation, som er handlingsrammen for beslutninger.

Hvad er beslutninger?

For at forstå, hvad en beslutning er, er det vigtigt at belyse den kontekst, i hvilken beslutningen indgår. I dette speciale er beslutninger i softwareorganisationen underlagt konsekvensstyring i interaktionen, hvilket vil sige, at beslutninger har konsekvenser for de personer, der indgår i det sociale funktionssystem, eksempelvis for medarbejdere, ledere, interessevirksomhed og slutbrugere. Konsekvensstyring er i denne sammenhæng ønsket om fremtidige konsekvenser. March betragter beslutninger deskriptivt som en bevidst, konsekvensstyret handling, der er baseret på fire præskriptive elementer:

> ***Et kendskab til alternativer.*** *Beslutningstagere har en række handlingsalternativer at vælge imellem.*
> ***Et kendskab til konsekvenser.*** *Beslutningstagere kender konsekvenserne af handlingsalternativerne, i det mindste i form af en sandsynlighedsfordeling.*
> ***Et konsistent sæt af præferencer.*** *Beslutningstagere har konsistente værdier, hvorpå konsekvenserne af handlingsalternativerne kan sammenlignes i form af deres subjektive værdi.*
> ***En beslutningsregel****. Beslutningstagere har regler, som sætter dem i stand til at vælge et enkelt handlingsalternativ på basis af dets konsekvenser for præferencerne.*
>
> [March, 1994, s. 2-3 & March, 1995, s. 52]

Denne model for beslutninger har ifølge March sin stærke side i sin fleksibilitet, da det er muligt at tilpasse hvert enkelt element, hvis modellen ikke ser ud til at passe, således at eksempelvis beslutningsregler ændres, sådan at der opstår en ny fortolkning, der kan besvare de centrale teser i beslutningsprocessen. I tilfælde af at præstationerne efterkommer det ønskede, stilles der ofte ikke større spørgsmål om mulige alternativer [March, 1995, s. 53]. I tilfælde af at præstationer ikke efterkommer forventningerne søges nye muligheder, og disse

hidtil uudnyttede ressourcer benævner March som *slack*. Slack er en benævnelse for de reserver, der ikke er nødvendige at mobilisere i den nuværende situation, men som kan mobiliseres, hvis situationen kræver det [March, 1995, s. 54].

> *(…) forståelsen af handlinger i situationer, der er præget af ufuldstændig information, måske afhænger mere af forestillinger om opmærksomhedsfokusering, end af forestillinger om beslutningstagen.*
>
> [March, 1995, s. 162]

March finder det særdeles interessant, at beslutningstagere ofte ser bort fra muligheder, der er meget usandsynlige, uanset hvilke konsekvenser det måtte have. Hermed sker der en kompleksitetsreduktion, hvor beslutningsgrundlaget mindskes, og udfaldsrummet er således reduceret, og beslutningstagere anvender dette udfaldsrum som grundlag for en given beslutning. [March, 1995, s.149]. Selve verificeringen af beslutningen bunder i selve grundlaget for beslutningen, hvilket inddrages senere i analysen, og derfor vil næste afsnit redegøre for dette speciales definition af en rationel beslutning, hvilket gerne skulle underbygge den kommende analyse.

Den rationelle beslutning

Dybest set handler rationalitet om, at befinder et individ sig i en situation, hvor han eller hun kan vælge mere eller mindre af noget ønskværdigt, vil vedkommende altid vælge mere frem for mindre. Disse valg er dog sjældent så simple som ovenstående eksempel, og enhver handling har konsekvenser enten i form af gevinster eller omkostninger. Hvis omkostningerne eller konsekvenserne er kendte, træffer individet det valg, som han eller hun foretrækker. Når omkostninger eller konsekvenser ikke er kendte, vil individet deducere og kalkulere den sandsynlige nytteværdi af handlingsalternativerne. Derfor handler rationalitet ikke om midler eller mål, men om den måde hvorpå individet når sit mål. Rationalitet er således ikke per se noget om individets konkrete præferencer og ønsker, men derimod et udtryk for en konsistent relation mellem præferencer, information og handling.

Selve ordet rationalitet har ifølge March flere betydninger, som afhængig af konteksten spænder fra kynisk og materialistisk over intelligent og succesfuld til fornuftigt og normalt. Den "rene" teori om rationelle valg bygger på forestillingen om, at alle præferencer, der ligger til grundlag for valget, er kortlagt, og alle mulige alternativer til valget er udforsket. Denne teori udmærker sig ved at være særdeles let at kvantificere, hvilket henvender sig til ingeniørdisciplinerne, men er sværere at validere, når det drejer sig om organisationer eller personer [March, 1994, s. 4].

Men der er faktisk flere tilstande af rationalitetstyper, som igennem analysen vil nuancere de valg af rationalitet, som de specifikke KPA i CMM modellens enkelte niveauer immanent vælger. Gennem det immanente valg af rationalitet er ideen at sammenligne valget af den iboende rationalitet i CMM modellen med den iboende rationalitet i XP, hvorfor Noorderhavens nedenstående opdeling af rationalitetstyper bliver interessant.

- **Substantiv rationalitet**. Det alternativ som er objektivt bedst, og udgør beslutningstagerens præference for at valg. Antagelsen om, at ingen mangel på information eller logiske fejl gør sig gældende.

- **Instrumentel rationalitet.** Beslutningstageren vælger de rette meninger ud fra sin overbevisning. Logiske fejlslutninger udelukkes, men overbevisningen modsvarer ikke nødvendigvis den objektive virkelighed.
- **Kognitiv rationalitet.** Alle tilgængelige informationer inddrages i beslutningssystemet og afspejles i beslutningstagerens overbevisning. En variant af den instrumentelle rationalitet.
- **Proceduremæssig rationalitet.** En fornuftig beslutningsproces forudsætter de tilgængelige informationer og den nødvendige refleksive behandling i præference til en konkret kontekst.

[Baseret på Noorderhaven, 1995, s. 45-49]

Den substantive rationalitet bygger på en positivistisk antagelse om, at alt kan beskrives fuldstændigt. Den instrumentelle rationalitet antager, at styring er det primære fokus, og retter sig dermed primært mod de beslutningstagere, der kan handle ud fra en subjektiv overbevisning, hvilket måske ikke altid stemmer overens med den objektive virkelighed. Den kognitive rationalitet bygger på antagelsen om, at der på et eller andet tidspunkt skal træffes en beslutning, og man derfor må vurdere og konkludere ud fra de informationer, der nu engang er tilgængelige på det pågældende tidspunkt; en rationalitet der inddrager tidshorisonten, som er en vigtig faktor i al softwareudvikling. Den proceduremæssige rationalitet bygger på refleksivitet og kontekstafhængighed; en rationalitet der vurderer informationer i konteksten og således sætter beslutninger i direkte forbindelse med den specifikke software.

Begrænset rationalitet

Teorien om *limited rationality* stammer fra Herbert Simon, som i øvrigt fik nobelprisen for netop anvendelsen af dette begreb. Begrænset rationalitet er særdeles vigtigt at nævne, da det i den teoretiske analyse vil fremgå, at rationalitet ikke bør tages for givet, når man taler personer og handlinger. I teorien er ovenstående distinktion mellem forskellige former for rationalitet særdeles hensigtsmæssig i dette speciale, da det hermed er muligt med relativt enkle midler i analysen at udrede de forskellige former for rationalitet ud fra grundlæggende antagelser i de anvendte teorier. Imidlertid er antagelsen om, at individet handler rationelt ikke i overensstemmelse med virkeligheden. At individets rangordning af alternativer sker med udgangspunkt i den forventede nytte modificerer ikke rationalitetsantagelsen. Individet handler stadig rationelt i den forstand, at han eller hun vælger det alternativ, der giver mest mening og dermed mest forventet nytteværdi. Imidlertid er der en del diskussion om den usikkerhed, der er tilknyttet dette valg, da ikke alle handlingsalternativer er blotlagt (og sandsynligvis ikke kan blotlægges). Diskussionen er både af teoretisk og praktisk karakter, da den drejer sig om, hvorvidt individet handler fuldt rationelt eller begrænset rationelt.

Nu handler dette speciale ikke alene om individets rationelle eller begrænsede rationelle handlinger, men distinktionen er vigtig, idet den grundlæggende enighed blandt forskellige teoretikere synes at være, at individet skal betragtes som begrænset rationelt. Det vil sige, at individet, i modsætning til den klassiske opfattelse af individet som værende fuldt rationelt, i dag er underlagt kognitive begrænsninger såvel teoretisk som praktisk. Det betyder, at individet ikke søger at undersøge et komplet genstandsfelt for at træffe en beslutning men stopper, når en tilfredsstillende og relevant løsning er fundet.

Uanset valget (bevidst eller ubevist) af rationalitet skal beslutningen repræsenteres. Beslutninger kan anskues som et sammensurium af den akkumulerede viden, der er tilgængelig på det givne tidspunkt, hvorfor specialet vil dvæle ved beslutninger som repræsentation af viden til brug i den videre analyse.

Videnrepræsentation

I dette afsnit vil videnrepræsentation blive diskuteret på et begrebsteoretisk niveau, såvel som beskrevet ud fra et pragmatisk standpunkt. Der tages udgangspunkt i Davis, Schrobe og Szolovits, eftersom deres gennembearbejdede teoretiske refleksion over videnrepræsentation udmønter sig i direkte praktiske anvendelsesmuligheder, som de selv ekspliciterer [Davis, Schrobe og Szolovits, 1993]. Tanken er hermed at koble softwareorganisation og beslutninger i et operationelt perspektiv, for på den måde at skabe grundlag for vurderingen af XP som beslutningsunderstøttende værktøj i den operationelle analytiske del af specialet.

Teoretisk videnrepræsentation

Davis, Schrobe og Szolovits går til det helt basale spørgsmål om, hvad videnrepræsentation er, og besvarer det ud fra den opfattelse, at det

> *"[...] best can be understood in terms of five important and distictly different roles that a representation plays [...]".*
>
> [Davis, Schrobe og Szolovits, 1993, s.1]

Grundlæggende er videnrepræsentation et surrogat. Desuden er det en ontologisk tilknytning, en fragmentarisk teori om intelligente udledninger, et medie for effektiv beregning og et medie for menneskelig udsagn, hvilket peger hen imod præmisserne for en given beslutning. Videnrepræsentation er et surrogat for et fænomen i virkeligheden, og fænomen skal her forstås som fysisk eksisterende objekter såvel som handlinger i verden. Dette begrundes med, at

> *"[...] reasoning is a process that goes on internally, while most things [...] exist only externally."*
>
> [Davis, Schrobe og Szolovits, 1993, s.3]

Videnrepræsentationen er altså det, som gør det tænkeligt at erstatte et fænomen (der er eksternt, i relation til en given entitet af bevidsthed) med tanken om et fænomen (der er internt, i samme relation). Helt simplificeret kan det udtrykkes, at verden uden for bevidstheden kan vise sig inden for bevidstheden gennem repræsentation. Deraf følger to centrale kvalitative størrelser for repræsentationen: dens identitet og dens nøjagtighed. Identiteten er, hvad den er et surrogat for, det vil sige, hvad den refererer til i virkeligheden; forbindelsen til et fænomen i virkeligheden er repræsentationens semantik. Nøjagtigheden omhandler, hvor tæt repræsentationen er på det, den faktisk refererer til, det den er et surrogat for, det vil sige, hvilke egenskaber af det virkelige fænomen repræsentationen ekspliciterer, og hvilke egenskaber den udelader. [Davis, Schrobe og Szolovits, 1993, s.4]

Davis, Schrobe og Szolovits påpeger, at den eneste fuldstændigt nøjagtige repræsentation af et fænomen er fænomenet selv, og deraf følger, at enhver anden repræsentation er en tilnærmelse, og det er derfor, viden-repræsentation også er en ontologisk tilknytning. Når al repræsentation er

> *"[...] imperfect approximations to reality, each approximation attending to some things and ignoring others […] we are in the very same act unavoidably making a set of decisions about how and what to see in the world".*
>
> [Davis, Schrobe og Szolovits, 1993, s.5]

Ontologi er i denne sammenhæng at forstå som en model på verden, hvad der eksisterer i verden. Igen kan en simplificering gavne forståelsen; Davis, Schrobe og Szolovits pointe kan sættes på spidsen således: hvis det ikke kan repræsenteres, så er det ikke. Dette er i særdeleshed en essentiel pointe, når konteksten som denne er softwareudvikling. Kan krav ikke repræsenteres, så kan de ikke udvikles.

Dette bør dog, indskærper Davis, Schrobe og Szolovits, ikke fornægtes men omfavnes som en styrke, fordi videnrepræsentation derved per definition også er kompleksitetsreduktion, der tilsteder os at indsnævre opmærksomheden på de elementer af verden, som måtte findes relevante:

> *"[...] they allow us to cope with what would otherwise be untenable complexity and detail."*
>
> [Davis, Schrobe og Szolovits, 1993, s.5]

Videnrepræsentation er også en fragmentarisk teori om intelligente udledninger, idet valget eller udviklingen af en sådan ofte er

> *"[...] motivated by some insight indicating how people reason intelligently, or by some belief about what it means to reason intelligently at all."*
>
> [Davis, Schrobe og Szolovits, 1993, s.7]

Der er to indlysende grunde til, at videnrepræsentation kan betegnes som en fragmentarisk teori om intelligente udledninger: 1. fordi repræsentationen oftest kun indbefatter dele af den indsigt i, hvordan intelligente udledninger foretages, som dannede grundlag for valget af den. 2. fordi den indsigt (antageligvis) i sig selv ikke er en komplet indsigt i dette område. Kort sagt vil repræsentation kun indbefatte det, som iagttageren iagttager, da man ikke kan se, hvad man ikke kan se, og dermed ikke indeholder alle mulige dele af det iagttagede område - deraf det fragmentariske. Videnrepræsentation er en teori om intelligente udledninger, fordi den afgrænser, hvad der rent faktisk kan udledes, såvel som den fordrer, hvad der bør udledes. [Davis, Schrobe og Szolovits, 1993, s.7] Dette kan derfor anskues sådan, at en given videnrepræsentation indtager en position i forhold til spørgsmålene:

> *"What does it mean to reason intelligently? [...] What can we infer from what we know? [..] What ought we to infer from what we know?".*
>
> [Davis, Schrobe og Szolovits, 1993, s.7]

En videnrepræsentations ontologiske tilknytning er afgørende for, hvad der faktisk kan udledes ud fra den. Dens teori om intelligent udledning er imidlertid afgørende for, hvad og hvordan der kan udledes deraf, såvel som hvad der bør udledes deraf. Videnrepræsentationens, ofte implicitte, indlejrede teori om intelligent udledning er derfor af afgørende betydning. Hvis den for eksempel definerer præcist, hvad der er tilladte (dvs. intelligente) udledninger af et givent grundlag men i ringe grad, hvad der er rimelige udledninger, vil valgmulighederne af udledninger hurtigt blive uoverskuelige.[3] Dette medfører, at den universelle og generelt anvendelige videnrepræsentation er en meget problematisk størrelsesorden, fordi

> *"[...] a representation with these goals cannot single out any particular set of inferences to recommend [...]".*
>
> [Davis, Schrobe og Szolovits, 1993, s.12]

Dels fordi de fleste strategier for udledninger vil være hensigtsmæssige i nogle sammenhænge men fatale i andre, dels fordi repræsentationens udtryk tildeles en bestemt eller anden betydning, hvis de knyttes til bestemte udledninger [det repræsentationelle udtryk ville få en anden semantisk værdi, eftersom det ikke længere refererer til fænomenet men til en bestemt konception af eller holdning til fænomenet], hvorved de ville miste deres deklarative egenskab og dermed deres universalitet.

Videnrepræsentation er også det medie, igennem hvilket vi udtrykker os om verden og kommunikerer derom med hinanden, hvilket er særdeles interessant i forhold til kravspecificering i en softwareudviklingsproces. Dette rejser i første omgang spørgsmålene om, hvor generel (anvendelig) repræsentationen er, hvor præcis den er, og om den muliggør tilstrækkelig høj grad af udtryksfuldhed. Men det rejser også spørgsmålet om, hvor funktionel repræsentationen er som kommunikationsmedie:

> *"[...] how easy is it for us to "talk" or think in that language?"*
>
> [Davis, Schrobe og Szolovits, 1993, s.15]

Spørgsmålet er møntet på pragmatisk anvendelighed frem for teknisk mulighed; det interessante er, mener Davis, Schrobe og Szolovits, ikke om noget kan udtrykkes eller kommunikeres med repræsentationen, men hvor overkommeligt det er at gøre det, eller som de selv udtrykker det:

> *"A representation is a language in which we communicate; hence we must be able to speak it without heroic effort."*
>
> [Davis, Schrobe og Szolovits, 1993, s.15]

[3] hvilket i sig selv kunne være en interessant indgangsvinkel til modallogikken, der jo blandt andet beskæftiger sig med at sætninger (og dermed videnrepræsentation) kan være sande på mange forskellige måder (en sandhedsfunktionel fortolkning), alt afhængigt af om sætninger fortolkes ekstentionelt (aktuel verden) eller intentionelt (andre verdener). Imidlertid er dette ikke omdrejningspunktet i specialet men et interessant perspektiv, som kunne forfølges såfremt den operationelle anvendelse tillader det.

Teoriens praktiske anvendelighed

Davis, Schrobe og Szolovits ekspliciterer, hvordan en sådan definition af og refleksion over videnrepræsentation kan anvendes i praksis. Netop det, at de tager udgangspunkt i den operationelle anvendelse og funktion af videnrepræsentation for at definere det, skal gøre teorien umiddelbart anvendelig:

> *"What is a knowledge representation? We argue that the notion can best be understood in terms of five distinct roles it plays, each crucial to the task at hand".*
>
> [Davis, Schrobe og Szolovits, 1993, s.2, understregning tilføjet]

Dermed fremhæver de ydermere, at den foreliggende opgave er særdeles afgørende for, hvilken repræsentation der er hensigtsmæssig. I relation til den operationelle anvendelighed er der imidlertid en konsekvens af deres teori, som er implicit i Davis, Schrobe og Szolovits redegørelse: det følger, at videnrepræsentation indebærer beslutningstagen, når de blandt andet skriver, at

> *"[...] in selecting any representation we are in the very same act unavoidably making a set of decisions about how and what to see in the world [...] decisions have to be made with all the representation technologies, because [...] they offer a way of seeing but don't indicate how to instantiate that view."*
>
> [Davis, Schrobe og Szolovits, 1993, s.6 understregning tilføjet]

Opsummering

Kapitlet viser en tæt relation mellem beslutninger og videnrepræsentation og illustrerer samtidig relevansen af operationel videnrepræsentation i forhold til softwareudvikling, hvilket er direkte operationaliserbar til den teoretiske analyse. Vurderingen af det implicitte valg af rationalitet og et kendskab til både præmisser og konsekvenser af en given beslutning må unægtelig præge den måde, hvorpå viden repræsenteres gennem beslutningerne. Selvom individet handler begrænset rationelt, er der kun lidt mening i at indeholde denne begrænsede rationalitet i den videre analyse, hvor CMM modellens KPA analyseres iht. XP og CMM modellen, da det er en usikkerhed, som man kun kan håndtere, men aldrig undgå. Det bliver derfor en uomtvistelig præmis der vil være til stede i analysen, men som operationelt ikke vil inddrages. Derudover ligger der i videnrepræsentationen også en forudindtaget præskriptiv holdning overfor mulige udfaldsrum, hvilket senere anvendes til at iagttage, om CMM modellen indeholder elementer, som ikke tages for givet i XP. Med den etablerede forståelsesramme for softwareorganisationen, CMM modellen, beslutninger og videnrepræsentation er det nu hensigtsmæssigt at beskrive XP som metodologi og eksplicitere de redskaber og metoder, som kan anvendes i et analytisk perspektiv.

Extreme Programming

It is a bad plan that admits of no modification.
Publus Syrus, 1. århundrede f.kr.

Dette kapitel har til formål at beskrive softwareudviklingsmetoden Extreme Programming (XP) med en operationel optik således, at der skabes grundlag for en analyse af XP som supporterende element i de organisatoriske beslutningsstrukturer, som er medvirkende til at konstituere softwareorganisationen på et af de forskellige CMM niveauer. Desuden har kapitlet en interesse i den objektorienterede del af XP, hvilket afkræver en kort forklaring af principperne bag objektorienteret udvikling. Den operationelle optik muliggør, at kapitlet i mindre udstrækning omhandler de teoretiske elementer i XP og i højere grad fokuserer på den organisatoriske anvendelighed. Ydermere er formålet at præsentere den verdensopfattelse, som immanent er repræsenteret i XP; en verdensopfattelse som er nødvendig for overhovedet at anvende XP organisatorisk. Denne verdensopfattelse medførte grundlæggelsen af *Agile Software Development* metodologien, hvor XP tjener som udgangspunkt [Eks. Cockburn, 2002]. Det er dog imidlertid givende at begynde med selve objektorienteringen, da også XP er udspringet af udviklingen af objektorienterede udviklingsmetoder. Der tages udgangspunkt i Lars Mathiassens *Objektorienteret analyse* [Mathiassen, 1993].

Objektorienteret udvikling

Med objektorienterede programmeringssprog introduceredes de grundlæggende objektorienterede koncepter, som senere også har fundet anvendelse i dette speciale i forbindelse med systemanalyse og design i selve forundersøgelsesfasen i ***m*2i3D**. Objektorienteret programmering kan fundamentalt ses som en videreudvikling og formalisering af struktureret programmerings abstrakte datatyper, i form af objekter som selvstændige enheder, hvilke omfatter såvel datatyper som processer. Objekter giver muligheder for en mere formålstjenlig opdeling af programmer i mindre enheder og gruppering af relaterede data/datatyper og processer, hvilket igen understøtter tanken i XP [Mathiassen, 1998, s. 6-7].

Tilknytningerne mellem forskellige programdele mindskes radikalt igennem specifikation af objekternes eksterne interface og indkapsling af implementeringsdetaljer, hvilket mindsker kompleksiteten i det samlede system, som henleder opmærksomheden på XP's grundlæggende værdier, som beskrevet på s. 32. Endeligt giver objektklasser og nedarving af egenskaber fra disse, mulighed for en samlet beskrivelse af fælles generelle egenskaber ved forskellige fænomener, samt en specialisering af disses særlige egenskaber til brug i det videre udviklingsforløb [Mathiassen, 1998, s. 8]. Dette kan ligeledes medvirke til at mindske kompleksiteten i det samlede system, hvilket igen understøtter XP's grundlæggende værdisæt.

Anvendelse af objektorienterede koncepter i programmeringssprog er dog ikke nogen garanti for god håndtering af kompleksitet i systemet. Dårlig implementering af objektorienterede koncepter i de enkelte programmeringssprog og anvendelse af disse kan tværtimod medvirke til væsentlig øget kompleksitet. For eksempel i form af løst specificerede eksterne interfaces og "overriding" af nedarvede metoder, der kan resultere i uigennemskuelige bindinger mellem forskellige objektklasser. Generelt må man dog sige at objektorienterede ud-

viklingsmetoder er at foretrække frem for ikke-objektorienterede udviklingsmetoder, idet de i sig selv sandsynliggør en bedre strukturering af koden.

De samme objektorienterede koncepter anvendes ligeledes i forbindelse med analyse og design, og produkterne af disse aktiviteter er ligeledes objektmodeller med specifikation af objekter og disses egenskaber i form af "attributter" og "metoder". Der er således ingen principiel/grundlæggende forskel på objektorienteret analyse, design og programmering. Det hele handler om objekter, ganske som XP handler om kommunikation, hvilket ekspliciteres på side 32.

De forskellige objektorienterede aktiviteter fokuserer blot på forskellige aspekter ved de i princippet identiske objekter. Analyse identificerer objekter og disses egenskaber samt relationer mellem objekterne. Design strukturerer objekterne i objektklassehierarkier og specificerer objekternes eksterne grænseflader. Programmering implementerer objekterne som specifikke datatyper/-strukturer og algoritmer/funktioner/procedurer. De forskellige aktiviteter er således ideelt set blot trinvis forbedring af de samme objekter.

I praksis holder enhedsmodelleringen igennem objektorienteret analyse, design og programmering dog ikke helt, idet de enkelte aktiviteter potentielt må arbejde med separate og måske meget forskellige objektmodeller, hvilket Mathiassen illustrerer ved at inddrage *Unified Modeling Language* (UML) [Matiassen, 1998, s. 325, 377-378]. Objekterne ved analyse er abstraktioner over fænomener og begreber i "den virkelige verden" - eventuelt med uformaliserede egenskaber. Objekterne ved design er computerrelaterede konstruktioner - eventuelt uden direkte eller umiddelbar relation til fænomener/begreber i "den virkelige verden". Objekterne ved programmering er specifikke implementeringer ud fra bestemte programmeringssprogs konkrete (og måske begrænsede) implementering af de objektorienterede koncepter - eventuelt uden direkte 1-til-1 relation til de abstrakte designkonstruktioner.

Muligheden for objektorienteret systemudvikling som et gensidigt sammenspil mellem analyse, design og programmering begrænses dermed, i det omfang der arbejdes med forskellige modeller, hvis elementer ikke er direkte relaterede, da det kræver (manuel) opdatering af de forskellige modeller for at sikre konsistens imellem dem. Dette trækker således i retning af et traditionelt, ensrettet systemudviklingsforløb, men er stadig ingen selvfølge.

En vigtig kvalitet ved objektorientering er, at objekter hævdes at være "naturlige" i forhold til "den virkelige verden" for såvel brugere som systemudviklere. Dette kan dog diskuteres, og det er i hvert fald ikke altid tilfældet. For eksempel er det ikke umiddelbart "naturligt", på hvilket objekt metoder skal placeres i forbindelse med processer, der implicerer flere ligeværdige objekter uden ét specifikt objekt, der er naturligt styrende. I forbindelse med objektorienteret analyse er identificeringen af objekter heller ikke altid umiddelbart naturlig, og der er givet meget forskellige, mere konkrete forslag til identificering af de "rigtige" objekter som for eksempel fysiske artefakter i "den virkelige verden" og problemmønstre.

Men på trods af disse problemer, er objektorienteringens forenende koncepter særdeles interessante, og problemerne er ikke ældre og større, end at der stadig er håb om løsninger og forbedringer, og det står klart, at

der er kohærens imellem objektorientering og XP. Objektorienteringen tillader, at man i højere grad kan opdele programmet i små dele, hvor hver enkelt del kan forstås uden at hele programmet kan forstås, hvilket betyder, at en fejl i en programdel kan rettes uden at resten af koden inddrages. Opdelingen af programmer i små moduler er et af kerneområderne i XP, og igennem beskrivelsen af XP identificeres denne modularisering, og XP opnår kohærens med generel modularisering. Men først en indføring i ideen bag XP, primært baseret på litteratur af Kent Beck og Martin Fowler, som kan betegnes som ophavsmændene til både XP og Agile Software Development.

Ideen bag Extreme Programming

XP er fortrinsvist udviklet til softwareudviklere af softwareudviklere og er resultatet af mange mere eller mindre fejlslagne projekter, som enten er blevet for dyre, har overskredet tidsplanen, har haft for dårlig kvalitet eller hvor resultatet ikke har stemt overens med målet. Derudover har XP metodologien en række erfaringer fra gode projekter, som har efterlevet god kvalitet på den aftalte tid. Disse erfaringer er repræsenteret og forbedret i metodologien. I Danmark kender vi lignende situationer fra eksempelvis AMANDA og de elektroniske patientjournaler (EPJ), som hver især udmærker sig ved ikke at have opfyldt nogen af de kriterier, man må forvente af et udviklingsprojekt; stabil økonomi, fastlagt tidsrum, kvalitet og opfyldelse af mål. Metodologien bag XP baseres på netop disse fire principper, som i XP betragtes som delvist variable kriterier i stedet for statiske, i stærk kontrast til generelle softwareudviklingsmetoder, men meget lig modellen for beslutninger (se side 21) [Beck, 2000, s. 15-19; Beck & Fowler, 2001, s. 27-30].

I og med at kriterierne betragtes som variable, opstår der en dynamisk interaktion mellem kriterierne, hvor der i en softwareudviklingsproces indenfor XP altid skal være mindst én variabel, så man kan nå i mål i relativ sikkerhed. Ideen om mindst én variabel i de fire grundlæggende kriterier gør, at man hele tiden kan justere et af kriterierne og dermed få hele kabalen til at gå op. Et eksempel: Tid, økonomi, kvalitet og mål er fastlagte i begyndelsen af udviklingsprocessen, men i løbet af udviklingen står det klart, at økonomi, kvalitet og mål kan opnås, men ikke indenfor den angivne tidsramme. Derfor er tiden den variabel, som ledelsen kan justere for at opfylde de tre andre kriterier. Dette giver en smidigere udvikling, selvom tidsgrænsen rykkes. I eksemplet med AMANDA blev alle fire kriterier justeret flere gange, og her er der ikke tale om en fleksibel udvikling, men om udvikling med dårlig ledelse og måske falske forudsætninger, kravspecificeringer og estimater. Denne situation kan kort beskrives, for at få et billede af den virkelighed, som højst sandsynligt har været gældende i udviklingen af AMANDA:

> *Forestil dig en rund, skrånende bane med mange mål, hvor folk spiller fodbold. Mange forskellige mennesker kan gå ind i spillet (eller forlade det) på forskellige tidspunkter. Nogle kan kaste nye bolde ind i spiller eller fjerne dem. Mens de deltager i spillet, prøver de hver især at sparke til enhver bold, der kommer forbi dem, i et af de mål, de synes om, og væk fra de mål, de ønsker at undgå.*
>
> [March, 1995, s. 63]

Der er ingen tvivl om at ikke alle kriterier skal opretholdes samtidig, men at hvert kriterium interagerer med de andre. Interaktionen mellem de fire kriterier forudsætter altså, at der også er statiske kriterier (i hvert fald i en periode), så forholdet mellem de fire kriterier er en ledelsesmæssig opgave, hvilket igen fører tilbage til den

rationelle beslutning og deraf direkte til videnrepræsentation. Forholdet mellem statiske og dynamiske kriterier er en problematisk ledelsesmæssig størrelse, da eksempelvis økonomi og tid er forholdsvis statiske kriterier, hvis man skal byde i en udbudsrunde, men mere dynamiske, hvis der er tale om intern udvikling. Dette forhold afkræver konstant ledelsesmæssig stillingtagen og er forhold, som man skal håndtere varsomt men ikke kan komme uden om. Ingen kriterier bør dog ifølge XP være statiske igennem en hel udviklingsproces, hvilket strider mod almindelige softwareudviklingsmetoder. Men først et blik på de grundlæggende værdier i XP.

Værdier

Operationel håndtering af ovenstående kriterier gør det tvingende nødvendigt i XP at fokusere på det sociale aspekt i softwareudvikling. Der er med andre ord brug for retningslinjer, så det er muligt kontinuerligt at iagttage, om det enkelte softwareprojekt går i den rigtige retning. Problemet er imidlertid, at de individuelle kortfristede mål ofte kommer i konflikt med organisationens langsigtede mål. For at inkorporere organisationens langsigtede mål i de individuelle kortsigtede mål på projektniveau, arbejder XP efter fire grundlæggende værdier; kommunikation, enkelhed, feedback og mod [Beck, 2000, s. 29, Beck & Fowler, 2001, s. 27]. Disse grundlæggende værdier bliver derfor specificeret nærmere i nedenstående afsnit for senere at kunne vurdere, om disse værdier kan supportere de organisatoriske beslutninger, som er indlejret i CMM modellens niveau 3.

Kommunikation

Kort fortalt handler denne første værdi om at få folk til at kommunikere. Det essentielle heri er ikke at forhindre, at dårlig kommunikation finder sted, da dette ville være et absurd og uopnåeligt mål, da dårlig kommunikation som misinformation eller mangel på kommunikation er en organisatorisk virkelighed, man ikke kan komme uden om. Nogle gange fortæller en udvikler ikke om kritiske designændringer til projektlederen eller kollegerne, og nogle gange stiller projektlederen ikke de rigtige spørgsmål til udviklerne og får derfor et mangelfuldt eller forkert indblik i projektets progression. Dette er en uundgåelig virkelighed, som man kun kan håndtere men aldrig undgå.

XP håndterer kommunikation ved at indlejre operationelle elementer, som ikke kan fungere uden kommunikation. Det er eksempelvis pair programming, kravudredning, unit testing og estimering, som er redskaber, der ikke kan anvendes uden en vis form for kommunikation. Et springende punkt er, at der i organisationen desuden dedikeres en person til at observere, om personer kommunikerer internt i de enkelte funktionssystemer. Estimerer en person eksempelvis et område uden konsultation fra kollegerne, er det den kommunikationsdedikerede persons opgave at reintroducere den estimeringsansvarlige for de andre områdeeksperter og dermed hjælpe kommunikationen i gang igen. Alle opgaver forløber derfor med en verificerende instans, så én person ikke per se kan træffe beslutninger på et funktionssystems vegne uden en verifikation af en anden person med kendskab til området. Herved tvinges man til at kommunikere [Beck, 2000, s. 29-30]. Den dedikerede person er dog ikke en garant for en direkte verifikation af beslutningerne, da personen udelukkende søger for, at de kommunikative strukturer er til stede men ikke per se er garant for udfaldet af disse kommunikative strukturer.

Enkelhed

I alt sin enkelhed omhandler denne anden grundlæggende værdi kompleksitetsreduktion. Ikke kun omkring den udviklede software, men også omkring den kommunikative organisatoriske struktur. XP indgår meget simplificeret et væddemål: at det på sigt er bedre at indføre et enkel element i dag og betale en lidt større pris for at rette det til i morgen, end at indføre et kompliceret element i dag, som måske aldring bliver anvendt igen [Beck, 2000, s. 31]. Dette gælder både softwaresiden og organisationen bag.

Derved opstår der et vidunderligt forhold mellem enkelhed og kommunikation, da kompleksitetsreduktionen ikke alene mindsker redundante og måske uanvendelige elementer, men også skaber overblik over både software og organisation. Overblikket medfører, at kommunikationen bliver klarere og mere specifik, da der er mindre at kommunikere om, grundet enkelhed som kompleksitetsreducerende faktor.

Feedback

Feedback eksisterer i forskellige former under selve udviklingen af systemet og i de bagvedliggende organisatoriske strukturer. Omkring softwareudviklingen er den grundlæggende ide at skabe et stabilt og operationelt system så hurtigt som muligt. "Under udvikling" bliver dermed et temporært prædikat, som systemet kun eksisterer under i meget kort tid. Feedback omkring selve systemet og de grundlæggende strukturer kan dermed rimeligt hurtigt vurderes fra både projektleder, kolleger og kunder, hvormed grundlæggende designfejl kan rettes tidligt, og ny funktionalitet kan tilføjes løbende, også selvom udviklingen går som planlagt. Det giver udviklerne konkrete feedback omkring kvaliteten af deres beslutninger og udviklingsprocessen som helhed. [Beck, 2000, s. 31]

Organisatorisk anvendes feedbackmekanismen som review af eksempelvis kundeorienterede krav og projektplanlægningen, hvormed man relativt hurtigt kan opdatere planerne i forhold til de reelle forhold. Hermed er det muligt at tilpasse organisatoriske strukturer, så de relaterer til virkeligheden og ikke kun til de oprindelige estimater [Beck, 2000, s. 32-33]. Denne feedbackmekanisme falder godt i tråd med de første to værdier, da konkret feedback giver en lettere kommunikation, og hvis kommunikationen er specifik, opstår der et bedre og enklere overblik over systemet og organisationen.

Mod

Grundlæggende baseres XP på en udviklingsmetode, hvor der kontinuerligt tilføjes mere kompleksitet til det udviklede system. Det sker dog i små bidder, så hver enkelt del ikke i sig selv kan forårsage store systemnedbrud, modsat store ændringer. Problemet med denne metode er, at små bidder ikke kan ordne store problemer, hvis de skulle opstå; også selv om systemet er modulariseret. Hertil har man brug for at gennemgå og forbedre store dele af systemet, hvilket umiddelbart modsiger den tilsigtede enkelhed i systemopbygningen. Men enkelheden i systemopbygningen medfører jo netop, at udviklerne og projektlederen har mulighed for at ændre i de grundlæggende strukturer, hvor man i traditionel softwareudvikling vil være fastlåst i de oprindelige valg og dermed ikke kan overskue 200.000 linjers kode og i særdeleshed at revidere dem. At foreslå en

sådan ændring kræver mod både af organisationen og af den enkelte, for selvom strukturen er enkel, er ændringen alligevel en stor men ikke uoverkommelig opgave[4].

Set udefra er det også særdeles modigt af både den enkelte og af organisationen at arbejde med denne dag-til-dag udvikling, men har man først besluttet at indføre denne udviklingsmetode, er mod i denne sammenhæng ikke længere en del af udviklingen. Der er truffet en organisatorisk beslutning, som man kun kan følge. Men mod er også det, der driver softwareorganisationen, og mod er måske den faktor, hvormed organisationen får et forspring i forhold til konkurrenterne. Mod kan i sidste ende være den værdi, som gør organisationen overlevelsesdygtig, hvis de andre værdier ikke skulle komme til udtryk.

Ser man overordnet på de fire grundlæggende værdier i XP, supporterer kommunikationen modet, da høj risiko kan medføre højt afkast. Enkelheden supporterer modet, da det er lettere radikalt at modificere et enkelt system. Feedback supporterer både mod og kommunikation, da præcis og kontinuerlig kommunikation bevirker, at personer kan få belyst og diskuteret store designændringer og dermed løse problemer, før de opstår, fordi alle ved, hvad der diskuteres på grund af det enkle systemdesign. Operationaliseringen af disse grundlæggende værdier i XP er til en vis grad indeholdt i softwareorganisationens kultur men også i konkrete redskaber, der underbygger værdierne og omvendt, hvilket illustreres i nedenstående afsnit.

Redskaber

Som operationelt udviklingsværktøj tilføjer XP som metode en række værktøjer, som ikke alene understøtter værdierne men også skaber konsistens i både udviklingen og i den omkringliggende organisation. Af disse nævnes der i indeværende speciale kun de redskaber, som menes at have reel mulighed for at supportere de organisatoriske beslutninger. De redskaber, som i dette kapitel sorteres fra, som eksempelvis kodestandarder, er primært møntet på specifik programmering og derfor ikke interessante set i lyset af specialets organisatoriske fokusområde. Det interessante er derimod, at værdierne skaber organisatoriske aktiviteter, der i dagligdagen kan understøtte værdierne operationelt, som belyses senere i dette kapitel. [Beck, 2000, s. 49].

Selvfølgelig indeholder XP redskaber, som eksisterer i stort set alle andre softwareudviklingsmetoder såsom management planlægning, strategi vedrørende udvikling, test osv. Men selv om redskaberne synes at gå igen i forskellige metoder, ligger der i redskabernes anvendelse en fundamental kulturændring, som adskiller XP fra andre udviklingsmetoder; menneskesynet er radikalt anderledes, hvilket illustreres gennem udviklingsstrategien i næste afsnit. Planlægning er af strategiske hensyn lige så essentiel i XP som i andre softwareudviklingsmetoder, selvom tilgangen er en hel del anderledes både operationelt og mht. modificerbarhed, hvilket også illustreres i nedenstående afsnit. Der er flere grunde til at selve planlægningsstrategien vægtes højt i dette speciale. Dels at planlægningsstrategien giver et unikt indblik i den videnudveksling, som kontinuerligt forbedrer de enkelte krav og det samlede system, dels at planlægningsstrategien kan anvendes af alle organisationer på alle niveauer til stort set hvad som helst. Eksempelvis har dette speciale fulgt denne planlægningsstrategi under

[4] Her er der en interessant kobling til Piagets iagttagelser af menneskelig erkendelse og de kognitive strukturer, som kontinuerligt assimilerer og udbygger de kognitive systemer, indtil de bliver for komplekse (eller ikke passer sammen mere), hvorfor man må kompleksitetsreducere.

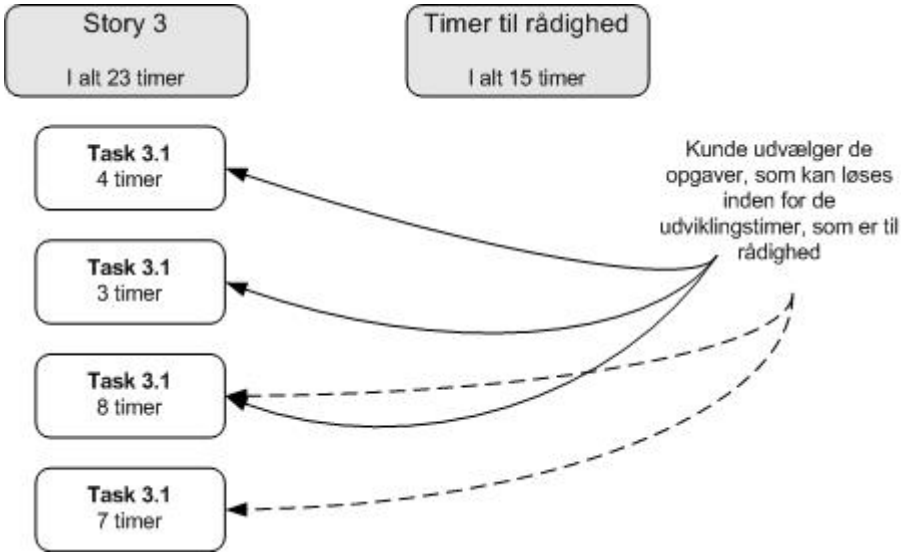
hele specialeforløbet og udnyttet både redskaber og strukturer til at modificere og forbedre de dele, som enten syntes mangelfulde eller problematiske. På den baggrund kan man uden at være selvforherligende konkludere, at udviklingsstrategien er både unik og universel.

Udviklingsstrategien

Som i alle andre softwareudviklingsmetoder er grundstenen i XP planlægningsfasen. Med en tilpasningsdygtig udviklingsstruktur skulle man umiddelbart antage, at XP er en "ad hoc" udviklingsmetode, som tager problemerne og udfordringerne som de manifesteres, men dette er ikke tilfældet. XP's betegnelse for denne planlægningsfase er "The Planning Game". Her anvendes storycards og taskcards som er indekskort, hvor en story er en beskrivelse af en begivenhed og task er de opgaver, som får begivenheden til at ske. Et storycard er altså en semantisk beskrivelse, som både udviklere og kunde er enige om, og taskcard er de opgaver som skal sikre, at denne story kan finde sted. De enkelte tasks estimeres med en faktor, eksempelvis mandetimer (hvor mange reelle udviklingstimer der kræves), hvorefter udviklingen kan påbegyndes. I de tilfælde, hvor der er problemer kan nedenstående figur illustrere den videre arbejdsgang.

Figur 3. Undersøgelsesfasen

Som det første opdeles de enkelte krav i opgaver, hvor udviklerne herefter estimerer opgaverne enkeltvis. Som planlægningsmæssig grundregel konstrueres der ikke tasks, som tager længere tid end en dags arbejde (som oftest 5-6 mandetimer). Herefter estimeres det hvor mange mandetimer, der er til rådighed i den enkelte iteration, hvorefter kunden vurderer, hvilke krav der skal indeholdes i iterationen i forhold til de timer, som er til rådighed, som figur 4 illustrerer. Nedenstående eksempel illustrerer estimeringen af den enkelte tasks, hvor kunden vælger de opgaver, som kan løses indenfor de timer, der er til rådighed.

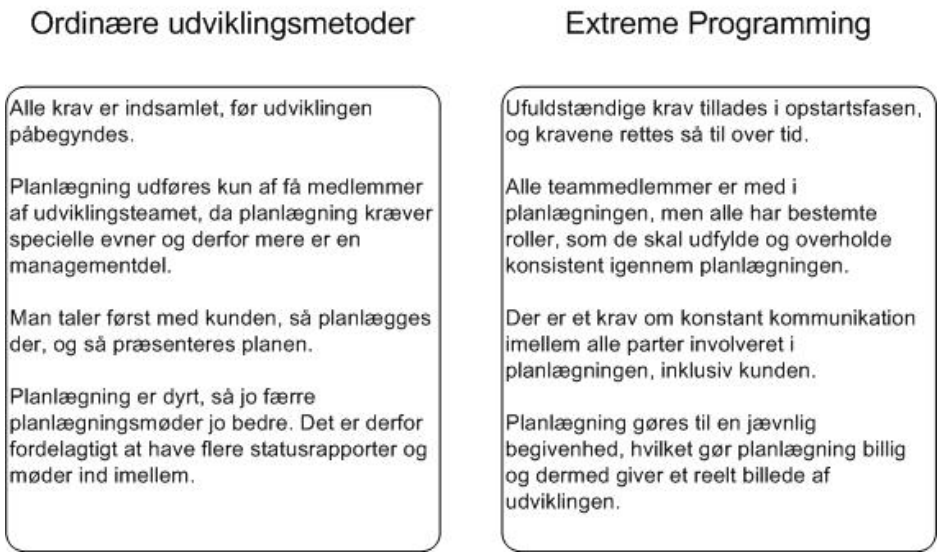


Figur 4 – planlægning af iteration

Hermed får både kunden, lederne, udviklerne og organisationen bag et meget realistisk billede af, hvad der udvikles hvornår. Estimering og planlægning er derfor to uadskillelige elementer i XP, og eftersom test og migrering er indeholdt i selve udviklingsstrategien for hver enkelt task, er der også indeholdt dokumentation i XP. Et andet skelsættende element i XP er udviklernes egen vurdering af, hvem der laver hvad. Man vælger simpelthen tasks på daglig basis, og har derfor mulighed for selv at tilrettelægge den enkelte dag ud fra personlige præferencer og begrænsninger. Hermed er man som leder sikker på at de personer, som vælger den specifikke task, forpligter sig på at fuldende opgaven i forhold til projektet. Denne opdeling minder i den grad om principperne ved modularisering, hvor systemet opdeles i små løst koblede dele, hvor hver enkelt del kan forstås uden nødvendigvis at forstå hele systemet. Modulariseringen opnås ved abstraktion (at se bort fra noget), hvilket er medvirkende til at strukturere koden (i XP igennem Stories) i en hensigtsmæssig samling af komponenter.

The Planning Game

Som i enhver anden softwareudviklingsmetode indeholder XP planlægningsværktøjer, som understøtter både udviklingsstrategien og de grundlæggende værdier i selve metoden. Planlægningsfasen er dog i XP mere detaljeret og modificerbar i et operationelt øjemed, hvormed den adskiller sig væsentligt fra eksempelvis vandfaldsmodellen. De grundlæggende forskelle illustreres i figur 5:



Figur 5 [Baseret på Auer & Miller, 2002, s. 97-98]

Det springende punkt i differencen mellem ordinære softwareudviklingsmetoder og XP er de ufuldstændige krav i opstartsfasen. Har man imidlertid gennemløbet en udviklingsproces, vil man let kunne genkende problematikken omkring kravene[5]. Der kommer som oftest nye krav til, og eksisterende krav modificeres ofte i processen. Dette kan selvfølgelig løses ved at fryse kravene tidligt i processen, og så senere lave et tilpasningsprojekt til de nye eller modificerede krav. Denne løsning er imidlertid ikke i kundens interesse, da man som kunde først meget sent i udviklingsprocessen har overblik over muligheder og løsninger på et problem. Derfor er det tvingende nødvendigt i XP at have en kontinuerlig kontakt med kunden på en daglig basis, så kravene kan specificeres løbende i processen[6]. Det kræver selvfølgelig, at selve arkitekturen af systemet gøres modificerbar, hvilket er en svær opgave i sig selv men ikke uløselig. Her kunne man med fordel tage udgangspunkt i objektorienteret udvikling, som kan være medvirkende til at konstruere et modulopbygget system inddelt i klasser, som diskuteret side 29.

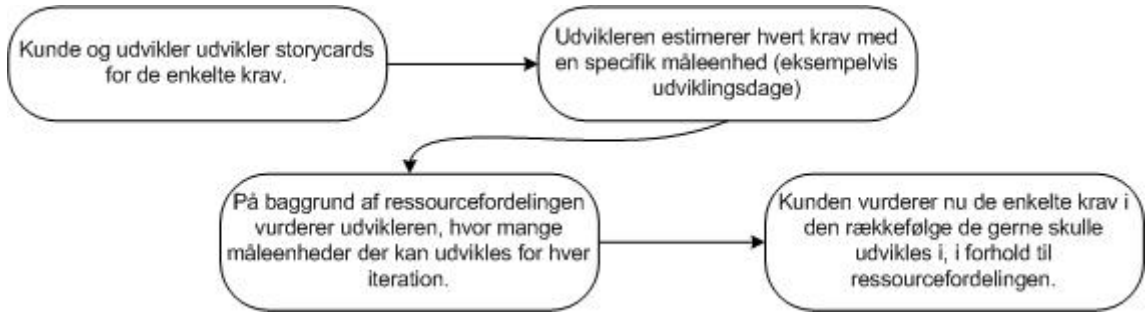
Et andet og særdeles væsentligt aspekt i forhold til specialets fokus er nivelleringen i planlægningsfasen mellem de enkelte funktionssystemer, eksempelvis ledelse og udviklere. Det virker underligt, at medarbejdere ikke er med til at planlægge, hvad der skal laves hvornår, men det er desværre ofte kendetegnende ved topstyrede organisationer. Ansvaret skal i topstyrede organisationer placeres hos en ledende person, som skal styre (ikke lede) medarbejderne under sig. Planlægningsfasen i XP udfordrer denne antagelse ved at sætte medarbejdere

[5] Som sammenligning giver bilag 1 et eksempel på en vandfaldsbaseret udviklingsproces.

[6] Her skal det nævnes, at andre udviklingsmetoder også indeholder kundekontakt, men dette er som regel i opstartfasen, hvor brugeren udelukkende anvendes til at specificere krav. XP bibeholder kundekontakten igennem hele udviklingsperioden og gerne med kunden som en fast del af udviklingsteamet, hvorved XP distancerer sig yderligere fra traditionel softwareudvikling.

ved samme bord som ledelsen og udnytter dermed det slack[7], som i sidste ende kan forbedre produktets konkurrenceevne, fordi hidtil uudnyttet viden indeholdes i designet af systemet.

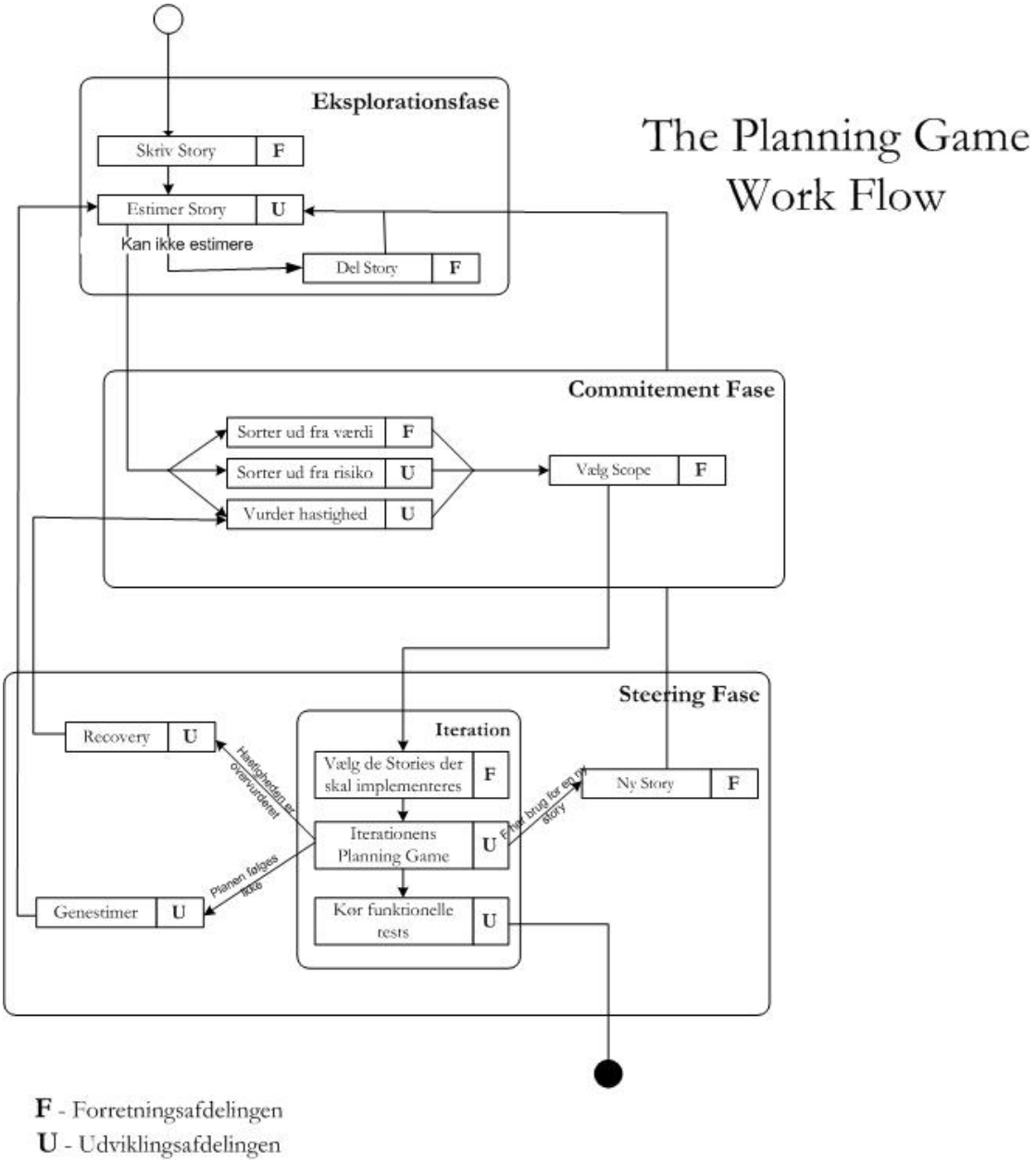
Selve udviklingen af krav anvendes dermed i planlægningsstrategien som et decideret redskab, hvor kravene udvikles i kontinuerlig kontakt med kunden. Denne kontakt gør, at kravene specificeres, så både kunder, udviklere og ledelse kan læse og vigtigst af alt forstå kravene til systemet. For at realisere dette anvendes såkaldte storycards, der beskriver kravene i detaljer, illustreret ved figur 6.



Figur 6 [Baseret på Auer & Miller, 2002, s. 123]

Ovenstående figur illustrerer omskifteligheden i udviklingen af kravene, da kun den første udviklingsiteration indeholder specifikke krav. Da hver iteration ofte løber mellem 1 og 4 uger, er der altså rigelig tid til at udbygge kravene senere. Den korte iterationsperiode er også med til at øge omskifteligheden i udviklingen, da kunden senere skal planlægge rækkefølgen af kravene i næste iteration og dermed planlægger i forhold til det reelle system fra første iteration. Hermed opnås der i første omgang et brugbart system baseret på kundens prioritering af kravene og samtidig en verificerende instans i planlægningsfasen af de resterende iterationer. Figur 7 illustrerer forløbet i The Planning Game. Der skelnes mellem forretningsafdelingen og udviklingsafdelingen på den måde, at forretningsafdelingen står for kundekontakt, planlægning, salg og marketing, mens udviklingsafdelingen står for udvikling, estimering og design (i samarbejde med kunden).

[7] Slack er som nævnt tidligere en benævnelse for de reserver, der ikke er nødvendige at mobilisere i den nuværende situation, men som kan mobiliseres, hvis situationen kræver det [March, 1995, s. 54].



Figur 7 – The Planning Game Workflow

I modsætning til traditionel softwareudvikling, består planlægningsfasen ikke alene af personer fra enten forretningsafdelingen eller udviklingsafdelingen, men er en form for operationel eklektisk kobling af den viden, som begge afdelinger ligger inde med. De involverede parter i selve planlægningsfasen bør følge en række rettesnore, for på den måde at udnytte den iboende viden. Et ufravigeligt krav i denne anledning er gensidig respekt [Beck, 2000, s. 86-87]. Selvom lønforskellen er stor, betyder det på ingen måde, at den bedst lønnede altid har ret. Følges denne regel ikke, er der ikke meget mening i at lave denne tværfaglige kobling. Etableres den gensidige respekt hurtigt, er der i XP en række holdepunkter til at nå forsvarligt gennem planlægningsfasen.

Målet
Målet med The Planning Game er at maksimere værdien af den producerede software, hvormed man kan reducere omkostninger og de implicitte risici i projektet som helhed.

Strategien
Udviklingsteamets strategi er at investere så lidt som muligt og få den mest værdiskabende funktionalitet i produktion så hurtigt som muligt i et kontinuerligt samarbejde med den overordnede designstrategi. Forretningsafdelingen skal derfor relativt hurtigt afgøre i samspil med kunden, hvilken funktionalitet der er mest værdiskabende for kunden, hvorefter udviklingsteamet sætter dette i produktion.

Spillerne
Det kan til tider være svært at overskue, hvem der spiller hvilke roller af de forskellige medlemmer i The Planning Game. Udviklingsafdelingen betegnes derfor som de personer, som er ansvarlige for implementeringen af systemet. Forretningsafdelingen betegnes som de personer, som træffer beslutninger om, hvad systemet skal forestilles at gøre. Forretningsafdelingen skal derfor planlægge og beslutte prioritering, indhold og rammer for systemet. Hvis de er smarte, gøres dette i samspil med både slutbrugere, fokusgrupper og salgsteamet (ganske som intenderet i den skandinaviske udviklingstradition).

Trinene
Der er tre trin i The Planning Game, som overordnet følges af forretningsafdelingen både før og gennem hver udviklingsiteration.

> **Udforskning** – find ud af hvilke nye ting systemet skal indeholde
> **Forpligtelse** – Beslut hvilke mulige krav der skal forfølges i næste iteration
> **Styring** – Styr udviklingen i takt med at virkeligheden modellerer planen
>
> [inspireret af Beck, 2000, s. 89]

Trinene er som regel fulgt i faser, men dette er ikke stringent. Derfor gennemgås de tre trin kort herefter.

Udforskning
Formålet med dette tin er at give en vurdering af, hvad systemet skal indeholde. Udforskningen indeholder to trin. På det første trin skriver forretningsafdelingen en Story, som kort beskriver, hvad systemet forventes at gøre. Storyen skrives på indekskort, som går videre til næste trin. På trin to estimerer udviklingsafdelingen, hvor lang tid den enkelte story vil tage at implementere. Hvis det ikke er lige til at estimere, indgås der et samarbejde med forretningsafdelingen, som enten skal lave en bedre beskrivelse af den enkelte story eller dele storyen op i flere mindre dele, som illustreret i figur 3, side 35. [Beck, 2000, s. 90]

Forpligtelse

Formålet med denne fase er for forretningsafdelingen at fastsætte en dato for den næste release, og derefter for udviklingsafdelingen at validere denne leveringsdato. Denne fase har fire trin. På det første trin vurderes hver story og inddeles i tre kategorier: **1.** funktionskritiske, **2.** mindre funktionskritiske men giver til gengæld en stor kundeværdi, og til sidst **3.** de, der er rare at have, men som ikke er hverken kritiske eller tilføjer nævneværdig værdi til systemet. Herefter vurderes hver story iht. risiko i estimeringen. De enkelte storys inddeles i tre kategorier; de der umiddelbart kan estimeres, de hvor noget kan estimeres, og til sidst de storys, der ikke kan estimeres. Herefter vurderer udviklingsafdelingen, hvor mange kalendertimer der er til rådighed i hver måned. Forretningsafdelingen vurderer herefter i samarbejde med kunden, hvilke storys der skal indeholdes i den næste iteration, enten hvor der tages udgangspunkt i de ønskede storys, eller hvor datoen fastsættes fra begyndelsen. [Beck, 2000, s. 90]

Styring

Styringsfasen har som formål kontinuerligt at opdatere planerne fra de tidligere faser i forhold til, hvad forretningsafdelingen og udviklingsafdelingen har lært. Styringsfasen er inddelt i fire faser. I begyndelsen af hver iteration vurderer forretningsafdelingen, hvilke af de mest værdiskabende storys, der skal implementeres. Disse skal dog udvælges, så der udvikles et funktionelt system, som kan køre fra start til slut, selvom det er ufuldstændigt. Hvis udviklingsafdelingen vurderer, at enkelte storys er overestimerede mht. hastigheden i udviklingen kan de i anden fase bede forretningsafdelingen om at genestimere iterationen, baseret på de nye informationer. I tredje fase har forretningsafdelingen mulighed for at indsætte en ny story i iterationen, såfremt nye krav skulle opstå. Denne kan dog kun indsættes, hvis det vurderes, at den nye story stadig overholder retningslinjerne fra forpligtelsesfasen omkring hastighed og tid. I sidste fase har forretningsafdelingen mulighed for at genestimere den enkelte iteration, hvis det vurderes, at der er et misforhold mellem mål og virkelighed.

Som så meget andet indenfor XP kan de vigtigste fraser holdes indenfor et begrænset spektrum.

> *"We will carefully craft a solution for today's problem today, and trust that we will be able to solve tomorrow's problem tomorrow." [Beck, 2000, s. 97]*

Det betyder operationelt, at dagens tasks er begrænset til det, der faktisk kan nås, og at tasks dermed ikke som udgangspunkt løber over flere dage. Udviklingsstrategien består af fire grundlæggende principper; kontinuerlig integration, kollektivt ejerskab, refactoring og pair programming.

Kontinuerlig integration

Kontinuerlig integration er en strategisk softwaremæssig beslutning, som inddeler softwaresystemet i mindstedele, som kan udvikles, testes og rettes til over en enkelt arbejdsdag. Den kontinuerlige integration udvikler systemet over en funktionel, stabil og testet kerne, hvor der hele tiden kan tilføjes nye elementer i form af funktionalitet eller tilpasning, hvilket ligner principperne for modulariseret softwareopbygning. Den kontinuerlige integration tilbyder den enkelte medarbejder nye daglige udfordringer, som alle skal afsluttes samme dag, hvilket igen giver et inspirerende udviklingsmiljø [Beck, 2000, s. 98]. Det betyder, at aktiviteterne kodning, test, lydhørhed og design bringer en rytme ind i udviklingen lidt i stil med åndedrættet. Først lærer

man (gennem testcases), så koder man, så tester man og så publicerer man. Næste dag kan man så starte med refactoring (rydde op i gårsdagens kode) og gå videre med dagens task. Derved understøtter man enkelhed i både udvikling og design samtidig med, at man afslutter en task og derfor er klar til den næste, hvilket tilfører personlig motivation i projektet.

Kollektivt ejerskab

Det kollektive ejerskab betyder, at hvis der konstrueres en gennemgående forårsrengøring af koden, er ejerskabet af koden kollektivt, så alle har både rettigheder og ansvar for, at koden er så enkel som muligt, samtidig med at funktionaliteten bevares. Eftersom der hele tiden tilføjes mere kode til den grundlæggende funktionelle systemkerne, opstår der nødvendigvis en vis form for redundans, som enten kan fjernes eller redesignes af den eller de, der er ansvarlige for forårsrengøringen. Det kollektive ejerskab viser sig også at mindske indførelsen af uoverskuelig eller kompleks kode, som ikke umiddelbart kan verificeres, og da alle udviklere har rettigheder til at gennemse og redigere koden, tænker man sig nok om en ekstra gang, før man indfører kompleks og uverificeret kode [Beck, 2000, s. 99]. Dermed nedsætter det kollektive ejerskab også projektets iboende risici.

Refactoring

Kort fortalt er refactoring forbedring til systemets design og arkitektur. Refactoring er den instans, som gennemgår den kode, som har kundens umiddelbare opmærksomhed (den enkelte story) og minimerer koden så meget som muligt ved at bl.a. at reducere redundans. Refactoring er særdeles vigtigt, fordi det er med til at sikre, at koden er overskuelig, og at det fortsat er ubesværligt at tilføje ny funktionalitet til det samlede system. Det er selvfølgelig ønskeligt, at koden er så simpel som muligt, men det kræver perfekt kode i første forsøg, hvilket i bund og grund er uopnåeligt. Det er både muligt at lave for lidt og for meget refactoring. For lidt refactoring kan iagttages ved, at det bliver stadigt sværere at tilføje ny funktionalitet, mens for meget refactoring giver udslag i, at det tager længere tid at finde steder, hvor man kan optimere koden. Refactoring er en helt naturlig del af den daglige programmering, da man efter migreringen og testen af de dagligt udviklede komponenter reducerer koden så meget som muligt [bl.a. Marchesi m.fl., 2003, s. 53-54 og 147-148].

Pair Programming

Pair Programming er en udviklingsdisciplin, som kun eksisterer i XP. En af reglerne i Pair Programming er, at der altid er to personer om at skrive koden. Det handler ikke om, at en person skriver koden, mens den anden kigger på, men mere omkring den kontinuerlige dialog i udviklingen af koden. Der er altså et kommunikativt analytisk element, et testelement og et designelement indlejret i Pair Programming. Det analytiske element henvender sig direkte til koden, da hele koden er til kontinuerlig revision af den, som ikke koder og dermed afkræver overblik over hele systemet. I og med at to personer kontinuerligt diskuterer kodestrukturen, er der et designelement i kommunikationen, som hele tiden redesigner både nuværende kode og hele systemets kode [Beck, 2000, s. 100]. Dermed indgår der i Pair Programming en kompleksitetsreducerende faktor, som tester koden, før den er skrevet og dermed reducerer systemets iboende risici. Man kan dog ikke eliminere alle risici i et system, da man ikke kan se, hvad man ikke kan se, men jo flere risici der reduceres på et tidligt tidspunkt i processen, jo bedre.

Pair Programming indeholder også et læringsmæssigt redskab, som er særdeles relevant for at beskrive XP som disciplin. Det er svært at forestille sig, at to udviklere er nøjagtigt lige gode til at kode, så pointen med at programmere i par er ikke kun møntet udelukkende på analyse, design og test men også på læring. I starten vil den ene programmør med al sandsynlighed overtage keyboardet i lang tid ad gangen, mens den anden observerer og interagerer og måske endda overtager keyboardet. Derved bliver den bedre af de to tvunget til at dele sin viden, så diskussionen hen ad vejen bliver bedre og bedre og efter et stykke tid nivellerer vidensniveauet hos de to udviklere og koden bliver dermed meget bedre [Beck, 2000, s. 100-101][8].

Opsummering

XP er en metode, hvormed der opstår et kommunikativt handlerum, hvor viden og ekspertise deles gennem hele organisationen, men dog specielt ved de par som programmerer. Den organisatoriske struktur er også indeholdt i XP, men mere som en forudsætning end en egentlig organisationsudvikling. Er organisationen imidlertid indstillet på at afprøve XP, indgår de grundlæggende værdier i XP som retningslinje, hvorved organisationen nødsages til at indbefatte disse værdier, eventuelt i et begrænset spektrum af organisationen, eksempelvis på projektbasis. Den indbyggede modificerbarhed i struktureringen af udviklingen (specialet kendetegnet ved den påkrævede modularisering af systemet i små dele) gør XP særdeles følsom overfor ændringer i krav til det udviklede system, da der meget hurtigt kan inkorporeres designændringer eller ny funktionalitet, hvormed XP adskiller sig mærkbart fra traditionelle udviklingsmetoder og mere drejer i retning af modularisering og objektorientering. De indlejrede redskaber i XP understøtter både værdierne og målene, hvilket gør XP sammenlignelig med CMM modellens krav, som deraf kan perspektiveres til beslutninger og repræsentation af viden gennem beslutninger i den kommende teoretiske analyse.

[8] Man kan betragte dette læringsfænomen som en form for praksisfællesskab, som bl.a. Wenger beskæftiger sig indgående med [eks. Wenger, 1998. s.179] hvilket kunne være et læringsmæssigt interessant fænomen, men da dette ikke ligger indenfor specialets fokus, udelades læringsperspektivet.

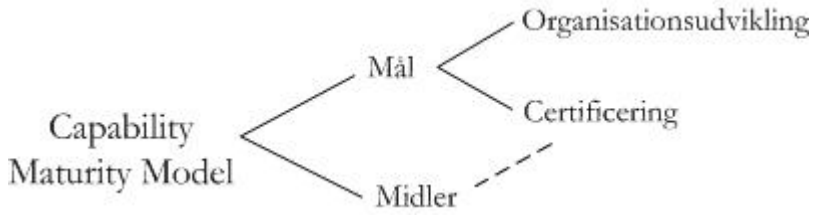
Teoretisk analyse

In anything at all, perfection is finally attained, not when there is no longer anything to add, but when there is no longer anything to take away.
- Antoine de Saint-Exupéry

Med udgangspunkt i CMM modellens enkelte niveauer vil den teoretiske analyse perspektivere de enkelte key process areas (KPA) indeholdt i dette niveau til XP som organisatorisk værktøj. Med fokus på beslutninger og repræsentation af viden gennem beslutninger vil denne teoretiske analyse gøre det muligt at konkludere teoretisk, i hvilken udstrækning grundideen i XP i teorien kan supportere de organisatoriske beslutninger, som er medvirkende til at konstituere softwareorganisationen på et givent CMM niveau. Den teoretiske analyse skal understøtte og inspirere den operationelle del af specialet, hvorfor det er intensionen at omsætte de teoretiske antagelser til praktiske organisatoriske eksperimenter i min egen organisation.

Inden selve analysen er det imidlertid vigtigt at påpege den dobbelte funktion, som CMM modellens KPA har. De enkelte KPA har deres berettigelse på to områder. Dels giver den enkelte KPA en specifik måleenhed, som i dette speciale kan anvendes til konkret at analysere på i forhold til det opstillede teoriapparat, hvilket giver en sand/falsk værdi. Dels giver de enkelte KPA den enkelte organisation konkrete retningslinjer for en bedre softwareproduktion. Det betyder, at en organisation enten kan tage udgangspunkt i CMM modellen for at forbedre eksisterende udviklingsmetoder og resultater eller anvende CMM modellens certificering til at vurdere, hvilket niveau man eksisterer på i forhold til andre softwareudviklingsorganisationer. Denne dobbelthed i KPA funktionen skal ses som CMM modellens styrke, dels fordi modellen åbner op for organisatorisk individuelle tiltag, dels fordi modellen er deskriptiv iht. løsningen og præskriptiv iht. resultatet. Der skelnes altså mellem CMM modellens certificerende intention og CMM modellens organisationsudviklende intention.

Men udover denne dobbelthed i selve KPA funktionen, er der et dualistisk perspektiv på selve CMM modellen, som kan anvendes til at vurdere, hvorvidt en organisation anvender modellen som organisationsudviklende værktøj eller som værktøj til at opnå en given CMM certificering. Nedenstående model illustrerer denne anvendelsesorienterede dualisme.



Figur 8 – CMM modellens dualisme

Som illustreret med figur 8 kan målet med anvendelsen af CMM modellen som organisation være at kvalificere organisationen yderligere i forhold til softwareudvikling, og ikke nødvendigvis gå efter en certificering af organisationen på et givent CMM niveau. Det betyder, at en organisation kan anvende de dele af CMM modellen, som stemmer overens med den organisatoriske praksis og dermed fokusere på de dele, som findes anvendelige og sortere resten fra. Fokuserer man som organisation på en reel CMM certificering, hænger mål og midler sammen, da eksempelvis de kvantitative redskaber og mekanismer, som fordres på de øverste CMM niveauer, skal være en del af organisationen for at opnå en certificering. Man kan ikke fravælge nogen KPA i forhold til det niveau, som man vil certificeres på og er dermed pålagt at anvende de redskaber, som CMM modellen fordrer. Dette er en vigtig distinktion, som vil forfølges gennem resten af specialet. Med disse dobbelte funktioner friske i erindringen kan analysen nu påbegyndes.

CMM niveau 2

Det er hensigtsmæssigt at begynde den teoretiske analyse ved begyndelsen, for dermed at belyse de KPA'er, som skal opfyldes, for at organisationen kan beskrives som værende på niveau 2. For at en organisation skal konstitueres på niveau 2 i CMM modellen, er det tvingende nødvendigt, at alle KPA er opfyldt. Niveau 2 er grundlæggende processuelt orienteret, hvor niveau 3 er af mere institutionaliserende karakter, hvormed niveau 2 er fokuseret på de mere praktiske dele af udviklingsprocessen. Denne del af analysen har til formål at belyse, hvilke komponenter, værdier og redskaber i XP, der kan medvirke til at opfylde de forskellige krav til CMM niveau 2.

Requirement management (RM)

Grundlæggende indeholder RM ifølge Jalote to kriterier:

- *Software requirements are controlled to establish a baseline for software engineering and management activities.*
- *Software plans, products and activities are kept consistent with requirements.*

[Jalote, s. 10]

I al softwareudvikling er kravene til det udviklede system oftest den guideline, som er holdesnoren gennem udviklingsfasen. I modsætning til det, Mathiassen benævner *system definition*, som er en overordnet beskrivelse af systemets funktion, er kravene eksplicitte elementer, som tillader kontrol og ledelse af de enkelte dele af systemet. I CMM modellen er det derfor ønskeligt at styre kravene gennem hele udviklingsprocessen, og dette passer glimrende ind i ideen med XP, hvor krav kontrolleres og styres sekventielt med udviklingen. Begrebet *Requirement Change Management* er en væsentlig bestanddel af CMM modellens niveau 2, og anvendes til at kontrollere, at modificeringen af krav kan styres gennem hele udviklingsfasen.

Når CMM modellen anvender betegnelsen styring i forhold til kontrol, er der tale om en kvantificerbar styring eksempelvis i form af metrikker som det primære ledelsesmæssige redskab, hvormed ledelsen kan kontrollere og styre ressourcer i forbindelse med det enkelte projekt, eksempelvis gennem *Balance Score Cards* (BSC). Den kvantificerbare styring af eksempelvis krav er i XP et forholdsvist ukendt begreb, da kravene løbende estimeres i forhold til den enkelte iteration og således er kausalitetsstyret i forhold til det allerede

udviklede. Selvfølgelig er der en vis form for styring indbygget i XP som organisatorisk disciplin, men det kvantificerbare element i softwareudviklingen er ikke indeholdt i XP.

CMM modellen pålægger ikke udviklingsafdelingen at følge en bestemt metode eller anvende specifikke redskaber for at opnå kontrol med kravene og deres udvikling. Jalote udtaler i denne sammenhæng, at krav uvægerligt vil ændres med tiden, da omverdenen ændrer sig under selve udviklingsperioden, hvorfor kravene må tilpasses denne omverdensændring [Jalote, s. 53]. Dette bryder både med vandfaldsmodellen og spiralmodellen, som begge fryser kravene tidligt i processen, hvormed de er statiske under resten af udviklingsperioden. CMM modellen kræver kun, at de krav der arbejdes med er frosset, mens andre krav er relativt åbne, indtil de indeholdes i den reelle udvikling. Dette passer glimrende ind i tankegangen bag Agile Software Development, der netop sigter mod at udvikle software samtidig med udviklingen af kravene[9].

Organisatorisk betyder det, at XP i sin grundstruktur indeholder håndtering af kravene i hele udviklingsfasen, og denne håndtering sammenholdes med den overordnede udviklingsstrategi gennem de korte iterationer, som kontinuerligt revurderes og ændres i et samarbejde med kunden og dermed afspejler den omverden, i hvilken de er indeholdt. Som beslutningsunderstøttende redskab anvender XP en kognitiv og en proceduremæssig rationalitet, der vurderer kravene kontekstafhængigt på det givne tidspunkt, men samtidig indeholder en tidsmæssig vurdering gennem de relativt korte iterationer. De krav, der under udviklingen er styrende for processen, er udover en guideline også en repræsentation af den temporale tilgængelige viden, og nøjagtigheden af surrogatet for det færdige krav er indeholdt i den kognitive rationalitet, der indeholder den tidsmæssige faktor. Surrogatet er dermed en kompleksitetsreducerende faktor i udviklingssammenhæng, ikke alene omkring kravene men også omkring den omstillingsparathed, som CMM modellen på sigt fordrer.

Der er flere redskaber i XP, som kan understøtte beslutningstagen i softwareorganisationen, eksempelvis Pair Programming, hvor to udviklere (eller beslutningstagere) udformer kode (eller beslutninger) i gensidigt samspil, hvorved der inkorporeres en verificerende instans i udviklingen af enten beslutninger eller kode. Ydermere indeholder de grundlæggende værdier i XP basale kompleksitetsreducerende elementer, som muliggør et mere overskueligt beslutningsgrundlag. Enkelheden og feedbackmekanismen understøtter gensidigt hinanden dels som kompleksitetsreducerende element, dels som eksternt vurderingsgrundlag i forhold til den præsenterede løsning. Derfor må det konkluderes, at RM er indeholdt i XP.

Software Project Planning (SPP)

Grundlæggende indeholder SPP ifølge Jalote tre kriterier:

- *Estimates are documented for use in planning and tracking the project.*
- *Project activities and commitments are planned and documented.*
- *Affected groups and individuals agree to their commitments related to the project.*

[9] Agile Software Development er som tidligere nævnt en grundlæggende udviklingsmetodologi, som er udsprunget fra skaberne af Extreme Programming, og er rettet dels mod udviklingen, dels med organisationen og personerne involveret i processerne.

[Jalote, s. 10]

XP er som alle andre metodologier bundet af nødvendigheden af planlægning og estimering, for at sikre en vis form for kontrol over udviklingen. CMM modellen er grundlæggende opbygget omkring traditionelle softwareudviklingsmodeller som eksempelvis vandfaldsmodellen og spiralmodellen, der i højere grad er afhængig af langtidsplanlægning og langtidsestimering mht. krav, end det er tilfældet med XP. Grunden hertil er de korte iterationer på 1-4 uger, hvor de traditionelle modeller oftest anvender både halve og hele år, hvilket nødvendigvis medfører en bekostelig og relativt usikker estimering og planlægningsstrategi i forhold til XP uanset estimeringsværktøj, herunder trepunktsestimering. Men for at beskrive denne planlægningsfase i forhold til SPP, er det hensigtsmæssigt at henvise til afsnittet omkring udviklingsstrategien (side 35).

De tre kriterier for SPP afkræver altså organisatoriske beslutninger i forhold til Jalotes udlægning, men disse organisatoriske beslutninger er i højere grad lagt ud til udviklerne end til ledere, hvormed man også delegerer ansvar, hvis man vælger XP som udviklingsmetode. Der er med andre ord tale om en substantiv rationalitet i valget af task og valget af løsning, mens der er tale om en proceduremæssig rationalitet omkring det ledelsesmæssige estimerende overblik. Repræsentationerne af disse beslutninger ekspliciteres gennem nedbrydningen af kravene i tasks, hvormed de personlige præferencer kommer til udtryk. Man vurderer i nuet frem for at vente på, at alle krav er specificeret. Fordelen ved at repræsentere viden i form af tasks er en mere klar eksplicitering af det semantiske indhold i det konkrete løsningsforlag, hvorfor feedbackmekanismen i XP kan supportere de umiddelbare løsningsforlag, så man relativt hurtigt kan gennemgå de forskellige løsningsmuligheder. SSP må derfor konkluderes at være indeholdt i XP.

Software Project Tracking and Oversight (SPTO)

Grundlæggende indeholder SPTO ifølge Jalote tre kriterier:

- *Actual results and performances are tracked against the software plans.*
- *Corrective actions are taken and managed to closure when actual results and performances deviate significantly from the software plans.*
- *Affected groups and individuals agree with the changes to commitments.*

[Jalote, s. 10]

SPTO er et af de tydeligste ledelsesmæssige KPA i hele CMM modellens niveau 2. Alene sporing som overordnet redskab har fortrinsvist én funktion, og det er at sikre, at udviklerne rent faktisk udvikler i stedet for at lave så meget andet. Der er altså i højere grad tale om styring end om ledelse, hvilket problematiserer kombinationen af XP og CMM modellen. Sporing i XP er den funktion, som sikrer, at de udviklede elementer kan spores tilbage til specifikke storys og tasks. I XP er selve vurderingen af både performance og opfølgning på den planlagte udviklingsstrategi lagt ud til udviklerne, idet det er udviklerne selv, der estimerer, hvor lang tid en given task tager og selv tager ansvar for de daglige tasks som respons til kundens vurdering af de enkelte tasks. CMM modellen baseres i disse KPA'er på en instrumentel rationalitet, hvor det er beslutningstageren, der lader sin egen overbevisning styre udviklerne, hvor XP på netop dette område anvender proceduremæssig

rationalitet. Selve sporingsbegrebet ændrer sig radikalt fra udelukkende at indbefatte udviklernes performance i CMM modellen til i XP også at indbefatte sporing fra indledende krav til færdigt produkt.

Men selvom XP og CMM modellen har forskellige indgangsvinkler til valget af rationalitet, betyder det ikke, at XP ikke håndterer disse områder i dette KPA. Både resultater og performance håndteres jo netop løbende gennem udviklingen, dels gennem de korte iterationer, dels ved kontinuerligt at afholde planlægningsmøder for at sikre, at den planlagte udviklingsstrategi følges. Dette illustreres bl.a. i figuren på side 32. De berørte personer har dermed ikke alene ret til at foreslå en anden strategi end den valgte, de har også ret til at planlægge realistisk ud fra egne ressourcer. Den eksplicitte repræsentation af viden fremkommer netop på disse planlægningsmøder, hvor man kan vurdere, om strategien følges, og om der skal laves strategiændringer for at følge tidsplanen. Dermed kan planlægningsmøderne betegnes som beslutningstagning om, hvad den iboende viden skal repræsentere, og hvad denne videns praktiske anvendelighed skal føre til [eks. Davis, Schrobe og Szolovits, 1993, s.6].

Beslutningerne er altså ikke kausalitetsstyret af ledelsen i XP, men indeholdt i udviklingskulturen, hvor værdier som feedbackmekanismen bevirker, at planlægningsfaserne gennem jævnlige planlægningsmøder tilpasser både udvikling og de organisatoriske strukturer, så de afspejler virkeligheden, hvormed man muligvis ændrer de tidligere estimater. Organisationen kan gennem den proceduremæssige rationalitet ændre de grundlæggende udviklingsmæssige strukturer, hvorfor ledelse kan betegnes som selvledelse modsat den topstyrede ledelse, som i CMM modellen kan betegnes som styring, hvormed SPTO allerede er indeholdt i XP.

Software Subcontract Management (SSM)

Grundlæggende indeholder SSM ifølge Jalote fire kriterier:

- *The prime contractor and the subcontractor agree to their commitments.*
- *The prime contractor tracks the subcontractor's actual results against its commitments.*
- *The prime contractor and the subcontractor maintain ongoing communication.*
- *The prime contractor tracks the subcontractor's actual performance against its commitments.*

[Jalote, s. 10]

Overordnet er SSM udlicitering af opgaver til eksterne partnere, hvilket ikke umiddelbart er indeholdt i XP, da al udvikling foregår internt i virksomheden. Området er dog en naturlig bestanddel af nyere softwareudvikling, da softwareudviklingsorganisationer ofte specialiserer sig inden for snævre områder, hvorfor det ofte er nødvendigt at hente ekspertise udefra. Det kunne eksempelvis være Java™ udvikling, animering, CMS systemer mm. XP opretholder jo netop sin force ved at opretholde den kontinuerlige kommunikation via en række redskaber og anvender test som en verificerende instans for den daglige udvikling endda før den reelle kodning påbegyndes. Hertil er udlicitering af opgaver en problematisk affære, som er en kontradiktorisk modsætning til XPs kerneværdier; kommunikation, enkelhed, feedback og mod. Umiddelbart faciliterer udlicitering enkelheden, da den interne udvikling undgår at beskæftige sig indenfor områder, hvor kernekompetencerne ikke rækker, men både kommunikation, feedback og mod lider under denne fremgangsmåde. Spørgsmålet er så, om XP tilbyder en håndtering af udlicitering, som kan spille sammen med CMM modellen.

En indgangsvinkel til udlicitering i XP kan være at håndtere udliciteringen som enhver anden udvikling. Der konstrueres en grundlæggende story, som ekspliciterer den overordnede funktionalitet og det overordnede formål. Herefter udarbejdes der en række testcases, som skal sikre, at den udliciterede software opfylder interne krav. Disse testcases kan være den monitorering af den eksterne organisation, som CMM modellen kræver, hvilket fordrer en fortsat kommunikation mellem udviklingsorganisationen og den virksomhed, som udvikler den udliciterede software. Denne form for udlicitering kræver en række forberedende foranstaltninger, da krav skal udredes og testcases skal konstrueres, og kommunikationen virker lige pludselig kunstigt foranlediget, idet formålet er monitorering [McConnell, 1996, s. 492]. En anden indgangsvinkel er intern udlicitering, hvor eksterne udviklere udvikler den efterspurgte software internt i organisationen, hvormed de eksterne medarbejdere indgår i de teams, som allerede eksisterer internt. Såfremt det er muligt at tilrettelægge en kontrakt, som sigter efter en intern udlicitering, er det klart at foretrække, da værdierne i XP bevares. Så kan man som softwareorganisation både monitorere, kommunikere dagligt, teste dagligt på de interne systemer og samtidig give umiddelbar feedback på den udliciterede software og dermed opfylde kriterierne for SSM.

Er der tale om intern udlicitering, arbejdes der med en proceduremæssig rationalitet, da tilgængelige informationer netop er samlet internt i organisationen, selvom der bliver tilknyttet eksterne medarbejdere. Videnrepræsentationen udnyttes dermed i højere grad eksplicit i form af daglige testrutiner, hvis de eksterne udviklere udvikler på samme præmisser som de interne. Ved almindelig udlicitering er der tale om en langt mere substantiv rationalitet, da verifikation af den tilgængelige viden ikke kontinuerligt er til stede, men bekræftes gennem den forholdsvis kunstigt anlagte kommunikation. Repræsentation af viden kan kun fremkomme eksplicit gennem testbare moduler leveret af den eksterne samarbejdspartner, hvilket er særdeles problematisk i forhold til de kommunikative verificerende instanser i XP. Derfor må konklusionen være, at udlicitering ikke er indeholdt i XP og derfor ikke understøtter SSM.

Software Quality Assurance (SQA)

Grundlæggende indeholder Software Quality Assurance ifølge Jalote fire kriterier:

- *Software quality assurance activities are planned.*
- *Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.*
- *Affected groups and individuals are informed of software quality assurance activities and results.*
- *Noncompliance issues that cannot be resolved within the project are addressed by senior management.*

[Jalote, s. 10]

Grundlæggende baseres SQA på redskaber, hvor ledere kan monitorere, at kvaliteten i udviklingen følger den overordnede strategi, og at kvaliteten af den udviklede software indeholder et minimum af fejl. SQA er som alle måleinstrumenter i CMM modellen baseret på en kvantificering af kvaliteten, hvilket ikke umiddelbart er kombinerbart med XP's redskaber. Men SQA er udover den monitorerende funktion også en standardisering af procedurer og aktiviteter, hvilket udmærket kan kombineres med XP.

Ser man overordnet på XP, er SQA faktisk indeholdt i XP's grundlæggende værdisæt, da selve testfunktionen gennem Pair Programming verificerer kvaliteten af den udviklede software. I den almindelige udviklingsfase i XP er udviklingen af testcases en præmis for selve kodningen, hvilket verificerer det high-level design, der kombinerer de mange tasks, som er afledt af de storys, som er udviklet af udviklerne i samarbejde med kunden. Alene redskaberne i udviklingsstrategien (beskrevet fra s. 35) sikrer, at de valgte standarder følges, og at de berørte personer er informerede om ændringer, aktiviteter og resultater.

Det betyder fundamentalt, at XP indeholder metoder og redskaber, som umiddelbart kan sikre, at udviklingsstandarter følges gennem hele processen og dermed opfylder dette KPA. SQA indeholder dog ikke nogen individuel beslutningstagningsdel, da SQA udelukkende er organisatorisk funderet. Dermed er det udelukkende en ledelsesmæssig opgave at sikre, at SQA er indeholdt i udviklingen, hvilket leder blikket hen på den instrumentelle rationalitet, hvor lederne beslutter ud fra personlige præferencer. Beslutningen omhandler udelukkende, om XP skal indføres som generelt udviklingsmetode, og derfor omhandler beslutningen også en personlig overbevisning om den planlagte kommunikative struktur i udviklingen. Hvis udviklingsstrategien funderes på en traditionel udviklingsmodel (illustreret i bilag 1), bør der indføres en række kontrolinstanser, hvor der kan skabes et ledelsesmæssigt overblik over både ressourcer og progression, som igen er lette at kvantificere. Overblikket skabes gennem generelle review og styregruppemøder, men indeholder til gengæld ikke en åben daglig kommunikation mellem ledelse og udviklere, hvilket gennem redskaberne er indeholdt i XP, hvilket som nævnt tidligere kendetegner den proceduremæssige rationalitet.

Den åbne og daglige kommunikation mellem udviklere og ledere bevirker jo netop, at repræsentationen af viden gennem kommunikationen kommer væsentligt tættere på det, den refererer til, da repræsentationen kontinuerligt revurderes og dermed nærmer sig virkelighedens fænomen bedre end månedlige styregruppemøder omkring produktkvalitet, der kun giver et øjebliksbillede, som er frosset indtil næste møde. Kontinuerlig kommunikation skaber dermed mulighed for en mere nøjagtig repræsentation, selvom den fuldstændige repræsentation jo altid er fænomenet selv og derfor uopnåelig i en kommunikativ sammenhæng [Davis, Schrobe og Szolovits, 1993, s.5].

Software Configuration Management (SCM)

Grundlæggende indeholder SCM ifølge Jalote fire kriterier:

- *Software configuration management activities are planned.*
- *Selected software work products are identified, controlled and available.*
- *Changes to identified software work products are controlled.*
- *Affected groups and individuals are informed of the status and content of software baselines.*

[Jalote, s. 10]

Dette sidste KPA på CMM niveau 2 indeholder næsten udelukkende kontrolinstanser for lederne under selve softwareudviklingen og baseres på en lineær udviklingsmodel, hvor kunderne først kommer ind i sidste ende af processen og godkender det endelige design modsat XP, hvor kunderne er med hele vejen. SCM er mest af alt et planlægningsværktøj, der er medvirkende til at konstruere aktiviteter, som systematiserer grundlæggende

procedurer som eksempelvis ændringsstyring, så al ændring af kode dokumenteres for at skabe overblik både for ledere og udviklere. Kontrolinstanserne er i denne henseende ikke kvantificerbare redskaber som i andre KPA'er, men standardiserede procedurer på organisatorisk niveau, som udviklerne skal følge for at producere den nødvendige dokumentation til senere videreudvikling. SCM befinder sig i starten af en traditionel vandfaldsbaseret udviklingsproces, hvor skabeloner til bl.a. versionering, benævnelser og sporbarhed aftales og produceres af ledelsen. Hermed skabes kontinuitet og ensrettethed i udviklingen, og dermed gøres den enkelte medarbejder undværlig, hvilket er et positivt træk i en udviklingsvirksomhed.

I XP er store dele af SCM indeholdt i både redskaber og værdier, men i en helt anden kontekst i den traditionelle softwareudvikling, som CMM modellen bygger på. Kontrolinstanser under selve udviklingen inkorporeres løbende gennem de ugentlige møder, og eftersom iterationerne under udviklingen er korte i forhold til traditionel softwareudvikling, er planlægning og review en mere dynamisk bestanddel i XP end i traditionelle udviklingsmetoder, hvilket skaber både overblik og modificerbarhed i udviklingen. I traditionel softwareudvikling er *user acceptance test* en instans, hvor brugeren kommenterer både funktionalitet og design. Denne instans er dog placeret i sidste del af udviklingen, og grundlæggende designændringer og funktionalitetsændringer kræver som oftest store ændringer i det overordnede high-level design (se bilag 1). Her indfører XP som nævnt tidligere et mere intenst samarbejde med kunden, som efter hver iteration skal forholde sig til det udviklede og endda er med i planlægningsfasen af kommende iterationer. Derfor har udviklerne større sandsynlighed for at ændre grundlæggende i opbygningen af den udviklede software, før kravene betyder alt for komplekse ændringer i systemarkitekturen end traditionelle softwareudviklingsmetoder, hvis alt går efter planen. Dog skal det i denne sammenhæng nævnes, at selv objektorienterede udviklingsmetoder er underlagt klassifikationer og hierarkier, som kan være særdeles problematiske at ændre i. Objektorientering er derfor ikke i sig selv garant for modificerbarhed, men sandsynligheden for at modificeringen kan gennemføres er langt større end med traditionel softwareudvikling. Ændringsstyring til det udviklede kører simultant med selve kodningen, hvor testcases fungerer som daglig verifikation af den udviklede funktionalitet både før og efter kodningen. Større ændringer til systemet fra brugernes side kommer med i udviklingen på linje med tasks fra storycardene, hvor traditionel softwareudvikling i højere grad kører ændringsstyring som selvstændige projekter. Selvfølgelig bliver kritiske tasks prioriteret højere end ny funktionalitet, ganske som i alle andre softwareudviklingsmetoder.

I planlægningsfasen ekspliciterer folkene bag XP, at hele metoden opbygges omkring et testbart system, hvor der løbende tilføjes ny funktionalitet; ganske som i modulariserede systemer. Der er med andre ord altid et funktionelt system, som kunderne kan forholde sig til, hvilket skaber et dynamisk samarbejde med kunden omkring design og funktionalitet i systemet. Det testbare system fungerer dermed som en repræsentation af den semantiske forståelse af de opstillede krav. Det betyder for den overordnede planlægningsfase, at der kun udvikles skabeloner til det, der er nødvendigt på det gældende tidspunkt, hvormed man undgår udvikling af funktionalitet, som alligevel ikke bliver indeholdt i det færdige system. XP indeholder dermed funktioner, som gennem den proceduremæssige rationalitet i forhold til kunden dagligt opstiller repræsentationer af tilgængelig viden og dermed tillader, at opmærksomheden indsnævres til de elementer, som findes relevante i den pågældende situation i den pågældende kontekst. SCM er dermed indeholdt i XP.

CMM niveau 3

Ganske som CMM niveau 2 er niveau 3 en række KPA, som skal dækkes fuldt ud af den organisation, som vil konstitueres på CMM niveau 3. Organisationen skal her på niveau 3 også dække de områder, som eksisterer på CMM niveau 2, da CMM modellen bygger på et inklusionsprincip, hvor nedenstående niveauer er inkluderet i de ovenstående. Det betyder, at den teoretiske analyse burde stoppe her, da SSM fra niveau 2 ikke er indeholdt i XP. Dog vil det af det praktiske eksperiment sandsynliggøres, at SSM praktisk kan indeholdes i XP, hvorfor analysen fortsætter videre. Niveau 3 er som tidligere nævnt af mere institutionaliserende karakter end niveau 2, hvormed blikket rettes mere mod organisationen end mod de processer, der er møntet på selve udviklingen.

Organization Process Focus (OPF)

Grundlæggende indeholder OPF ifølge Jalote tre kriterier:

- *Software process development and improvement activities are coordinated across the organization.*
- *The strengths and weaknesses of the software processes used are defined.*
- *Organization-level process development and improvement are planned.*

[Jalote, s. 10]

OPF er grundlæggende en mekanisme i CMM modellen, som her på niveau 3 forholder organisationen til den interne læringsdel, så organisationen har mulighed for at forbedre og tilpasse sig efterhånden som udviklingen skrider fremad, samtidig med, at det organisatoriske element i udviklingen holdes konsistent. Det betyder, at ledelsen i dette KPA har langt større indflydelse på udviklingen, da det organisatoriske fokus kan skride ind og påvirke udviklingen, hvis udviklingen modstrider den organisatoriske grundstruktur. Hvorvidt denne indgriben skal ændre udviklingen eller den organisatoriske grundstruktur er ikke dette speciales fokus, men det er klart, at der med dette fokus på ændring skabes kohærens mellem metode og organisation, hvilket i sig selv er særdeles interessant. Beslutningerne i denne indgriben er udelukkende afhængige af den respektive organisation, hvormed det ikke kan konkluderes endeligt hvilken vej, der er den mest farbare.

Her står det utroligt klart, at den interne strukturelle del af XP opfylder dette KPA, da XP vha. værktøjer som Pair Programming holder en konstant progression omkring udvikling af både redskaber og metoder. Alle personer i udviklingen har mulighed for at komme med forbedringsforslag, som så adresseres af de organisationsansvarlige; ledelsen. Ud over det overordnede organisatoriske perspektiv indeholder XP's redskaber ligeledes et individuelt læringselement, da de par, som programmerer, hele tiden bliver bedre, og kompetencerne mellem de to programmører nivelleres, så parrene hele tiden tilpasses hinanden. Hermed har organisationen selv på mindste niveau mulighed for at tilpasse udviklingen, gøre opmærksom på mulige forbedringer, og påpege fejl eller mangler i softwaren, som så kan adresseres i næste iteration. I og med, at iterationerne holdes mellem en og fire ugers længde, har en organisation baseret på XP stor mulighed for hurtigt at tilpasse nye elementer i udviklingsstrukturen og planlægge dette til hurtigt at blive inkluderet i kommende iterationer.

Men hvis organisationen er hierarkisk opdelt med klare regler og procedurer omkring udvikling (som de fleste udviklingsvirksomheder er), er XP ikke den metode, som fordrer kohærens mellem organisationen og udviklingen. Tværtimod skal der velvilje til hos ledelsen for at inkorporere en så omskiftelig struktur, som XP foranlediger. Er denne velvilje imidlertid til stede, er der ingen tvivl om, at XP understøtter den organisation, den anvendes i, fordi ledelsen kan holde XP afdelingen som en selvstændig afdeling i organisationen, også selvom organisationen er hierarkisk opbygget.

I og med at dette KPA har fokus på organisationen, kan blikket nu i større grad end tidligere rettes mod ledelsen, da det organisatoriske overblik ikke ansvarsmæssigt ligger hos medarbejderne. Beslutningskompetencen omkring udviklingen får dermed et organisatorisk perspektiv i den indeværende kontekst i dette speciale. Det betyder, at XP fordrer en proceduremæssig rationalitet, idet strukturen i XP bevirker en refleksiv behandling af beslutningerne og tilpasser både beslutninger og kontekst for at skabe kohærens. Dermed skabes der en ontologisk repræsentation af organisationen, hvormed der skabes en særdeles vigtig organisatorisk kompleksitetsreduktion, som gør, at ledelsen får bedre overblik over sammenhængen mellem organisation og udviklingsmetode og dermed sandsynligvis kan træffe bedre beslutninger.

Som videnrepræsenterende medie er OPF det redskab, som via de specifikke krav til dokumentation af styrker og svagheder i procedurerne, muliggør en eksplicitering af den indlejrede viden omkring procedurer. Det betyder, at dokumentationen referer til specifikke objekter i verden, hvilket i meget høj grad nærmer sig virkelighedens fænomen, og der er dermed noget at forholde sig til i procesforbedringsfaserne. Igen er der et verificerende element med i videnrepræsentationen. Hvis det ikke kan repræsenteres via dokumentation, så er der ikke noget håndterbart problem. Hvorvidt det semantiske indhold i dokumentationen er umisforståeligt er afhængigt af repræsentationen, men hvis det som i XP er modificerbart, så kan det semantiske indhold ændres, så det med tiden er så tæt på virkeligheden som muligt.

Organization Process Definition (OPD)

Grundlæggende indeholder OPD ifølge Jalote to kriterier:

- *A standard software process for the organization is developed and maintained.*
- *Information related to the organization's standard software process by the software projects are collected and reviewed and made available.*

[Jalote, s. 11]

Som softwareorganisation er det utroligt vigtigt at have en overordnet beskrivelse af de praksisser, som softwareudviklingen skal følge gennem hele udviklingen. OPD afkræver dette på to områder. Der skal være en standartproces, som beskriver den generelle praksis omkring udviklingen og en database, som overvåger denne praksis. Det betyder, at der skal være kvantificerbare elementer med i dette KPA samtidig med, at der er en beskrivelse over praksis, og at denne er dokumenteret.

Umiddelbart tilbyder XP igennem hele udviklingsprocessen en række praksisser, som giver en standardiseret udviklingsskabelon over hele organisationens udvikling, hvilket falder godt i tråd med kravene i OPD, men

den kvantificerbare størrelse er et krav, som kun fordrer kompleksitet, hvilket strider mod en af de fire værdier i XP; enkelhed. Dokumentation i XP er kun til stede for at give overblik over det udviklede system, hvilket betyder, at dokumentation minimeres i forhold til traditionel udvikling. Selv i større organisationer er dokumentation med XP som udviklingsmetode holdt på et meget lavt niveau, og energien er dermed lagt i udviklingen og ikke i dokumentationen. Derfor kan OPD kun opfyldes delvist med XP som organisatorisk udviklingsmetode, da det ikke er muligt at inkorporere kvantitative metoder i en kvalitativ metode, som XP inderst inde er, idet XP som nævnt tidligere primært struktureres og anvendes vha. kommunikative strukturer.

Som beslutningsunderstøttende værktøj kan OPD underbygge den substantive rationalitet, idet det umiddelbart kan antages, at ingen mangel på information gør sig gældende. Dette er en særdeles uheldig virkning, idet beslutningstagere kan vælge at handle udelukkende på baggrund at de kvantitative redskaber, som eksempelvis en database. Dette kan i høj grad medvirke til, at fokus fra udviklerne lægges i dokumentationen, hvorfor XP mister en del af sit moment i selve udviklingen. Der er en problematisk virkelighed, som kun kan håndteres af en kompetent ledelse, der drejer beslutningerne mere i retning af en kognitiv rationalitet eller en proceduremæssig rationalitet, hvor den refleksive behandling af informationerne får mere vægt. I de forskellige bonusordninger er *Balance Score Card* (BSC) dog et vigtigt værktøj, som bygger på denne substantive rationalitet, hvor bonus netop kan begrundes i det kvantitative element som eksempelvis en database. Denne database kan dog modificeres, hvis produkter eller tjenester kommer i produktion tidligere end planlagt men med væsentlig flere fejl. Hermed opnår medarbejderen en ekstra bonus, idet BSC ikke indeholder krav om kvalitet, kun om ressourcer og tid. Dette er en problematisk virkelighed, som XP ikke kan afhjælpe. Det er en ledelsesmæssig opgave at vurdere, hvorvidt vægten skal lægges på dokumentation eller kvalitet, for XP tilbyder i højere grad kvalitet i udviklingen end i dokumentation.

Som videnrepræsentation er OPD problematisk, idet kravet lægger op til en fuldstændig repræsentation af virkeligheden, hvorpå ledere kan kvantificere og konkludere, men denne repræsentation bør kun være guideline til en mere kvalitativ vurdering, idet kvantificerbare elementer i deres grundstruktur ikke i særlig høj grad er i stand til at repræsentere virkeligheden. Der er med andre ord stor sandsynlighed for, at kvantificeringen kan repræsentere flere semantiske virkeligheder alt afhængigt af individ, hvilket er problematisk som beslutningsunderstøttende værktøj. Indlejres der derimod kvalitative elementer som dokumentation, er der en større sandsynlighed for en bedre repræsentation af virkeligheden, men denne kan kun tilnærmes, så længe repræsentationen er dynamisk og ikke statisk, som intenderet i OPD. Det betyder, at OPD kun delvist er indeholdt i XP som organisatorisk værktøj.

Training Program (TP)

Grundlæggende indeholder TP ifølge Jalote tre kriterier:

- *Training activities are planned*
- *Training for developing skills and knowledge needed to perform software management and technical roles is provided.*
- *Individuals in the software engineering group and software-related groups receive the training necessary to perform their jobs.*

[Jalote, s. 11]

I stort set alle organisationer har medarbejdere mulighed for at videreuddanne sig, så de bedre kan udfylde deres rolle eller måske uddanne sig til at udfylde en helt anden rolle. TP er den funktion i CMM modellen, som sikrer denne interne uddannelse i organisationen. Pair Programming, som er et meget essentielt redskab i XP, har intern træning indlejret, men kun inden for specifikke områder. I opstartsfasen af et udviklingsforløb er der som oftest en udvikler i et par, som er bedre end den anden. Med Pair Programming udlignes dette niveau, så begge udviklere inden for relativt kort tid fungerer på lige vilkår, hvilket betyder, at der er en vis form for træning indlejret i Pair Programming. Men denne interne træning er ikke fyldestgørende for TP, som kræver, at træning og videreudvikling er en stabil og bestandig del af organisationen. Hertil er XP som udviklingsmetode ikke fyldestgørende nok, selvom XP tilbyder en vis træning med Pair Programming. TP er et organisatorisk element, som ganske vist ikke ligger langt fra XP, men som kun kan siges at være delvist indeholdt i XP, da XP ikke i særlig stor grad omhandler organisationen, men mere individet og processer omkring udvikling.

Som beslutningsunderstøttende element er hverken TP eller XP redskaber, som kan siges at være tilknyttet en bestemt rationalitet, da det er en subjektiv overbevisning, der afgør, om videreudvikling af færdigheder og mere uddannelse er påkrævet. Men i og med at XP indeholder Pair Programming, er der i XP indeholdt et redskab, som kan afgøre, om den enkelte har brug for mere uddannelse, da den ene af et par hele tiden monitorerer den andens færdigheder, som jo også er en del af samarbejdet. Det er altså tale om en ledelsesmæssig beslutning baseret på en refleksiv behandling i den specifikke kontekst, hvilket henleder opmærksomheden på den proceduremæssige rationalitet. Dette er dog ikke fyldestgørende, da der kan komme en ledelsesmæssig beslutning omkring indførelsen af nye standarter, som eksempelvis ".net (dot-net)[10]", som påkræver yderlige uddannelse af medarbejderne. Dette kan udmærket baseres på en substantiv rationalitet, hvis lederen ikke har sin daglige gang i udviklingsafdelingen og dermed ikke har overblik over de interne kompetencer. Dette er dog et problem, som hverken XP eller TP kan håndtere. XP tilbyder altså kun videreuddannelse som en del af Pair Programming, hvilket kun nivellerer afdelingen og ikke videreuddanner.

Hvis man i denne sammenhæng vurderer Pair Programmings kommunikative redskaber, så er der ingen tvivl om, at Pair Programming i meget høj udstrækning kan give en særdeles god repræsentation af eventuelle manglende færdigheder i selve udviklingsafdelingen, såfremt det ønskes fra ledelsen. Det tætte samarbejde i udviklingsafdelingen kan give et meget nøjagtigt billede af den iboende viden fra de forskellige udviklere, hvilket igen kan give ledelsen et reelt billede af de ressourcer, som skal stilles til rådighed for at videreuddanne afdelingen. Denne repræsentation af viden er selvfølgelig kommunikativt anlagt og dermed underlagt iboende risici for forskellige semantiske opfattelser. Hertil kan de daglige møder i udviklingsafdelingen være medvirkende til en mere nøjagtig repræsentation af den enkelte afdelings krav om videreuddannelse. XP og TP er altså ikke kontradiktoriske, og de indledende manøvrer for at afdække uddannelsesbehovet i afdelingen er til stede, men den egentlige uddannelse er ikke dækket af XP, hvilket betyder, at TP kun er delvist dækket af XP.

[10] Dot-net er en udviklingsstandard, hvor enkelte applikationer kan konfigureres, så forskellige applikationer lettere kan tale med hinanden.

Integrated Software Management (ISM)

Grundlæggende indeholder ISM ifølge Jalote to kriterier:

- *The projects defined software process is a tailored version of the organization's standard software process.*
- *The project is planned and managed according to the projects defined software process.*

[Jalote, s. 11]

Umiddelbart er ISM et organisatorisk KPA, der udelukkende tilfredsstiller organisationens behov for kvantificerbar dokumentation, men da ideen er at vurdere XP som organisatorisk metode til at supportere organisatoriske beslutninger til at konstituere softwareorganisationen på et givent CMM niveau, er dette KPA særdeles interessant. ISM afkræver nemlig, at der opstilles en defineret softwareproces, som er udsprunget af den organisatoriske strukturering, hvilket betyder, at kombinationen af CMM og XP i dette KPA finder sin mulige begrundelse. ISM kræver, at der er kohærens mellem organisation og udviklingsmetode, hvilket dette speciale jo netop forfølger operationelt.

Dog må man sige, at ISM som enkeltstående KPA ikke finder sin begrundelse i XP's metoder. Målet med at anvende XP som organisatorisk metode er så vidt vides ikke afprøvet før, idet XP er metodisk og processuelt anlagt, hvilket er delvist uafhængigt at de organisatoriske processer. Det er netop derfor, at XP kan anvendes af stort set alle softwareorganisationer og dermed er organisatorisk universel. De enkelte softwareprocesser er både planlagt og håndteret i henhold til en defineret softwareproces, men denne tager på ingen måde højde for den organisatoriske strukturering. Derfor kan ISM ikke siges at være indeholdt i XP. Det interessante vil være at observere, om kombinationen af XP og CMM gør dette KPA opnåeligt, hvilket er muligt, hvis organisationen struktureres vha. XP. ISM er dermed møntet mere på styring end ledelse, hvilket ganske naturligt henleder opmærksomheden på den kognitive rationalitet.

Hvis det antages, at IMS er udsprunget af organisationens standartproces, så repræsenterer IMS et særdeles virkelighedsnært billede, som hænger organisatorisk sammen. Så er spørgsmålet bare hvilken repræsentation, der er hensigtsmæssig. Hertil vil XP som organisatorisk strukturelt værktøj foreslå en modificerbar struktur, som kan ændres løbende, og som dermed afspejler den indlejrede viden i øjeblikket. Det kræver dog en kontinuerlig monitorering af denne struktur, hvorefter det så er op til ledelsen at monitorere dette med input fra de implicerede parter og dermed indeholder denne struktur en verificeret semantisk repræsentation. Hertil er det funktionelle system et godt sted at starte, idet det sikrer faste holdepunkter for diskussionen. XP kan således indeholdes partielt i en hierarkisk organisation så vidt som en flad organisation, hvormed XP beviser sine universelle egenskaber, men at antage at XP bygger direkte ovenpå organisationens standardproces er ikke et iboende krav i XP og kan derfor ikke siges at understøtte ISM.

Software Product Engineering (SPE)

Grundlæggende indeholder SPE ifølge Jalote to kriterier:

- *The software engineering task are defined, integrated and consistently performed to produce the software.*

- *Software work products are kept consistent with each other.*

[Jalote, s. 11]

SPE er et af de KPA, som fuldstændigt er dækket af XP som udviklingsmetode. De specifikke opgaver tilrettelægges i *The Planning Game*, opgaverne integreres dagligt i det overordnede udviklingsmøde og anvendes gennem hele udviklingsprocessen. Konsistensen holdes gennem hele udviklingen, da alle følger samme metode med de samme redskaber. Dette KPA er derfor fuldt dækket i XP, idet XP grundlæggende baseres på kommunikation, som da også er en af de fire grundlæggende værdier. Kommunikationen medfører jo kohærens mellem udviklerne, der fortløbende planlægger iterationer sammen med kunden, og tager derfor vare på mulige problematikker, så produktet i sidste ende bliver både konsistent og modificerbart. Konsistensen i XP kommer i høj grad til udtryk i det funktionelle system, som er endnu et krav for XP. Der skal hele tiden være en funktionel kerne, som man hele tiden videreudvikler. Dette vil ikke være muligt, hvis der ikke er enighed om de forskellige tilgange til udviklingen, herunder API'er (håndtag) til de forskellige moduler.

Det betyder rent beslutningsmæssigt, at SPE bygger på en proceduremæssig rationalitet, hvor genstandsområdet afdækkes fra kompetente interessenter, og at der gennem kommunikationen og samarbejdet sikres en refleksiv behandling, inden der træffes en beslutning. Resultatet af beslutningerne igennem udviklingen verificeres i det kontinuerlige funktionelle system, hvorefter de enkelte elementer eller det samlede system kan modificeres gennem redskaberne i XP. Her er refactoring et vigtigt redskab, der ikke alene verificerer koden, men også reducerer koden, så den indeholder mindst mulig redundans. Dette underbygger selve konsistensen i produktet som helhed.

Som videnrepræsenterende medie er SPE et virkemiddel, som kommer så tæt på virkeligheden som overhovedet muligt. Den kontinuerlige opdatering af det funktionelle system er den verificerende instans, som sikrer, at alle kan referere og diskutere på samme præmisser. Dette betyder, at kommunikationen er direkte foranlediget af observerbare objekter i verdenen, frem for diffuse objekter med varierende semantisk forståelse. Der er med andre ord en stor chance for, at individer diskuterer på samme præmisser, idet den semantiske repræsentation af systemet (eller koden) muliggør henvisninger til konkrete objekter, hvilket også er intensionen med anvendelsen af det funktionelle system. Specielt i samarbejdet med kunden, hvor både krav og funktionalitet er til kontinuerlig debat, sikrer det funktionelle system holdepunkter i diskussionen, hvilket formentlig medfører både forbedringer og videreudvikling.

Intergroup Coordination (IC)

Grundlæggende indeholder IC ifølge Jalote tre kriterier:

- *All affected groups agree to the costumer's requirements.*
- *All groups agree to the commitments between different groups.*
- *The groups identify, track and resolve intergroups issues.*

[Jalote, s. 11]

IC er ganske som SPE et KPA, som er fuldt dækket af XP som udviklingsmetode: De elementer, som er med til at opfylde IC er bl.a. kontinuerlig kontakt med kunder (som sidder sammen med udviklerne eller har en repræsentant hos udviklerne). Udviklingen i XP drejer sig jo netop om at skabe software, som opfylder kundens behov, hvorved der kontinuerligt og konstant skal være kommunikation mellem udviklere og kunder. I selve testfasen af softwaren (som gennemløbes dagligt), har kunden altid et funktionelt system, som repræsenterer de krav, kunden har. Ydermere er kunden også med til at bestemme, hvilke tasks og stories der skal opfyldes i den enkelte iteration, hvilket betyder, at udviklerne altid arbejder med reelle opgaver, som forbedrer produktet. Ligeledes er Pair Programming medvirkende til at skabe kohærens og stabilitet i udviklingen, da problemer i de forskellige grupper eller i de forskellige par adresseres umiddelbart og effektivt. XP's fokusering på kommunikation som aktivt redskab og grundlæggende værdi er igen medvirkende til at koordinere udviklingen omkring produktet, men også på procesforbedring. Dette skaber så igen engagement omkring produktet såvel som procesforbedringer.

Nu skal man dog huske på at CMM modellen er udviklet på baggrund af traditionelle udviklingsmetoder som eksempelvis vandfaldsmodellen, hvorfor de interrelaterede interessentgrupper er adskilt med større skillelinjer end det er tilfældet i grupper som arbejder med XP. Det betyder ledelsesmæssigt, at IC har rod i en hierarkisk beslutningsstruktur, hvilket kan resultere i en mere substantiv rationalitet end den proceduremæssige rationalitet, som XP tilbyder. Der er ingen tvivl om, at den udvidede kommunikation mellem ledere og udviklere i de forskellige grupper er medvirkende til en bedre verificering af beslutninger uanset om man kigger på IC eller XP. IC lægger i højere grad op til en forbedring af den interne kommunikation, hvilket kun kan medføre en bedre verificering. Derfor er det sandsynligt, at IC stræber mod en proceduremæssig rationalitet og dermed følger trop med XP.

IC fordrer en kontinuerlig opfølgning af accept af kundens behov, og da kunden er placeret sammen med udviklingsteamet i XP strukturen, er der mulighed for at følge op på de områder, der er dårligt repræsenteret, hvilket gør, at repræsentationen af viden hele tiden kan specificeres, indtil alle er enige om det semantiske indhold. Her er Storycards medvirkende til yderligere verificering af krav fra enten kunde eller forretning, og da de enkelte storys er inddelt yderlige i tasks, er der altså mulighed for en detaljerigdom, der kommer så tæt på virkelighedens objekter som overhovedet muligt, og IC må derfor siges at være indeholdt i XP.

Peer Reviews (PR)

Grundlæggende indeholder PR ifølge Jalote to kriterier:

- *Peer review activities are planned.*
- *Defects in the software work products are identified and removed.*

[Jalote, s. 11]

PR er i CMM modellen med til at sikre, at der udføres review på både proces og produkt. I traditionelle softwareudviklingsmetoder er reviewfasen en del af softwareudviklingen men forgår udelukkende i projektets afslutning (se eks. Bilag 1), og ikke som i XP gennem hele processen. Pair Programming er endnu engang det redskab, som dokumenterer, at der udføres review på koden under hele udviklingen, da den, der ikke taster,

hele tiden har et overordnet overblik over koden og kan sige til, hvis koden ikke refererer til de rigtige elementer, laver forkerte forespørgsler eller skriver kode, der allerede er skrevet en gang. I den fase, som betegnes refactoring, gennemgås hele koden for at reducere redundans og maksimere effektiviteten og konsistensen af koden. Dermed er der i XP indlejret flere elementer af review, hvilket opfylder PR til fulde.

Specielt reviewfunktionen er som verificerende element et redskab, der underbygger den proceduremæssige rationalitet i PR, hvilket er i fuldstændig overensstemmelse med refactoringfunktionen i XP, selv om denne er mere operativt anlagt, idet refactoring er udviklernes redskab til at køre review på produktet. Faktum er dog, at PR er til stede i både The Planning Game og i den enkelte iteration.

Som videnrepræsenterende værktøj kan PR ikke tilføje den store værdi som enkeltstående værktøj, men i samspil med de andre krav i CMM modellens niveau 3 sikrer PR den reviewfunktion, som er indlejret i XP. Reviewfunktionen har jo ligeledes den funktion at opdatere den tilgængelige viden, og den sikrer dermed, at alle områder gennemløbes med et vist interval, så repræsentationerne af viden hele tiden afspejler de nyeste ændringer. Specielt er det interessant at se sammenhængen mellem PR og OPF, da det jo også er vigtigt at køre review på de organisatoriske elementer i enhver organisation for at sikre konsistens mellem idé og praksis. Her indfører PR et verificerende element, som ikke er indlejret i XP som organisatorisk virkemiddel, hvilket kan gøre, at organisationen har mulighed for at tilrette procedurer og redskaber og dermed i højere grad tilnærme sig repræsentationer af virkeligheden frem for repræsentationer af ideer.

CMM niveau 4

Det står klart, at alle KPA på niveau 3 ikke er fuldt dækket, og dermed umuliggøres en reel CMM certificering på dette niveau, hvilket burde standse analysen her. Imidlertid sandsynliggør det praktiske eksperiment (s. 73), at SSM og ISM praktisk kan indeholdes i XP, hvorfor analysen fortsætter videre. Ud over de krav, der er til niveau 3, indeholder niveau 4 i CMM modellen en kvantitativ forståelse af processernes egenskaber og formåen, hvormed det er muligt kvantitativt at forudse og kontrollere processerne i de enkelte projekter. Det er interessant, at der fra niveau 4 indføres kontrollerende instanser, som i den grad fjerner ansvar fra medarbejderne og pålægger ledelsen dette ansvar. Dette bratte skrift i intentionaliteten på niveau 4 (og senere igen på niveau 5) diskuteres og problematiseres senere i diskussionen s. 85. Men så snart denne kvantitative kontrolinstans er til stede, er det ifølge Jalote muligt for organisationer at forbedre både processer, metoder og redskaber og herefter vurdere disse forbedringer kvantitativt. Derfor er der i forhold til de tre tidligere niveauer i CMM modellen et langt kraftigere ledelsesrettet fokus med indførelse af kvantificerbare kontrolinstanser her på niveau 4, som nedenstående analyse af de enkelte KPA illustrerer. Nedenstående citat af Jalote illustrerer indgangen til CMM niveau 4 og samtidig det fokus, som lægges i anvendelsen af KPA på både CMM niveau 4 og 5:

> *Quantitative quality management has two key aspects: setting a quantitative quality goal and then managing the software development process quantitatively so that the quality goal is met. (…) To achieve this goal, it is essential to predict the values of some parameters at different stages in the project such that controlling these parameters during the projects execution will ensure that the final product has the desired quality. If such predic-*

> *tions can be made, then the actual data garnered during the execution of the process can be used to judge whether the process has been applied effectively.*
>
> [Jalote, s. 148]

Quantitative Process Management (QPM)

Grundlæggende indeholder QPM ifølge Jalote tre kriterier:

- *The quantitative process management activities are planned.*
- *The process performance of the project's defined software process is controlled and defined.*
- *The process capability of the organization's standard software process is known in quantitative terms.*

[Jalote, s. 12]

I store træk omhandler QPM ikke så meget selve ledelsesdelen, som det strukturelle aspekt i dataindsamling for at få kvantitative data omkring udviklingen. QPM er et kvantitativt redskab, som inddeler den lineære projektudvikling i målelige dele, som eksempelvis software kvalitet, produktivitet, tidsplan, fejlretningsprocent o.l. Disse data opdateres løbende igennem softwareudviklingsprocessen og anvendes ledelsesmæssigt til at skabe overblik over bl.a. allokering af ressourcer. QPM er i CMM modellen det kvantitative redskab, som på sigt kan hjælpe organisationen til at forbedre og videreudvikle eksempelvis videndeling på baggrund af kvantitative resultater fra tidligere projekter. Specielt spiller QPM sammen med Process Change Management (PCM) på niveau 5 (s. 65), da PCM til dels afhænger af den kvantitative metrik, som skabes i QPM.

Som videnrepræsenterende værktøj er QPM problematisk, idet det repræsentationelle udtryk vil få en anden semantisk værdi for den enkelte iagttager, eftersom det ikke længere refererer til fænomenet men til en bestemt konception af eller holdning til fænomenet, hvilket klar henviser til en substantiv rationalitet. Spørgsmålet er, om QPM overhovedet er i stand til at repræsentere semantisk værdi. Det er umiddelbart påkrævet, at organisationen er bevidst om den semantiske tolkning af data, hvorfor organisationen vil være nødsaget til at opsætte specifikke og udførlige regler for dataindsamlingen, så det repræsentationelle udtryk får en sigende semantisk værdi, som kan anvendes efterfølgende. Og det er netop her, at QPM bliver farlig, for håndteres dataindsamling på et spinkelt grundlag, kan de indsamlede data ikke give et tilnærmet billede af virkeligheden, som er intensionen, men et falsk billede bygget på falske forudsætninger. Håndteres QPM derimod forsvarligt, er der angiveligt en vis semantisk værdi at hente, men det er tvivlsomt, i hvilket omfang disse værdier kan anvendes organisatorisk til at skabe merværdi, i forhold til at anvende kommunikative redskaber. Det skal kort sagt kræve sin leder udelukkende at sætte sin lid til kvantitative data uden kommunikativ rygdækning. Med QPM som enkeltstående værdisæt, finder jeg det ikke forsvarligt at træffe en beslutning med QPM ud fra de præmisser, som March sætter op (se side 21).

Dette kvantificerbare aspekt er på ingen måde indeholdt i XP, selvom selve videndelingen er til stede i XP gennem både udviklingen af storys og gennem Pair Programming. QPM bygger jo netop på en substantiv rationalitet, hvor alt kan og skal beskrives fuldstændigt, hvilket ikke er kompatibelt med XP's proceduremæssige rationalitet. Det må konkluderes ganske kort, at QPM er et redskab, som vil få det særdeles svært i et XP

udviklingsmiljø, og at metrikkerne, som er målet med QPM, ikke er indeholdt i XP's grundstruktur, værktøjer eller værdier.

Software Quality Management (SQM)

Grundlæggende indeholder SQM ifølge Jalote tre kriterier:

- *The projects software quality activities are planned*
- *Measurable goals for software product quality and their priorities are defined.*
- *Actual process toward achieving the quality goals for the software products is quantified and managed.*

[Jalote, s. 12]

Dette KPA sigter mod at gøre organisationen i stand til løbende at vurdere stabiliteten i det udviklede system gennem kvantificerbare målinger, som ofte vil være baseret på softwarefejl. Det betyder, at projektledelsen for det enkelte projekt skal have klare mål og delmål fra begyndelsen af projektet, så man har noget at måle op imod senere i processen, når den reelle vurdering skal finde sted. De kvantificerbare elementer giver den enkelte leder en mulighed for kontinuerligt at holde øje med, hvor langt projektet er kommet, hvor mange fejl systemet har produceret og hvor mange fejl der er blevet adresseret og løst på et givent tidspunkt i processen [Jalote, s. 145-149]. SQM søger et lighedstegn mellem fejlrettelsesprocenten og kvaliteten i softwaren, hvilket i sig selv er temmelig interessant, idet kvalitet søges opnået gennem kvantitative tiltag, som kun kan måles vha. antallet af fejlrettelser. SQM sætter sin lid til en sikker estimeringsprocedure, som ofte er trepunktsestimering. Disse estimeringer skal så løbende tilpasses og kontrolleres igennem udviklingsprocessen, hvilket klart er en ledelsesmæssig opgave.

SQM illustrerer med største tydelighed, at CMM modellen er struktureret og udviklet på baggrund af lineære udviklingsprojekter, meget lig den, som er illustreret i bilag 1. Denne fremgangsmåde kræver en meget lang og detaljeret forundersøgelsesfase, som ikke er indbefattet i XP som sådan. The Planning Game kommer tæt på men er for lidt detaljeret til at kunne anvendes kvantitativt. XP kan uden problemer indlejre kvantificerbare elementer, som vurderer, hvor langt den enkelte iteration er kommet, men at sætte lighedstegn mellem fejlrettelsesprocenten og kvaliteten i software tager på ingen måde hånd om arkitekturmæssige omstruktureringer, der jo i XP er en af de præmisser, som gør koden overskuelig, og hvor det gennem refactoring gøres lettere at tilføje ny funktionalitet til det samlede system. Hvis der i lineære projekter tilføres ny funktionalitet i midten af kodningsprocessen, kræver det, at grundlaget for en SQM vurdering revideres, og at nye målepunkter fastlægges. Dette er selvfølgelig en tidskrævende proces, som XP udviklere ganske givet vil betegne som KISS (Keep It Simple, Stupid). XP arbejder jo netop med korte overskuelige iterationer, hvor alle storys og tasks er tilgængelig for alle, og hvor de udviklede elementer er gennemtestede. Opstår der fejl i det udviklede, oprettes der en task, som håndterer denne fejl, hvorefter udviklerne kan fortsætte migreringen eller udviklingen af nye tasks.

Som videnrepræsenterende værktøj er SQM underlagt de samme problemer som QPM, idet de kvantitative data ikke nødvendigvis giver et tilnærmet billede af virkeligheden. Igen skal lederne være opmærksomme på de regler, de opsætter for dataindsamlingen, så præmisserne for en given beslutning er på plads. Igen må

spørgsmålet stilles om de kvantitative udtryk, som enkeltstående element i en beslutningsproces, er en fornuftig basis for en reel beslutning.

Det virker klart som om SQM bygger på en særdeles substantiv rationalitet, dels fordi SQM forudsætter en lineær udviklingsproces, dels fordi SQM forudsætter en omfattende estimeringsproces baseret på en lang forundersøgelsesfase, og dels fordi SQM er kvantitativt opbygget, hvilket nærmest er en kontradiktorisk modsætning til grundstrukturen i XP. Hvis XP skulle indbefatte SQM, ville det udelukkende være på den enkelte iteration, og opfølgningen på de enkelte tasks vil være delegeret til udviklerne. Det er sandsynligvis muligt at indføre kvantificerbare størrelser, men det vil passe meget dårligt ind i XP strategien, da fejl som oftest løses på stedet, og hvis fejl først skal registreres og beskrives, ryger der vigtig tid fra selv opgaveløsningen. Det er selvfølgelig en ledelsesmæssig beslutning, hvorvidt man vil indføre kvantificerbare elementer i en XP-udvikling, men det vil uvægerligt flytte en stor del af ansvaret for det udviklede over på lederne i stedet for udviklerne, hvilket er kontradiktorisk i forhold til XP. Derfor må det konkluderes, at SQM ikke er realiserbar med XP, og at det selv på iterationsniveau kun vil kunne indføres ved at gå på kompromis med XP's grundlæggende værdi omkring enkelhed, hvilket må frarådes.

CMM niveau 5

Nu virker det som en kontradiktion at bevæge analysen videre til CMM niveau 5, da inklusionsprincippet i CMM modellen fordrer certificering på nedenstående niveauer, før man kan certificeres på ovenstående, og eftersom en XP organisation teoretisk ikke kan certificeres på niveau 4 kan organisationen heller ikke certificeres på niveau 5. Imidlertid er niveau 5 interessant at analysere, da intensionen med CMM niveau 5 er indenfor XP's rækkevidde uden dermed at muliggøre en reel CMM certificering. Derfor analyseres niveau 5 med det formål at belyse denne intentionelle kobling mellem CMM modellen og XP senere i diskussionen (s. 85). På dette øverste niveau i CMM modellen underbygges de kvantitative elementer fra niveau 4 med yderligere kvantificerbare elementer, som muliggør en kvantitativ evaluering af de procesforbedrende initiativer fra niveau 4. Dermed befinder niveau 5 sig på en form for metaniveau, idet redskaberne på niveau 5 har til formål kontinuerligt at optimere og forbedre initiativer fra alle underliggende niveauer, som analysen af nedenstående KPA illustrerer. Jalote illustrerer dette således:

> *One approach to quantitative quality management is through defect prediction. In this approach, the quality goal is set in the terms of delivered defect density. The other aspect of quality management – managing the development process – requires that the process be controlled (quantitatively) in a manner such that the desired defect density goal is met.*
>
> [Jalote, s. 149]

Defect Prevention (DP)

Grundlæggende indeholder DP ifølge Jalote tre kriterier:

- *Defect prevention activities are planned*
- *Common causes of defect are sought and identified.*
- *Common causes for defect are prioritized and systematically eliminated.*

[Jalote, s. 12]

DP er det evalueringsværktøj på CMM niveau 5, som muliggør en identificering af mulige fejl i den udviklede software, og denne evaluerende instans bygger på viden fra tidligere projekter via en database, som muliggør kvantificering. Denne database indsamler erfaring fra tidligere projekter om, i hvilke faser af udviklingen fejl er identificeret, hvormed det bliver muligt at optimere processer, som kan eliminere fejl i den udviklede software. DP er dermed afhængig af en stabil udviklingsproces på et organisatorisk niveau, da DP afhænger af, at alle projekter følger den samme udviklingsmodel og metode, hvilket hænger sammen med Organization Process Definition (OPD) fra niveau 3 (s. 53).

I en lineær softwareudviklingsproces er verifikationen af koden en indlejret del af udviklingen men er placeret i slutningen af udviklingsprocessen i form af review. XP udnytter de verificerende instanser i både Pair Programming, The Planning Game og i refactoring således, at fejl og mangler identificeres meget tidligt i udviklingen, hvorfor forbedring sandsynligvis ikke bliver en særlig kostelig affære. Derfor er DP et element i XP, som allerede er indeholdt i hvert fald hvad angår identificering af fejl. De kvantitative strukturer er kun lidt bevendt i XP, da udviklingen foregår i iterationer, og hvor sammensætningen af par varierer fra story til story. Det betyder dog ikke, at XP ikke vil have fordel af at indføre kvantificerbare metoder, men som beskrevet tidligere spiller kvantificerbare metoder ikke sammen med XPs grundlæggende værdier. XP ville få meget ud af at identificere og forbedre udviklingsstrukturer, som muliggør fejl i det udviklede system, men denne identificering bør foregå kommunikativt, idet XP er en foranderlig udviklingsmetode baseret på kommunikation, som er i stand til at tilpasse sig omgivelserne og ikke stringent holder fast i specifikke metoder eller strukturer, hvis det viser sig at kunne udføres bedre på en anden måde.

Som videnrepræsenterende værktøj er DP meget specifik, og denne viden er et udtryk for det, den undersøger, altså det tætteste man kommer på verden uden at være verden. DP er som alle andre videnrepræsentationelle værktøjer ikke i stand til at se, hvad den ikke man se, hvilket er forhold, man som organisation ikke kan komme ud over. DP yder dog organisationen en god bistand til at anvende Process Change Management (PCM) senere på dette niveau (s. 65). DP er ligeledes et uvurderligt redskab til at håndtere fejl og problemer med softwaren og skaber selv med de kvantitative metrikker et solidt grundlag for beslutninger omkring fejlrettelser o.l. Hertil må DP vurderes at være uundværlig.

Det må konkluderes, at endnu et KPA på disse øverste niveauer i den grad baseres på en substantiv rationalitet, hvor alt både kan og skal beskrives for at kunne anvendes kvantificerbart. XP er i området omkring DP primært baseret på en kognitiv rationalitet, hvor man handler ud fra de informationer, der er til rådighed, hvilket begrundes med, at udviklere selv identificerer og løser fejl og mangler i det øjeblik de identificeres i den enkelte iteration. Der er ikke i XP behov for kvantificerbare elementer, da de identificerende strukturer omkring fejlfinding er inkorporeret i XPs redskaber. Procesforbedrende aktiviteter er indlejret i XP på et kommunikativt niveau, idet de daglige møder muliggør omstrukturering af udviklingsaktiviteter på daglig basis. Derfor må konklusionen være, at XP indeholder fejlidentificerende redskaber, men at disse informationer kun mundtligt distribueres i organisationen, hvorfor det ikke er muligt kvantitativt at behandle disse informationer, og derfor er DP kun delvist indeholdt i XP.

Technology Change Management (TCM)

Grundlæggende indeholder TCM ifølge Jalote tre kriterier:

- *Incorporation of technology changes is planned.*
- *New technologies are evaluated to determine their effect on quality and productivity.*
- *Appropriate new technologies are transferred into normal practice across the organization.*

[Jalote, s. 12]

Selvom det ikke nævnes eksplicit i kravene til dette KPA, er TCM igen et kvantitativt vurderingsredskab, der gennem en struktureret analyse, afprøvning og vurdering af nye teknologier er i stand til kvantitativt at vurdere nye teknologiers indflydelse på procesforbedring på et organisatorisk niveau [Jalote s. 13]. TCM hænger nært sammen med Qualitative Process Management (QPM) på niveau 4, men TCM fokuserer primært på procesforbedrende teknologier frem for procesforbedring generelt. Denne instans medfører, at organisationen er på forkant med nye teknologiers indflydelse på arbejdsforhold, hvilket kan betyde, at organisationen er i stand til at udvikle nye procesforbedrende aktiviteter, hvormed man er på forkant med udviklingen. Denne instans afkræver dog en særskilt afdeling, som igennem et veludviklet testmiljø er i stand til at vurdere disse nye teknologier, hvilket kan være en særdeles bekostelig affære. Iværksætter man som organisation TCM på et organisatorisk niveau, må det vurderes, at det på sigt vil være indsatsen værd for organisationen som helhed.

Det er problematisk at identificere specifikke videnrepræsentationelle fordele eller ulemper i TCM, da den er en selvstændig del af organisationen, og alt afhængigt af hvordan denne isolerede del af organisationen fungerer, vil TCM kunne bibringe både lidt og meget viden omkring teknologiske vurderinger i forhold til organisationen. Det er meget afhængigt af virksomheden, som implementerer denne funktion. I forbindelse med de kvantitative cost/benefit analyser i forhold til de enkelte teknologivurderinger yder denne selvstændige afdeling dog en særdeles profitabel viden omkring nye teknologier, hvilket må siges at være et positivt træk i enhver virksomhed, som anvender teknologivurderinger. Fungerer denne afdeling optimalt, vil virksomhedens ledelse være i stand til at træffe beslutninger omkring indførelse af ny teknologi på et solidt grundlag, og i modsat fald kan disse beslutninger være fatale for virksomheden.

Igen må det konstateres, at XP på ingen måde er i stand til at opfylde dette KPA. Der er ikke tale om nogen udtalt rationalitet, idet TCM primært er et kvantitativt vurderingsredskab. At TCM er i tæt samarbejde med QPM antyder en substantiv rationalitet, idet TCM er kvantitativt anlagt, men det kan ikke konkluderes endeligt, idet det igen er op til den enkelte organisation at håndtere denne afdeling forsvarligt.

Process Change Management (PCM)

Grundlæggende indeholder PCM ifølge Jalote tre kriterier:

- *Continuous process improvement is planned.*
- *Participation in the organization's software process improvement activities is organization-wide.*

- *The organization's standard software process and the project's defined software process are improved continuously.*

[Jalote, s. 12]

Dette sidste KPA i CMM modellen sigter mod generel procesforbedring, både på organisatorisk niveau og på projektspecifikt niveau. PCM er som både Defect Prevention (DP) og Technology Change Management (TCM) kvantitativt anlagt, idet procesforbedringer skal initieres på et organisatorisk strukturelt niveau [Jalote s. 13]. PCM giver organisationen en tempofordel i forhold til organisationer på lavere CMM niveauer, idet PCM medfører kontinuerlig procesforbedring, hvormed organisationen konstant optimerer metoder og procedurer. PCM baserer denne optimering på de resultater, der er opnået i niveau 4 ved Qualitative Process Management (QPM), men hvor QPM primært baseres på selve dataindsamlingen, sørger PCM for, at disse data anvendes specifikt procesforbedrende i hele organisationen og i samarbejde med et tværsnit af hele organisationen, dvs. personer fra alle afdelinger og på alle niveauer.

Som videnrepræsenterende værktøj yder PCM en høj grad af opmærksomhed omkring den udviklede software, og de fejl, som måtte komme som følge heraf i forhold til de anvendte processer. Denne viden er meget specifik og detaljeret, hvilket må siges at give en konstruktiv viden om den udviklede software. Fokus på de processuelle tiltag, som skal minimere fejl, skaber en øget opmærksomhed omkring metoder, procedurer og redskaber, hvilket tilbringer organisationen en stor viden omkring organisationens softwareudvikling, som ganske sikkert vil give organisationen et stort forspring i forhold til andre softwareorganisationer. Det betyder, at organisationen er i stand til at træffe beslutninger omkring ændringer i selve udviklingen på et meget solidt grundlag, til trods for de kvantitative vurderingsredskaber.

Som videnrepræsenterende værktøj er PCM særdeles interessant, idet viden til procesforbedring trækkes fra alle niveauer og afdelinger i organisationen, og at denne viden akkumuleres kvantitativt medvirker til, at netop PCM er et værktøj, som stort set alle organisationer burde stræbe imod. Evnen til kontinuerligt at optimere processer og metoder virker utroligt tiltalende i XP regi, selvom det må kræve helt utroligt meget arbejde at indsamle viden og gennemføre disse procesforbedringer. Hvis en organisation er i stand til at lære gennem kvantitativt indsamlede data og distribuere denne viden på tværs af organisationen, er der virkelig tale om en lærende organisation, som uundgåeligt vil præge fremtidig softwareudvikling både på et organisatorisk og projektmæssigt niveau.

XP indeholder ganske vist procesforbedrende værktøjer, der på projektniveau og organisatorisk niveau er i stand til at forbedre procedurer og metoder, men XP indeholder ikke en standardiseret metode til at overvåge og vurdere procesforbedrende initiativer. Derfor må det vurderes, at XP ikke indeholder PCM og derfor ikke kan opfylde dette KPA.

Opsummering

Efter denne analyse af de enkelte KPA i CMM modellen, er det nu muligt at konkludere teoretisk. Denne teoretiske konklusion senere vil danne basis for det praktiske eksperiment for at afgøre, om de KPA som ikke er indeholdt i XP ifølge den teoretiske analyse praktisk kan indeholdes i XP. Desuden vil den kommende diskussion problematisere det identificerede intentionelle skift i CMM mellem niveau 3 og niveau 4 og gene-

relt diskutere, hvorvidt de kvantitative mekanismer er vejen frem for en organisation baseret på XP, og som har til hensigt at opfylde de intentionelle kriterier til CMM modellens niveau 5.

Teoretisk konklusion

640K ought to be enough for anybody
- Bill Gates, 1981

Den analytiske del af dette speciale illustrerer foreløbigt, at XP teoretisk indeholder strukturer, som delvist kan effektuere CMM modellens *key process areas* på niveau 2 og 3, såfremt implementeringen heraf er autoritativ. Niveau 4 og 5 må konstateres at være uden for XP rækkevidde, også selvom Defect Prevention (DP) på niveau 5 er delvist opfyldt. CMM modellens inklusionsprincip skal stadig holdes for øje, hvorfor det ikke er muligt for en organisation at certificeres på niveau 5, såfremt kravene til niveau 4 ikke er opfyldt. Den grundlæggende proceduremæssige rationalitet er det kommunikative redskab i XP, som bevirker, at videnrepræsentationen gennem beslutningerne i højere grad tilnærmer sig virkelighedens fænomen, end den substantive og instrumentelle rationalitet, som er karakteristisk for traditionelle udviklingsmetoder, som CMM modellen tager udgangspunkt i. De modsatrettede valg af rationalitet er dog ikke noget kombinatorisk problem mellem XP og CMM modellen, da CMM modellen ikke pålægger udviklingsvirksomhederne specifikke redskaber eller metoder til at opfylde de forskellige KPA, men dog pålægger organisationen kvantificerbare styringsmekanismer. CMM modellen opstiller kriterier, som skal opfyldes, før organisationen kan tilskrives at eksistere på et bestemt niveau, men hvordan disse kriterier opfyldes er op til den enkelte organisation.

En gennemgående tanke i CMM modellen er dog en styringsmæssig ledelsesstil frem for en ledelsesmæssig ledelsesstil, og denne styringsmæssige tanke fremkalder de kvantificerbare mekanismer, som CMM modellen både eksplicit og immanent lægger op til. I det styringsmæssige perspektiv er XP's værdigrundlag særdeles divergerende, da hele grundideen i XP fokuserer på delegering og tillid, frem for kontrol og styring. Delegering og tillid er i XP netop de værdier, som igennem kommunikation muliggør organisatoriske forandringer og hurtig omstilling, hvilket CMM modellen intentionelt kræver på de tre øverste niveauer. Selvfølgelig kan alle organisationer teoretisk opstille og anvende redskaber, som netop muliggør relativt hurtige organisatoriske forandringer, men vejen hertil kan være lang i forhold til strukturelt at forandre en organisation baseret på XP. De kvantitative mekanismer, som bedst kommer til udtryk på CMM niveau 4 og 5, er problematiske at indeholde i en organisation baseret på XP, med mindre organisationen er villig til at gå på kompromis med XPs grundlæggende værdisæt. Om det er hensigtsmæssigt eller ej skal forblive usagt i specialet, da det er op til den enkelte organisation at afgøre dette, men rent teoretisk er det et problematisk kompromis at indgå; endsige umuligt.

Der kan rent teoretisk være god fornuft i at basere en organisation med XP som grundlæggende udviklingsværktøj, idet organisationen har god mulighed for at tilnærme sig på niveau 3, som denne teoretiske analyse illustrerer, dog uden at kunne opnå en reel certificering, som kræver, at alle KPA er opfyldt. Har man som etableret organisation et ledelsesmæssigt ønske om at tilnærme sig CMM modellens niveau 3, og dermed kvalificere organisationen yderligere i eksempelvis udbudsrunder, kan XP's grundlæggende værdier supportere dette ønske, dog med undtagelse af Software Subcontract Management (SSM) og Integrated Software Management (ISM), idet hverken redskaber eller iboende værdier i XP supporterer udlicitering, med mindre udlicitering kan etableres som en del af den interne organisation. Men kan man nøjes med at skræddersy SSM og

ISM i den enkelte organisation, er det en meget billig pris at betale i forhold til at skræddersy alle 13 KPA i den enkelte organisation. Det burde være en overkommelig opgave for en organisation baseret på XP. Det ville også være nærliggende at forholde XP til en organisation, hvor SSM og ISM ikke er repræsenteret, men så kan organisationen ikke certificeres på et CMM niveau, hvorfor hele ideen med koblingen mellem XP og CMM umiddelbart bliver uinteressant. Dette ændres dog, hvis man medregner CMM modellens dualisme, hvor CMM modellen også kan anvendes til at guide organisationen til bedre softwareudvikling, uden at indeholde selve certificeringen.

De store problemer viser sig primært i CMM modellens niveau 4 og 5 i forhold til en organisation baseret på XP. Her er de kvantitative mekanismer i de 5 KPA så omfangsrige, at de går på tværs af organisationen, hvilket i sig selv ikke er dårligt. Det dårlige ved denne tværorganisatoriske tilgang er de kvantitative procedurer, som kvantitativt skal kvalificere organisationen til at optimere og revurdere softwareprocesser, metoder og tekniske redskaber. Denne kvantitative tilgang går ind og rammer XP's grundlæggende værdisæt lige i hjertet, hvor ansvaret fratages den enkelte medarbejder og lægges over til ledelsen, og hvor tillid dermed kun ydes, hvis de kvantitativt opstillede kriterier opfyldes af den enkelte medarbejder. I denne sammenhæng er det svært at tro, at en organisation baseret på XP er villig til at gå så meget på kompromis for at etablere sig på CMM niveau 4 eller 5. I det praktiske eksperiment (side 85) vil jeg dog forsøge at begrunde, hvorfor intensionen med niveau 5 allerede er indeholdt i XP's grundstruktur og værdier, men allerede nu står det klart, at en XP organisation ikke kan certificeres på niveau 4 og dermed ikke på niveau 5. I forbindelse med det praktiske eksperiment vil det dog forsøges at indeholde Integrated Software Management og Software Subcontract Management som en del af XP, hvorfor organisationen kan certificeres på CMM niveau 3. Den teoretiske analyse illustrerer, at det ikke er realistisk at forsøge at certificere en XP baseret organisation på højere niveauer grundet de kvantitative mekanismer, men den senere diskussion vil forsøge at indeholde intensionen med CMM niveau 5 i den XP baserede organisation.

Den teoretiske analyse viser, at XP delvist kan kombineres med kravene i CMM modellens niveau 3, selvom det immanente valg af rationalitet er divergerende for XP og CMM. Hvor CMM modellen grundlæggende tager udgangspunkt i en substantiv rationalitet, har XP hovedsageligt et udgangspunkt i den proceduremæssige rationalitet, men dette forhindrer på ingen måde, at XP kan understøtte en stor del af kravene fra CMM modellens niveau 3, som tabel 1 viser. XP's kommunikative tilgang til udvikling medfører sandsynligvis denne proceduremæssige rationalitet, men også verdenssynet spiller ind, da skellet mellem ledelse og medarbejdere tilsyneladende ikke er så stort som det skel CMM modellen tager udgangspunkt i. Den proceduremæssige rationalitet bevirker også et langt mere virkelighedstro billede af verdenen end det billede den substantive rationalitet tillader, hvilket klart er at foretrække som udviklingsvirksomhed.

Niveau 2 KPA	Opfyldelse	Niveau 3 KPA	Opfyldelse
Requirements Management	√√	Organization Process Focus	√√
Software Project Planning	√√	Organization Process Definition	√
Software Project Tracking and Oversight	√√	Training Program	√
Software Subcontract Management	--	Integrated Software Management	--
Software Quality Assurance	√√	Software Product Engineering	√√
Software Configuration Management	√√	Intergroup Coordination	√√
		Peer Reviws	√√
Niveau 4 KPA	**Opfyldelse**	**Niveau 5 KPA**	**Opfyldelse**
Qualitative Process Management	--	Defect Prevention	√
Software Quality Management	--	Technology Change Management	--
		Process Change Management	--

Tabel 1 – Oversigt over KPA iht. XP
√ : Delvist indeholdt i XP
√√ : Dækket af XP
-- : Ikke dækket af XP

Måden hvorpå XP understøtter CMM modellens krav divergerer med den underforståede anvendelse i CMM modellen, men da CMM modellen ikke foreskriver valg af værktøjer, er der ikke et metodisk problem i kombinationen, hvorfor det nu vil være muligt at anvende denne teoretiske konklusion operationelt i det praktiske eksperiment for at se, hvordan de ikke dækkede KPA kan indføres og anvendes i praksis ved at indføre nye redskaber og metoder til XP metodologien.

Praktisk eksperiment

At vove er at miste fodfæstet for en stund. At vove intet er at miste livet for bestandig.
- Søren Kierkegaard

Dette kapitel vil beskrive det praktiske eksperiment, som vil udvide den teoretiske konklusion og vurdere, hvorvidt XP kan indføre Software Subcontract Managment (SSM), Integrated Software Management (ISM) og Training Program (TP) som en del af metodologien og samtidig bibeholde XP's værdier, så organisationen reelt kan certificeres på CMM niveau 3. Målet i forbindelse med eksperimentet var kort fortalt at afprøve XP i en virksomhed på et organisatorisk niveau. Det optimale eksperiment ville være dels at få en praktisk forståelse af XP som organisatorisk udviklingsværktøj, dels at etablere en måling som viser om XP konstituerer en organisation på et bestemt CMM niveau. Men da dette ikke har været mulig grundet manglende udviklingskapital, vil dette praktiske eksperiment kun beskæftige sig med den organisatoriske strukturering og indledende forundersøgelser frem for reel udviklingsproces med XP som grundlæggende udviklingsredskab. Dette praktiske eksperiment vil ikke indeholde den forundersøgelse, som projektgruppen udarbejdede, idet den anses for værende af fortrolig forretningsmæssig viden. Jeg har prioriteret, at det er mere givende at udarbejde et speciale som er tilgængeligt for alle, hvormed den iboende viden kan distribueres, frem for et fortroligt speciale, som kun kan læses af vejleder og censor.

Ideen bag organisationen

Hele ideen bag organisationen blev til i forbindelse med praktikken på 8. semester (forår 2003), som foregik i Sonofons udviklingsafdeling. Her skabtes ideen om at kombinere Extreme Programming med organisatoriske teorier, hvilket ledte frem til indeværende speciale. Det stod dog klart, at koblingen burde verificeres operationelt, men dette ville betyde, at den implicerede organisation blev nødsaget til at ændre eksisterende politikker og udviklingsmetoder, for at opfylde et teoretisk mål, der måske ikke på sigt ville være opnåeligt. Derfor stod det hurtigt klart, at indeværende speciale skulle eksperimentere på en organisation, hvor der var fuld kontrol over metoder og redskaber, for på den måde at vurdere, hvordan de operationelle problemstillinger ville svare til de teoretiske ideer. Den 5. januar 2005 blev dette en realitet, da jeg egenhændigt opstartede virksomheden ***m*2 i 3D** (kvadratmeter i tre dimensioner) sideløbende med udarbejdelsen af dette speciale.

Virksomheden er dog ikke alene til for at understøtte ideen med specialet. Målet er at udvikle en tjeneste til ejendomsmæglere, hvor den enkelte mægler kan repræsentere boliger i tre dimensioner direkte på Internettet, og hvor brugere (de mulige købere) vil have mulighed for at modificere boligen ved eksempelvis at tilføje eller fjerne vægge eller vinduer, ændre i farvekombinationerne eller på anden måde ændre ejendommen, så latente muligheder bliver synlige for brugeren. Dermed ligger der også en kulturændring for salg af ejendomme gennem Internettet, hvor brugerne bliver proaktive i valget af ejendomme modsat nuværende tjenester, hvor brugerne er reaktive. Med opstarten af egen virksomhed får jeg desuden mulighed for at afprøve de kvalifikationer, som uddannelsen har rustet mig med, hvilket formodentligt på sigt vil give mig en fordel på arbejdsmarkedet.

Indledende manøvrer

Nu starter man ikke bare en softwareudviklingsvirksomhed på en enkelt dag, hvis man som i dette tilfælde er studerende og derfor uden de store kapitalreserver. Før der overhovedet kan ansættes udviklere, er det nødvendigt at udforme en forretningsplan, som giver et strukturelt overblik over organisationen og økonomien bag, og som samtidig kan tiltrække kapital fra investorer. Dermed hænger specialet og virksomheden sammen, eftersom virksomheden er den operationelle del af specialet og specialet er det bagvedliggende teoretiske fundament, som gør virksomheden mulig. Der er dog ikke tale om et dialektisk forhold mellem speciale og virksomhed, da begge dele udmærket kan stå alene (hvilket også er ønskeligt), men hvor koblingen har en særdeles positiv indvirkning på både virksomhed og speciale.

I januar 2004 fik jeg på baggrund af min forretningsplan et scholarship på Dreamhouse i Aalborg, som er et kommunalt iværksætterhus, hvor enkeltmandsvirksomheder mod betaling kan få stillet en arbejdsplads til rådighed, og hvor man samtidig får et netværk af personer med vidt forskellig forretningsmæssig baggrund. Dette netværk er af helt utrolig betydning i opstartsfasen, hvor erfaringer og gode råd medvirker til, at man ikke selv skal opfinde den dybe tallerken hver dag. Man trækker på hinanden og stiller gerne sin viden og erfaring til rådighed for de andre virksomheder, hvormed der opstår et helt unikt kommunikativt rum, som mange virksomheder kan lære noget af. Ud over det tildelte scholarship, fik jeg i februar tilknyttet en projektgruppe fra 6. semester humanistisk datalogi fra Aalborg universitet, som primært skulle lave en forundersøgelse, som ***m*2i3D** kunne bruge i den videre udvikling. Dette samarbejde blev dog udvidet efter forundersøgelsen til også at indbefatte virksomhedens interne strukturering, primært møntet på videndeling i forhold til XP. Universitetsgruppen ville udvikle et redskab, som skal anvendes til at dele viden internt i organisationen, især fordi ***m*2i3D** i en del af udviklingen vil anvende studentermedhjælpere. Universitetsgruppen deltog udover forundersøgelsen i en to-dages workshop, hvor vi anvendte redskaberne i The Planning Game (se side 36). Ydermere har universitetsgruppen medvirket i en længere samtale, hvor vi diskuterede deres oplevelser af XP som udviklingsmetode i forbindelse med The Planning Game, og hvor vi ligeledes diskuterede de mere problematiske sider af XP i forhold til de kvantitative målinger som CMM modellen kræver. Transskriptionen af denne samtale forefindes i Bilag 4, s.119.

Praktiske iagttagelser og paradokser

Det praktiske eksperiments meget personlige ophav er i sig selv en problematisk størrelse, og begrundelsen for at anvende teorien kan egentlig udtrykkes ganske kort: det kan simpelthen ikke gøres anderledes. Det er selvfølgelig problematisk selv at være en del af selve empirien, men det er sandelig også et problem at iagttage udefra, og set i lyset af en snarlig introduktion på arbejdsmarkedet er det at foretrække at få så meget praktisk erfaring som muligt igennem dette specialeforløb. Men da denne begrundelse ikke just er en videnskabsteoretisk begrundelse, er det passende at ofre tiden på at beskrive de iagttagelser og paradokser, som uundværligt vil konstateres gennem eksperimentet, og som derfor også er en væsentlig del af problemstillingen.

Verificeringen

Den første og måske vigtigste iagttagelse af den opmærksomme læser er selve verificeringen af det praktiske eksperiment. Hvordan kan de empiriske dele som modeller og procedurer verificeres, så de ikke fremstår som opdigtede glorificerende elementer, som udelukkende understøtter den teoretiske konklusion og ikke har nogen operationel berettigelse? Her skal man så tage med i betragtningen, at eksperiment og teori i dette speciale hænger sammen, så elementer i teorien påvirker eksperimentet, og ligeledes påvirker eksperimentet de teoretiske antagelser. Der er altså ikke tale om, at eksperimentet skal stå alene; det skal ses i konteksten af teoriens antagelser. Målet med specialet er jo netop at skabe en handlingsramme, hvori det kan vurderes, om redskaberne i XP kan supportere de organisatoriske processer, som er medvirkende til at konstituere en opstartsvirksomhed på et givent CMM niveau. Dermed kommer man helt uden om verificeringsgraden af eksperimentet, da konklusionen jo ikke er et enten/eller, men handler om i hvilken grad dette er muligt. Verificeringen af det praktiske eksperiment ligger med andre ord i sondringen mellem de teoretiske antagelser og den organisatoriske berettigelse, hvormed eksperimentet som enkeltstående verificeret enhed bliver sagen uvedkommende.

Optimalt	Realitet
En praktisk vurdering af XP som organisatorisk redskab	En teoretisk vurdering af XP som organisatorisk værktøj
En måling som viser om XP kan konstituere en organisation på et givent CMM niveau	En praktisk anvendelse til at opstarte en organisation vha. XP.
	Anvendelse The Planning Game.
	Samtale med universitetsgruppe omkring anvendesen af XP strukturer

Tabel 2 – overblik over empiri

Wittgensteins dilemma

I og med at jeg selv er en del af empirien, er det ikke muligt at konstruere en organisation, hvor det er muligt at betragte redskaber og procedurer objektivt i et fænomenologisk perspektiv. Det kunne selvfølgelig være ønskeligt at besvare problemstillingen uden egne præferencer og dermed konstruere en iagttagelseshorisont, som muliggør denne distinktion. Wittgenstein udtrykker det således:

> *En edderkop kravler rundt i sit spind, eftersom den har sat sig det mål at overskue sig selv og hele sit net. Men den er optaget af en temmelig futil opgave, idet det aldrig vil kunne lade sig gøre for edderkoppen at fuldføre opgaven: Hvis den skulle overskue hele nettet, ville den blive nødt til at fjerne sig fra det, fordi den kun på nettet kan overskue enkelte områder ad gangen. Og hvis den løftede sig ud af nettet, vil den netop fjerne sig fra det og dermed stadig ikke overskue sig selv sammen med nettet.*
>
> [Wittgenstein citeret efter Føllesdal m.fl., 1997, s.99]

Hvis man sætter ovenstående eksempel i relation til førnævnte begreber, kan sammenhængen forklares således: Som edderkoppen i sit spind, ser jeg min egen deltagelse i eksperimentet som at kravle rundt og afdække enkelte partielle områder af sandheden og tilsammen tildele dem mening, på baggrund af blot en formodning om en verden; en verden bestående af interesser. Men jeg kan aldrig forstå hele sandheden, eller afdække alle de interrelaterede fakta jeg møder, idet det vil betyde, at jeg da kender til alle forhold i verden. Og at kende til alle forhold i verden, vil sige at miste sin væren i verden; at hænge udenfor nettet. [Inspireret af Dreyfus & Dreyfus, 1986, s. 77]

Derfor er det hverken ønskeligt eller tilsigtet at hænge uden for nettet og observere tingenes reelle tilstand uden selv at være en del af denne verden. Det spændende i dette speciale er jo netop muligheden for at ændre og tilpasse de enkelte redskaber, så de på den måde får merværdi for både organisationen og specialet. Derfor er det praktiske eksperiment den størrelse, som gør, at jeg kontinuerligt gennem specialet er i stand til at forholde teori og eksperiment med et operationelt perspektiv og dermed finde ud af, hvilke organisatoriske modifikationer der skal til, for at skabe den optimale tilpasning af de operationelle redskaber i Extreme Programming. Derfor vil næste afsnit beskrive selve eksperimentet, hvorefter det er muligt at foretage en praktisk analyse af SSM, TP og ISM i forhold til organisationen.

Forundersøgelsesfasen

Det første samarbejde, som universitetsgruppen og jeg etablerede, var en decideret forundersøgelsesfase, hvor nuværende arbejdsgange og rutiner hos Home a/s skulle belyses for at kunne udvikle et system, som også kunne afhjælpe problematiske eller redundante beslutningsprocedurer eller arbejdsgange. Ideen var, at det udviklede system direkte skulle tilbyde væsentlige forbedringer i dagligdagen samtidig med at tilbyde et nyt produkt. På vores første møde præsenterede jeg mine egne forundersøgelser, og vi kom frem til, hvad målet for forundersøgelsen fra min side skulle være. Vi opstillede en række succeskriterier, hvorefter ansvaret blev lagt ud til universitetsgruppen, som havde råderum til at håndtere opgaven, som de fandt hensigtsmæssigt.

Forundersøgelsen blev dermed til en decideret udlicitering af en opgave, som ganske vist ikke var indenfor XP's råderum, men som adresserede Software Subcontract Management (SSM), som var et af de identificerede problematiske områder i CMM modellen i forhold til XP. Det gode resultat af forundersøgelsen viste med al tydelighed, at udlicitering er mulig, selvom man internt anvender XP som udviklingsmetode[11]. Personligt er jeg af den overbevisning, at så længe man skaber incitament til kvalitet i den enkelte opgave, eksempelvis igennem distribuering af ansvar og etablering af tillid, kan man sagtens udlicitere opgaver, så længe man er bevidst om resultatet af den udliciterede opgave, og man løbende holder kontakt for at sikre konsistens i forhold til resten af organisationen. Eksempelvis anvendte jeg i forundersøgelsesfasen ingen målemekanismer, men forholdte mig gennem udliciteringsfasen løbende kritisk til de forskellige løsninger og påpegede nye indgangsvinkler til genstandsfeltet. Jeg var så at sige den verificerende kommunikative instans, som sikrede konsistens mellem forundersøgelsen og den forretningsmæssige idé.

[11] Denne forundersøgelse er som nævnt tidligere fortrolig, og vil derfor ikke være at finde som bilag.

Efter denne forundersøgelse, indvilgede universitetsgruppen til min store glæde i at fortsætte samarbejdet, hvor omdrejningspunktet blev anvendelsen af studentermedhjælpere i en virksomhed, men hvor deres tilknytning til ***m*2i3D** kunne belyse problematikker omkring studentermedhjælpere, som så senere kunne anvendes til den reelle ansættelse af studentermedhjælpere. Der blev i dette videre samarbejde fokuseret mere på organisatoriske problematikker end på det rent udviklingsmæssige aspekt, hvilket er særdeles interessant i henhold til specialets fokus. Samarbejdet startede med en workshop.

Workshop omkring videndeling

Mandag den 19. april mødtes den tilknyttede universitetsgruppe med undertegnede i Dreamhouse's mødelokale Sokrates. Målet var at planlægge og strukturere et videnindsamlingssystem, som både kan skaleres og som samtidig skal tage højde for den overvejende anvendelse af studentermedhjælpere, som er planlagt i den interne anvendelse af ressourcer. Dette afsnit vil kort beskrive de anvendte mekanismer i denne planlægningsproces, og samtidig inddrage de XP værktøjer, som hurtigt blev en del af arbejdsgangen.

Indgangen til denne del af eksperimentet er dels et direkte ønske fra projektgruppen om at deltage i en XP designproces og dels et ønske fra min side om at afprøve XP strukturer på et organisatorisk niveau. På et møde aftalte vi at anvende The Planning Game på et reelt problem, nemlig at håndtere og repræsentere viden i en organisation, som i stor stil anvender studentermedhjælpere. Denne problemstilling er ganske reel, da **m2i3D** anlægger store dele af udviklingen på anvendelse af studentermedhjælpere. Anvendelse af studentermedhjælpere er en strategisk beslutning ud fra en cost/benefit analyse, da der ganske enkelt kommer flere udviklingstimer fra studentermedhjælpere end fra fastansatte udviklere. Derfor er det på et organisatorisk niveau et behov for at analysere denne udviklingsproblematik og på et strategisk niveau at håndtere studentermedhjælpere i en udstrækning, som reelt medfører en bedre kvalitet i udviklingen af tasks i **m2i3D**. Nedenstående figur illustrerer, hvordan de enkelte storys adresserer den overordnede forståelse af viden, som workshoppen startede ud med at diskutere og formulere, både på et overordnet niveau og på et organisatorisk niveau. Beskrivelsen af de enkelte storys og de dertilhørende tasks er illustreret i bilag 3, side 109. Tabel 3 er fremkommet i afslutningen af workshoppen og formålet var ikke, at alle storys skulle inkludere alt, men udelukkende vurdere de enkelte storys interrelaterede områder, for på den måde at vurdere, hvilke storys, der adresserer hvilken type viden og til hvem.

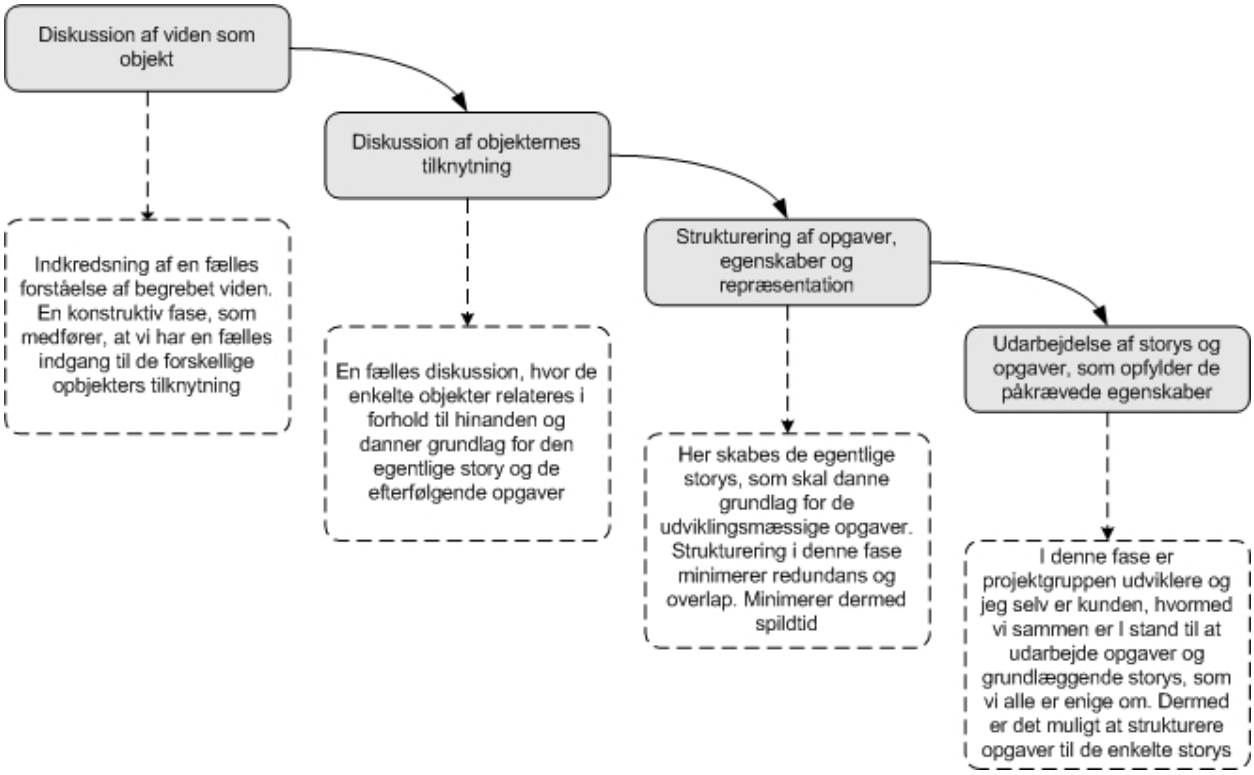
	Story 1 Firmaprofil	**Story 2** Projektniveau	**Story 3** Personlig profil	**Story 4** Administrativt modul	**Story 6** Idégrundlag	**Story 7** Erfarings-opsamling
Overordnet						
Hvilken viden	√	√		√	√	√
Til hvem	√	√	√	√	√	√
Hvornår			√			√
Hvordan			√			
Hvorfor			√	√		
Organisatorisk						
Personer	√		√	√	√	√
Strukturelt	√		√			√
Processuelt	√		√	√		√
Fordeling af ansvar	√		√	√	√	√
Arbejdsmiljø			√			
Kompetence	√		√			
Formel/uformel kommunikation			√	√	√	√

Tabel 3 – relationer mellem storys

Tabel 3 illustrerer hvilken viden der adresseres i den enkelte story, hvem denne viden henvender sig til og hvilket organisatorisk fokus, der ligger i den enkelte story. Tabellens formål er at skabe overblik, så alle områder adresseres i den samlede præsentation af et videnindsamlingssystem, hvormed det samlede system opfylder de kriterier, workshoppen startede med at identificere.

Designprocessen

Projektgruppen og jeg planlagde jeg en fælles workshop, hvor vi skulle etablere en fælles forståelse af viden som objekt, dels som brug til empiri til deres 6. semesters projekt, dels til (for mit vedkommende) at vurdere hvordan XP strukturer og redskaber fungerer uden selve programmeringsdelen, og hvordan viden deles under The Planning Game. Designprocessen blev opdelt i fire elementer, som illustreret i figur 8.



Figur 8 – strukturering af workshop

I de første tre faser diskuteres der i plenum, og selvom starten var temmelig træg og diskussionen og problematiseringen af diverse elementer udeblev, skete der pludseligt et skred i gruppens opfattelse af mig som ligemand og ikke som en ældre studerende eller virksomhedsleder, som de måske opfattede mig som. I hvert fald åbnede diskussionen op i starten af 2. fase, hvor vi diskuterede de enkelte objekters tilknytning. Her bredte diskussionen sig ud og alle kom med i diskussionen på lige præmisser. Da vi efter et stykke tid løb tør for ideer, begyndte designet af de enkelte storys. Nogle storys var mere omfattende end andre, men generelt var indtrykket, at der var incitament til 7 storys. Herefter gik beskrivelsen af de enkelte storys i gang, og der blev designet en række tasks, som relaterede til den pågældende story. Her blev det dog hurtigt klart at to storys kunne slås sammen til en enkelt, hvilket gjorde processen lidt lettere[12]. Der blev altså i XP terminologi anvendt refactoring gennem udviklingen af stories, hvilket hurtigt mindskede systemets kompleksitet.

Der blev hurtigt dannet to par, og jeg selv fungerede lidt i begge grupper, for at sikre en vis form for konsistens og mindske redundans i de enkelte storys. Min rolle mindede meget om den kommunikationsdedikerede medarbejder, som sørger for at kommunikationen forløber hensigtsmæssig. Derfor havde jeg lige så meget fokus på de kommunikative aspekter af denne workshop, som selve løsningen af de enkelte storys. Denne rolle bevirkede, at jeg kunne modificere de kommunikative strukturer og anvendelsen af redskaber i XP (primært Pair Programming) til at virke efter hensigten og hjælpe kommunikation og verifikationen i stedet for at obstruere den, hvilket der især i opstartsfasen var brug for.

[12] Alle Stories og Tasks kan forefindes i bilag 3, side 109.

Udgangspunktet for workshoppen var dels beskrivelsen og modellen af The Planning Game, dels de organisatoriske overvejelser som eksempelvis viden og læring, som er en del af CMM modellens kriterier. I den videre analyse vil jeg diskutere eksperimentet i forhold til den teoretiske konklusion og derudfra vurdere, hvorvidt XP kan supportere CMM modellens kriterier.

Samtale omkring XP

Efter workshoppen fandt jeg det nødvendigt at indhente de erfaringer, som universitetsgruppen havde gjort sig i forbindelse med workshoppen og forundersøgelsen. Dels for at indhente forbedringsforslag, dels for at problematisere områder fra specialets teoretiske analyse. Planen var at diskutere universitetsgruppens oplevelser indenfor: deling af viden, delegering af ansvar, udlicitering af opgaver i forundersøgelsen, sikring af kvalitet, processtyring og procesændringer, videreudvikling af egne evner ved anvendelsen af Pair Programming, koordinering i grupperne, en mulig indførelse af kvantitative måleværktøjer (iht. CMM niveau 4 & 5) og muligheden for at strukturere en organisation udelukkende baseret på værdier og redskaber i XP. Den fulde transskription af denne samtale forefindes i bilag 4.

Analyse af XP som organisatorisk virkemiddel

Udgangspunktet for opstarten af virksomheden var selvfølgelig, at ***m*2i3D** skulle være i stand til at anvende XP som udviklingsmetode og samtidig indeholde XP's grundlæggende værdisæt på et organisatorisk niveau. Det betød, at ***m*2i3D** i sin opbygning skulle være enkel, bygge på kommunikative redskaber og tillade feedback på alle niveauer og i alle områder. Det største problem ved kommunikationen var imidlertid, at der kun var en enkelt person tilknyttet virksomheden, nemlig undertegnede. Her udnyttede jeg det netværk i det kommunale iværksætterhus Dreamhouse, som knytter erfaringer og viden fra de 25 andre virksomheder, til at opbygge de strukturer, som nu engang skal eksistere (eksempelvis marketing, jura og økonomi), hvormed jeg var i stand til at anvende reelle kommunikative redskaber til at træffe de organisatoriske beslutninger. Enkelheden opstod som en naturlig del af denne kommunikation, da jeg som udgangspunkt skulle være i stand til at ændre beslutninger, hvis et andet alternativ dukkede op, som viste sig at være bedre end det først antagede. Nedenstående afsnit vil derfor kort beskrive anvendelsen af kommunikation i forbindelse med udarbejdelsen af forretningsplanen, som er en essentiel del af opstarten af en virksomhed. Dernæst vil samarbejdet med universitetsgruppen vurderes i forhold til SSM, ISM og TP.

Forretningsplanen

I forbindelse med kurset "mod egen virksomhed" som en del af mit 9. semester på humanistisk datalogi, udarbejdede jeg en mini forretningsplan, hvor jeg fik en føling med de forventninger, der er til opbygning, produktpræsentation og økonomi, som forventes af en færdig forretningsplan. Denne mini forretningsplan blev udgangspunktet for ***m*2i3D**, hvor jeg i samarbejde med mulige investorer og virksomhederne i Dreamhouse videreudviklede denne mini forretningsplan til en færdig forretningsplan, som kunne præsenteres for mulige investorer og samarbejdspartnere. I forbindelsen med udarbejdelsen af den færdige forretningsplan, anvendte jeg XP som udviklingsredskab, idet jeg inddelte forretningsplanen i storys, hvor jeg opstillede testscenarier og derudfra udviklede tasks, som kunne opfylde disse testscenarier. Dermed blev forretningsplanen

opdelt i 27 stories med dertilhørende tasks. Udarbejdelsen af forretningsplanen blev dermed til et reelt udviklingsprojekt, hvor XP blev anvendt i vid udstrækning, og hvor kernen i udviklingen var mini forretningsplanen, som blev udviklet og godkendt af underviserne og de tilknyttede forretningsfolk på 9.semester.

De enkelte tasks blev dog ikke udført af to personer, som XP kræver i Pair Programming, men kun af mig. De verificerende strukturer blev dog i høj grad bibeholdt, idet jeg udnyttede et kontinuerligt samspil med virksomhederne i Dreamhouse til at belyse områder, jeg ikke havde erfaring indenfor, og herefter til at verificere den valgte tilgang. Dette samspil medførte ganske hurtigt, at jeg fik en stor forretningsmæssig viden, hvorefter jeg var i stand til at præsentere forretningsplanen for både samarbejdspartnere og mulige investorer.

Denne relativt korte anvendelse af XP som udviklingsmetode på andet end software gav en vis fortrolighed med XP, hvorfor jeg kunne fokusere og problematisere de dele af XP, som ikke umiddelbart var indeholdt i CMM modellens nederste niveauer; Software Subcontract Management(SSM), Integrated Software Management (ISM) og til dels Training Program (TP), hvilket nedenstående afsnit vil diskutere ud fra samarbejdet med universitetsgruppen.

Software Subcontract Management

Den første udfordring efter færdiggørelsen af forretningsplanen var at overbevise Home a/s i Aalborg om, at et samarbejde med ***m*2i3D** ville være en stor forretningsmæssig fordel. Overbevisningen skulle fremstå professionelt og i overensstemmelse med det ***m*2i3D** rent faktisk kunne levere, såfremt produktet blev udviklet. I denne forundersøgelsesfase skulle der udarbejdes både økonomiske estimater, visuelle præsentationer og en analyse af nuværende arbejdsgange for at være i stand til at fokusere på procesforbedrende arbejdsgange med højere grad af effektivitet og kvalitet. Både økonomiske estimater og visuelle præsentationer var jeg selv i stand til at producere som en del af de daglige arbejdsopgaver i ***m*2i3D**. Problemet var at afse tid til at lave en gennemgående analyse af arbejdsgangene, hvorfor kontakten blev etableret med universitetsgruppen, som indvilgede i at samarbejde omkring denne forundersøgelsesfase, hvor jeg delegere beslutningskompetence og ansvar til universitetsgruppen, der dog stadig havde mulighed for at få deres tilgange verificeret med mig, før de gik videre. Denne delegering af beslutningskompetence og ansvar er i den grad udlicitering af opgaver, hvilket er kernen i SSM; hvordan håndterer man udliciterede opgaver?

Med min egen manglende erfaring med udlicitering, handlede jeg ud fra XP's værdisæt, hvor tillid og delegering af ansvar er dominerende, hvor kommunikation er midlet og enkelheden er målet. Jeg var (og er) af den overbevisning, at hvis en universitetsgruppe byder ind på en sådan opgave, må de nødvendigvis besidde kompetence til at udfylde denne opgave, hvilket jeg ikke betvivlede undervejs. Jeg skabte kontakten med Home a/s og præsenterede universitetsgruppen for mine egne forundersøgelser, resultater og kontaktpersoner i Home a/s, hvorefter de fik råderum til at designe og planlægge deres egen forundersøgelse, som de baserede på MUST-metoden. Jeg har selv en del erfaring med MUST-metoden gennem diverse semestre, så jeg havde en viden om, hvad metoden var i stand til, og hvilke mangler metoden havde. Dermed var jeg i stand til at supplere og guide universitetsgruppen i planlægningsfasen, så udliciteringen af forundersøgelsen i høj grad blev præget af mine egne krav i forhold til de elementer, som MUST-metoden kan belyse.

I forhold til XP var problemet, at denne forundersøgelsesfase ikke fra universitetsgruppens side anvendte XP som udviklingsværktøj, hvorfor vi i samarbejdet var nødsaget til at anvende en ret traditionel kommunikativ tilgang til udviklingen af forundersøgelsen. Med hensyn til forundersøgelsen og udlicitering generelt virkede dette som en fornuftig tilgang, situationen taget i betragtning, men det mindsker selvfølgelig vurderingen af XP som værktøj mht. SSM. I forundersøgelsesfasen må det konkluderes, at det ikke er problematisk at anvende en anden udviklingsmetode end den gængse, da der ikke ligger decideret softwareudvikling i denne fase af udviklingsprocessen. Det kan dog med empirien taget i betragtning ikke vurderes, hvorvidt XP kan understøtte udlicitering generelt. Her må igen henvises til den i den teoretiske analyse opstillede tese om at konstruere en intern udlicitering, hvor den eksternt tilknyttede organisation sætter udviklere ned i den organisation, hvor kontrakten er indgået. For eksperimentets vedkommende har denne anvendelse af traditionelle kommunikative værktøjer været særdeles givende i forundersøgelsesfasen i forhold til det resultat, som universitetsgruppen præsterede, hvilket grundlæggende betyder, at udlicitering ikke i nævneværdig grad påvirker organisationens anvendelse af XP, og at SSM derfor relativt uproblematisk kan indlejres operationelt i den XP baserede organisations softwaremetode.

Integrated Software Management

ISM er som beskrevet tidligere udsprunget af organisationens standard softwareudviklingsproces, og er i empirisk sammenhæng en lidt speciel størrelse, da ***m*2i3D** som sådan ikke har nogen gennemprøvet udviklingsmodel til softwareudviklingsprojekter, som har rod i den organisatoriske struktur. Intensionen er ganske vist at inkorporere XP som generel udviklingsmetode og samtidig strukturere organisationen på baggrund af XP's grundlæggende værdisæt, men dette har ikke været muligt at etablere på den korte tid, som ***m*2i3D** har eksisteret. Derfor tages der udgangspunkt i intensionen i resten af afsnittet.

I forbindelse med workshoppen og afprøvningen af The Planning Game, ser jeg ingen problemer i at skræddersy en organisatorisk funderet udviklingsproces med rod i XP, hvor det enkelte projekt kan modellere udviklingsmodellen til at understøtte de projektmæssige afhængigheder inklusive udlicitering, som beskrevet ovenfor. Universitetsgruppen oplevede en høj grad af ansvar i forbindelse med gennemførelsen af The Planning Game og ifølge dem selv for meget ansvar (se bilag 4). Men det er måske netop problemet med XP som organisatorisk softwareudviklingsgrundlag. Både tillid og ansvar delegeres i vid udstrækning, hvilket fordrer en helt speciel sammensætning af personer, der har rygrad og mod nok til at sige både til og fra, hvor ansvaret bliver for meget eller for lidt. Denne problematik havde jeg ikke forudset i struktureringen af The Planning Game, og det rejser nogle strukturelle problematikker i forhold til at udvikle en standardiseret organisatorisk funderet softwareudviklingsmetode.

Skal man anvende XP som standard softwareudviklingsmetode, skal der som i alle andre softwareudviklingsmetoder indeholdes modificerbare elementer, som kan tilpasses det enkelte projekt, men samtidig skaber konsistens imellem de enkelte projekter i organisationen. Her er det oplagt at konstruere en dokumenteret softwareudviklingsmetode baseret på XP, hvor de enkelte elementer både indeholder delegering af ansvar og kontinuerlig feedback fra udviklere og ledelse, så modellen kan tilpasses løbende til den organisatoriske virkelighed. Her skal der dog være en verificerende instans, som kan følge op på, at ideen med udviklingsmetoden rent faktisk følges uden at være dikterende. Her er det oplagt at udvide den kommunikationsdedikerede med-

arbejders rolle til at indeholde disse verificerende elementer, hvorefter XP's grundlæggende værdisæt og intension kan bevares, selvom XP modificeres. Derfor må det konkluderes i forhold til det empiriske grundlag, at det på sigt vil være muligt for **m2i3D** at inkorporere XP i den organisatorisk funderede udviklingsmetode, som samtidig opfylder XP's grundlæggende værdisæt omkring kommunikation, enkelhed, feedback og mod, hvorfor ISM kan blive en realitet i den XP baserede organisation.

Training Program

TP er indeholdt i dette eksperiment, idet den teoretiske analyse kun illustrerer en begrænset dækning af dette KPA ved anvendelsen af XP som organisatorisk metode. Samtalen med universitetsgruppen efter forundersøgelsesfasen og workshoppen viser, at der sker deling af viden og en nivellering af kompetencer. Men i og med at workshoppen kun strakte sig over to dage, er det svært at konkludere endeligt, om der sker reelle kvantespring i forhold til en personlig forøgelse af viden. Der er ingen tvivl om, at XP udnytter den latente viden i en projektgruppe til det yderste ved udformningen af storys og tasks og ved det faktum, at alle sidder i det samme rum. Denne sammensætning medfører, at der hurtigt skabes et overblik over kompetencer og at man som leder hurtigt kan udfylde de huller som mangler i sammensætningen af grupper. XP skaber ved sin blotte anvendelse af redskaber det grundlag, hvorpå den enkelte leder kan vurdere, hvilke kurser der skal tilrettelægges, eller hvilken viden der skal indhentes i organisationen, hvorfor det må konkluderes, at selvom XP ikke direkte tilfører ny viden til organisationen, så skaber XP ved anvendelsen af redskaber det grundlag, hvormed ledelsen kan tilrettelægge reelle uddannelsesforløb. Det betyder, at TP faktisk er indeholdt i XP som udviklingsredskab og som organisatorisk redskab.

Opsummering

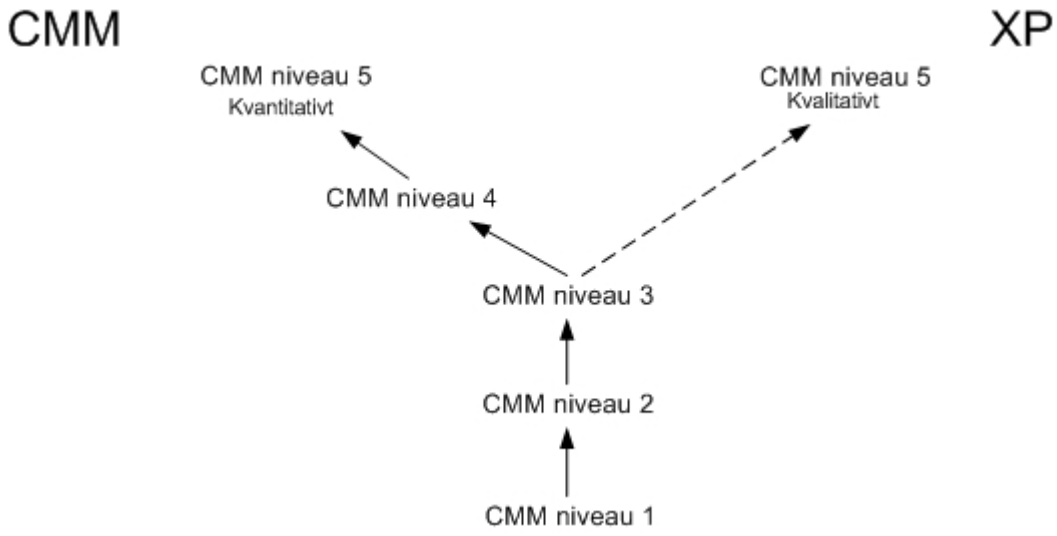
Den praktiske anvendelse af XP som organisatorisk værktøj i min egen virksomhed påviser, at de KPA på CMM niveau 3, som ikke er indeholdt i XP ifølge den teoretiske konklusion, relativt uproblematisk kan indeholdes i en organisation baseret på XP uden af gå på kompromis med XP's grundlæggende værdisæt. Jeg mener, at anvendelsen af XP i forskellige former i virksomheden illustrerer denne kombination, og at det selvfølgelig ville være ønskeligt med en mere detaljeret empiri end det har været tilfældet, for dermed at kunne konkludere endeligt. Eksperimentet tillader partitionelle observationer, men som alle andre teser er den reelle vurdering kun muligt gennem et længere forløb, hvor de alle analyserede KPA kan anvendes operationelt. Ifølge dette praktiske eksperiment er det muligt at etablere og certificere en organisation på CMM niveau 3 baseret på XP som softwareudviklingsmetode og som organisatorisk værktøj. Imidlertid er det særdeles interessant at videreføre dette eksperiment med en generel refleksion over det intentionelle CMM perspektiv, hvorfor diskussionen vil forfølge dette og vurdere, om XP kan understøtte de intentionelle mål med CMM modellens øverste niveauer.

Diskussion

"It is better to train people and risk they leave than do noting and risk they stay"
- Systematics videnregnskab 2004

Indledning

Formålet med denne diskussion er fortrinsvist at problematisere det store intentionelle skift mellem CMM niveau 3 og de to overliggende niveauer og samtidig problematisere den grundlæggende tanke med CMM niveau 4 og 5 og de redskaber med hvilke de enkelte KPA i disse niveauer vil effektuere denne intension. Ydermere vil diskussionen forsøge at vurdere, om intensionen med de kvantitative elementer i CMM niveau 4 og CMM niveau 5 er indeholdt i XP, og i hvor høj grad disse kvantitative elementer kan substitueres med kommunikative redskaber, som i højere grad kan forenes med XP som udviklingsmetode. Hypotesen er, at XP tilbyder redskaber, som intentionelt kan kvalificere en organisation på CMM niveau 5 uden de kvantitative redskaber, og således vise en anden farbar vej end den CMM modellen viser. Nedenstående model illustrerer dette:



Figur 9 Intentionel divergens mellem XP og CMM

Intensionen med CMM modellen

Først og fremmest er CMM modellen som tidligere nævnt et certificeringsbevis, som softwareorganisationen profileringsmæssigt kan anvende som garant for en sikker og stabil udviklingsproces i forhold til de krav, som de enkelte niveauer forudsætter. Denne certificering vil i høj grad anvendes af virksomheden til at differentiere sig fra andre lignende softwareudviklingsvirksomheder og samtidig vise, at virksomheden er i stand til af styre og lede en udviklingsproces uanset størrelsen af de enkelte projekter. Men CMM modellen viser sig ved den teoretiske analyse også at have indlejret en høj grad af styringsværktøjer frem for udviklingsværktøjer,

hvilket især gør sig gældende på CMM niveau 4 og 5. CMM modellen indeholder dog også en organisationsudviklende intention, som denne diskussion vil fokusere på. Der sker et markant skift fra CMM niveau 3 til CMM niveau 4, idet der fordres redskaber, som skal kvalitetssikre processer kvantitativt. Dette kvantificerende fokus flytter blikket fra medarbejderne til ledelsen i langt højere grad end CMM niveau 2 og 3, og delegeringen af ansvar til medarbejderne flyttes nu igen over til ledelsen af virksomheden.

Dette skift i intentionalitet flytter ikke alene ansvar fra medarbejdere til ledelse, men fjerner også en stor del af den tillid til medarbejderne, som er skabt på CMM niveau 3. Intentionen med CMM modellen er på alle niveauer at skabe en stabil udviklingsproces, som kan gentages i alle udviklingsprojekter og skabe en omskiftelig organisation, der kan tilpasses det enkelte udviklingsprojekt og stadig opfylde kravene til de enkelte niveauer i CMM modellen. Spørgsmålet er så, om CMM modellens øverste niveauer medfører en tilpasningsdygtig organisation, når de kvantitative redskaber tillægges så stor betydning, og hvor de kommunikative redskaber i XP tilsyneladende tilbyder lignende verificerende strukturer, hvilket vil diskuteres efterfølgende.

Den tilpasningsdygtige organisation?

I og med at de kvantitative elementer indføres i CMM modellens øverste niveauer, problematiseres den omskiftelighed, som etableres på både CMM niveau 2 og 3, og som er en væsentlig bestanddel af XPs grundstruktur. De kvantitative redskaber medfører en langt højere grad af styring end den ledelse, som trods alt har været fremherskende på CMM niveau 2 og 3. Denne styring medfører en markant nedgang i tillid til de medarbejdere, som trods alt skal udføre de opgaver, som udviklingsprocessen fordrer, og det er derfor tvivlsomt, om det i det hele taget er ønskeligt at konstituere en softwareorganisation på CMM niveau 4 eller 5, når der i så stor grad differentieres mellem ledelse og medarbejdere.

Målet med en tilpasningsdygtig organisation må være at have medarbejdere, som selvfølgelig er kvalificerede og fortrolige med udviklingsstrukturen, men som også har tillid til ledelsen og føler ansvar for udviklingsprojekterne. Ledelsen kan på CMM niveau 4 og 5 få et detaljeret billede af de kompetencer, udviklerne besidder og planlægge kommende sammensætninger af udviklere ud fra disse. Man kan selvfølgelig diskutere, hvorvidt den enkelte medarbejder er tilfreds med at blive koblet på nye udviklingsprojekter udelukkende baseret på disse kvantificerede billeder af organisationens samlede kompetencer, for dette er trods alt individuelt anlagt. Og for at en organisation kan sammensætte en gruppe, som arbejder godt sammen, skal man også tage tidligere projekter med i beregningerne og vurdere disse ud fra kvalitetskriterier. Alt dette indeholder både CMM niveau 4 og 5, og der er ingen tvivl om, at disse organisationer er i stand til at producere software af høj kvalitet. Denne kvantificerbare anvendelse skaber dog ikke nødvendigvis en tilpasningsdygtig struktur i organisationen. Den store kompleksitet, som er en uundgåelig del af enhver softwareudviklingsproces søges reduceret gennem anvendelsen af kvantitative redskaber, men denne kompleksitet kan ifølge Luhmann også reduceres gennem aktiv anvendelse af tillid som kompleksitetsreducerende redskab.

> *Trods alle bestræbelser på organisation og rationel planlægning kan al handling ikke blive ledet af en mere sikker beregning af dens virkninger. Der bliver ved med at være usikkerheder, der skal absorberes, og der må findes roller, som i særlig grad er pålagt denne opgave. Sådanne roller som f.eks. politikerens eller den ledende managers, bliver typisk ikke kontrolleret af standarter, men ved succes, fordi den nøjagtige handlen netop ikke*

> *kan erkendes tilstrækkeligt nøjagtigt i forvejen. Men succes kommer først efter der er handlet – ellers kommer den ikke.*
>
> [Luhmann, 1999, s. 61-62]

Den tilpasningsdygtige struktur handler grundlæggende om tillid til, at de mest kompetente personer sidder på de rigtige pladser (såvel medarbejdere som ledelse), og at de åbenlyst giver udtryk for deres lyster og kompetencer. Derved kan organisationen forvente af i hvert fald medarbejderne, at de er dedikerede til den task de skal løse, det projekt de skal lede eller styre, og at arbejdet fortrinsvist er af lyst og ikke af pligt. En sådan tilpasningsdygtig struktur handler fundamentalt om tillid. Tillid fra ledelsen til den enkelte medarbejder om at arbejdet udføres med den forventede kvalitet og indenfor estimaterne, og tillid fra den enkelte medarbejder til ledelsen om at de organisatoriske rammer til stede for at overholde estimaterne og sikre den forventede kvalitet. Dette er forudsætninger, som CMM modellen efter min overbevisning ikke nødvendigvis inkluderer, når de kvantitative redskaber medregnes. Luhmann tager jo netop udgangspunkt i en skelnen mellem system og omverden, altså i en forskel på "indeni" og "udenfor". Kun fordi organisationen gennem de interne processer arbejder selektivt ved at bearbejde og optage relevante omverdensdata, tjener de interne problemer i organisationen som arbejdsgrundlag for deres omverdenstilpasning. Ved at delegere ansvar og dermed tillid bliver organisationen i stand til at reducere kompleksitet internt i organisationen, og dermed bliver handling det symptom, som symboliserer kompleksitetsreduktionen. Denne kompleksitetsreduktion kan ifølge Luhmann ikke måles gennem almindelige arbejdsgange, men kun i arbejdsgange som indeholder beslutninger. Ved delegering af beslutningskompetence er kompleksitetsreduktionen en realitet. Luhmann udtaler i denne sammenhæng:

> *Tillid til andre mennesker indbefatter ikke længere uden videre, at deres verdenssyn betragtes som toneangivende. Man må lære at udholde "verdensanskuelsesmæssige" differencer og til trods for det tilslutte sin egen adfærd til de fremmedes selektionsydelser. Tilliden bliver så at sige privatiseret, psykologiseret og dermed baseret på noget individuelt-tolerant; eller den bliver specificeret funktionelt i form af kommunikationer af bestemt art, som den anden er efterviseligt kompetent i.*
>
> [Luhmann, 1999, s. 95]

Dette citat understøtter de definitioner af ledelse (s. 17), hvor organisationen har personer i rette stillinger, som gennem opretholdelsen af tillid udgør den sociale beslutningspræmis, og hvor organisationen opretholder sin funktion ved at foretage, formidle og implementere disse beslutninger.

Det er dog problematisk, at CMM modellen på niveau 4 indfører Quantitative Process Management (QPM) og Software Quality Management (SQM), som skal måles vha. kvantitative værktøjer. På de foregående niveauer er der etableret metoder, som sikrer, at de nødvendige procedurer er til stede for at sikre både kvalitet og proces. Disse processer fungerer på niveau 3 og sikrer både videndeling, planlægning, sporing, koordination og reviews, som alle er essentielle i en udviklingsproces. Med både QPM og SQM er intensionen at lægge et yderligere lag på stort set alle foregående KPA på niveau 2 og 3 for at sikre, at der genereres kvantitative data. Dette lag tilfører en ekstra kompleksitet til udviklingsprocessen, som pålægges den enkelte projektleder og den enkelte medarbejder. Dermed flyttes en del af arbejdstiden til administrative opgaver frem for udvik-

lingsmæssige opgaver, hvormed projekterne i den enkelte virksomhed ressourcemæssigt forøges. Dertil skal medregnes, at de kvalitetssikrende data oftest ikke vurderes ud fra brugeranvendelse og brugertilfredshed men i højere grad i antallet af fejlrettelser, tilpasningsopgaver og gennemløbstid, som alle er direkte kvantificerbare. Hvorvidt disse data i sig selv er garant for kvalitet er særdeles tvivlsomt. I denne sammenhæng kan nævnes flere projekter, hvor systemerne fungerer fejlfrit i forhold til de kvantitative data, men hvor den direkte brugeranvendelse er både mangelfuld og misinformerende. Her kan nævnes Siebel CRM systems, som er verdens absolut største CRM leverandør, men hvor designet af brugerinterfacet vanskeliggør en logisk arbejdsproces, og arbejdsformidlingens AMANDA system, som i starten led af de samme børnesygdomme.

På niveau 5 er intensionen med CMM modellen at videreudvikle processer, redskaber og strukturer løbende igennem projekterne, så organisationen er i stand til at afspejle ændringer i omverden, nye krav, nye metoder og nye standarter og deslige; kort sagt, at organisationen skal lære at lære. Niveau 5 anvender i høj grad de kvantitative redskaber fra niveau 4, som kan skabe detaljerede og umiddelbare overblik over mindstedele i organisationen, lige fra placering af kompetencer til målinger af kvalitet i de enkelte udviklingsprojekter. Men her sker der igen et afbræk i delegering af ansvar og tillid i CMM modellen, for på niveau 5 anvendes den kvantitative viden fra niveau 4 til eksempelvis at forbedre processer og systematisk at eliminere processer, der generelt genererer fejl i den udviklede software. Det er selvfølgelig en fordel for enhver organisation at kunne eliminere disse processer, og det vil være ønskværdigt at stræbe imod denne eliminering af procedurer, redskaber eller metoder, som generer fejl i det færdige system[13]. Det virker særdeles usikkert fra et humanistisk synspunkt om de kvantitative data kan tilføre en sådan revideringsproces med informationer, som når ud over det kvantificerbare og er i stand til at identificere fejlgenererende strukturer, som eksempelvis er affødt af dårlig eller mangelfuld kommunikation fra ledelsen, dårlige begrundelser for indførelse af ny teknologi eller metoder eller andre områder, som i kraft af omstruktureringer i organisationen medfører frustrationer for medarbejderne, mellemlederne eller endda hos ledelsen.

Spørgsmålet er, om kvantitative data reelt kan sikre, at beslutningstagerne er i stand til at reducere frustrationer, dårlige kommunikative arbejdsgange og i det hele taget skabe det generelle overblik over organisationen, som kan forbedre uhensigtsmæssigheder på et organisatorisk strategisk niveau. Ydermere er det tvivlsomt, om disse kvantitative værktøjer kan sikre et stabilt beslutningsgrundlag uden kommunikative verificerende instanser. Det bør organisatorisk problematiseres ved anvendelse af kvantitative redskaber, hvilken viden der kan repræsenteres, og hvad der bør udledes heraf. Derfor vil det i næste afsnit tilstræbes at anskueliggøre de kommunikative strukturer, som godt nok ikke genererer kvantitative data, men som i kraft af deres åbne kommunikation skaber overblik over organisationen og samtidig minimerer arbejdsgange, der genererer fejl. Samtidig kan disse kommunikative strukturer formodentlig sikre et bedre beslutningsgrundlag, idet viden kan repræsenteres mere hensigtsmæssigt end enkeltstående kvantitative målinger.

[13] Et interessant spørgsmål i denne kontekst er hvorvidt kvantitative mekanismer kan fjernes, hvis de øger kompleksiteten uhensigtsmæssigt, hvormed organisationen mister sin certificering på CMM niveau 4. Det er ikke noget Jalote diskuterer, men som kunne være en organisatorisk virkelighed især ved organisationer baseret på XP.

Den kommunikative organisation

I forbindelse med det intensive arbejde med både Extreme Programming og CMM modellen i indeværende speciale, er det interessant at observere, at XP teoretisk i sin grundstruktur tilbyder organisationen procesforbedrende kommunikative redskaber, som mindsker fejl i den udviklede software og ikke anvender kvantificerende målinger til at identificere disse fejlgenererende processer. XP trives bedst i en organisation, som anvender åben og ligeværdig kommunikation mellem ledelse og medarbejdere, og hvor ledelsen har både vilje og forstand til at undersøge de procesforbedrende forslag, som medarbejdere eller ledere konfronteres med i forbindelse med udviklingen. At XP ikke anvender kvantitative strukturer betyder selvfølgelig, at organisationer baseret på XP ikke kan certificeres på CMM niveau 4, men det betyder ikke, at disse organisationer ikke kan efterleve de intentionelle forventninger til en organisation på CMM niveau 5. Det kræver selvfølgelig, at man undlader de kvantitative strukturer, som er forventet af en organisation på CMM niveau 4, hvorved en reel certificering umuliggøres.

Men hvordan forventes organisationer så at udøve de reelle processuelle ændringer, som afkræves af de identificerede problematiske områder indenfor eksempelvis Process Change Management (PCM)? Der eksisterer umiddelbart to mulige organisatoriske effektueringer: **1.** Beslutningstageren kan handle udelukkende ud fra de kvantitative data, som bl.a. identificerer processuelle problemer. Det betyder, at beslutningstageren i høj grad handler i overensstemmelse med en subjektiv overbevisning om, at en bestemt handling eller procesændring vil forbedre disse områder, hvilket bygger på en instrumentel rationalitet. **2**. Beslutningstageren kan tage udgangspunkt i de kvantitative data og følge op på disse med kommunikative redskaber som eksempelvis møder eller samtaler for at forbedre fundamentet for beslutningsgrundlaget for de kommende processuelle forbedringer, hvilket betyder, at denne tilgang bygger på en mere proceduremæssig rationalitet, hvor konteksten har afgørende betydning. Det handler dermed i høj grad om repræsentation af viden; om hvor tæt den repræsenterede viden er på det, den faktisk repræsenterer, altså hvilke egenskaber den valgte tilgang ekspliciterer, og hvilke den udelader. Derfor bør repræsentation af viden som grundlaget for beslutninger diskuteres organisatorisk; både hvad der kan udledes og hvad der bør udledes heraf.

Som humanist virker det uforståeligt, at beslutningstagere træffer valg baseret udelukkende på kvantitative målinger, men virkeligheden er, at beslutningstagere på kort tid skal træffe organisatoriske beslutninger, som påvirker mange mennesker, og for at overskue en stor organisation er kvantitative data en overkommelig måde at skabe dette overblik. Det undrer mig derfor meget, at CMM modellen på de øverste niveauer tillægger de kvantitative værdier så stor indflydelse i stedet for at fokusere mere på grundlaget for de kvantitative data. Der er ingen tvivl om, at hvis de kvantitative data understøttes af kvalitative informationer (fra eksempelvis de berørte personer), er der langt større mulighed for at tilrettelægge reelle processuelle forbedringer, som er i overensstemmelse med medarbejdernes virkelighed og ikke en subjektiv overbevisning baseret på tal. Derfor er det min overbevisning, at CMM modellen burde indføre en verificerende struktur, som i højere grad fokuserer på verificering af beslutningsgrundlag end på indsamlingen af relevante kvantificerbare data. Hertil kan XP muligvis tilbyde en løsning.

En XP håndtering af CMM niveau 5

Som udgangspunkt er der i dette tænkte eksempel tale om en organisation certificeret på CMM niveau 3 (hvilket den hidtidige analyse og det praktiske eksperiment har identificeret som muligt), baseret på XP som udviklingsværktøj, og hvor en del af ledelsesværktøjerne tager udgangspunkt i XP metodologien. Det betyder kommunikativt, at der er daglige møder, hvor udviklerne i samarbejde med lederne udvælger dagens tasks, og hvor der ugentligt er et møde, hvor ugens stories udvælges i samarbejde med kunden. Der er selvfølgelig ikke tale om en certificering på CMM niveau 5, da CMM niveau 5 afkræver en certificering på CMM niveau 4. I dette tænkte eksempel springes niveau 4 simpelthen over, da niveauet indeholder kvantitative redskaber, som i forvejen er umulige at indeholde i en XP organisation, som illustreret i den teoretiske konklusion. I dette eksempel er intentionen at undersøge, om det er muligt at opfylde de intentionelle krav på CMM niveau 5; de intentioner som skaber organisationsudvikling frem for certificering. For at opfylde kravene til niveau 5 i CMM modellen skal de tre KPA opfyldes; Defect Prevention (DP), Technologi Change Management (TCM) og Process Change Management (PCM). Det interessant er nu, om XP kan tilbyde kommunikative redskaber, som kan opfylde disse tre kriterier uden at indføre de kvantitative strukturer, som er sprunget over i CMM niveau 4.

DP er, som beskrevet i den teoretiske analyse (s. 63), delvist indeholdt i XP, dog med undtagelse af de kvantitative redskaber fra CMM modellen. Der mangler altså kun værktøjer, som kan skabe overblik over organisationen for at identificere områder, som gentagne gange genererer fejl i den udviklede software. Her vil den XP baserede organisationen sandsynligvis udnytte den iboende tillid til medarbejderne, som i kraft af den distribuerede ansvarsfordeling i XP medfører, at medarbejderne kan træffe beslutninger inden for deres kompetenceområde. Denne tillid udmønter sig i et åbnet kommunikationsforum, hvor medarbejderne kan adressere udviklingsmæssige problemer direkte til lederne og derved påpege processuelle forbedringer, der vil mindske antallet af fejl i den pågældende proces, hvilket henleder opmærksomheden på PCM, som således også er indeholdt i dette åbne kommunikative forum. Herefter vil den pågældende leder udnytte det fælles ejerskab af både kode og procesbeskrivelser til at pålægge den pågældende medarbejder at afsætte tid til, i samarbejde med en anden person, at forbedre den organisatoriske projektstruktur for at imødekomme disse procesforbedrende tiltag. Der er altså tale om en kommunikativ kultur, som løbende identificerer processuelle forbedringer gennem de daglige eller ugentlige møder, og hvor de forbedrende tiltag effektueres af de personer, der har kompetence herfor. Lederens rolle er i denne henseende at skabe en ressourcefordeling, som kan tilgodese den udnyttelse af iboende viden hos medarbejderne for at sikre denne løbende verificering og forbedring af organisationens udviklingsproces. Hermed ser jeg, at XP kan tilgodese CMM modellens intension med både DP og PCM uden at tilføje yderligere kompleksitet i form af kvantitative måleværktøjer.

Som beskrevet i den teoretiske analyse (s. 65) afkræver TCM en selvstændig afdeling, der kan afprøve nye teknologier; også i forhold til de organisatoriske udviklingsprocesser. Det kan i denne sammenhæng meget vel tænkes, at organisationen i forbindelse med de procesforbedrende kommunikative tiltag, at nye teknologier i denne sammenhæng kan undersøges med henblik på at udnytte nye teknologier i organisationens udviklingsproces. Det kræver dog igen, at organisationen beslutter sig for at afsætte både økonomiske ressourcer og medarbejderressourcer til at håndtere disse teknologiske tiltag. TCM afkræver dog igen kvantitative tiltag, men disse kan med fordel omdannes til en del af den verificerende kommunikation, der på de ugentligt møder

vurderer mulige teknologier, der kan afhjælpe procesmæssige problemer eller generering af fejl. TCM kan uproblematisk indføres i en organisation baseret på XP, men afhænger udelukkende af, at den detaljerede beskrivelse af den organisatoriske udviklingsproces indeholder teknologivurdering i forbindelse med forbedringer. Jeg ser ingen grund til at spilde ressourcer ved kvantitativt at estimere de enkelte teknologiske muligheder for at en leder via et kvantificeret overblik kan vurdere, om et givent tiltag er den bedste løsning. Dette ansvar er iboende i den tillid til den enkelte medarbejder, som er dedikeret til den enkelte opgave, hvilket er iboende i XP.

Opsummering

Den fremherskende intension med CMM modellen som helhed er uden tvivl på sigt at certificere organisationer på niveau 5, der imødekommer krav om omskiftelighed og kontinuerlige procesforbedringer både på et organisatorisk strukturelt niveau og på et teknologisk niveau. CMM modellen indfører allerede på niveau 4 kvantitative redskaber, som øger kompleksiteten i de eksisterende metoder og procedurer, som er etableret på niveau 2 og 3, som garant for, at disse procesforbedrende tiltag kan håndteres og vurderes ud fra kvantitative målinger. Det forbliver usagt hos Jalote, hvor mange ekstra ressourcer, der skal anvendes, for at opfylde disse kvantificerbare krav, men der er ingen tvivl om, at projekter i den enkelte organisation afkræver væsentligt flere ressourcer, end hvis organisationen vælger ikke at indføre disse kvantitative redskaber. Dette skal selvfølgelig sammenholdes med den forbedrede kvalitet i den udviklede software, og det skal selvfølgelig afgøres ved en cost/benefit analyse, om disse kvantitative redskaber bør indføres. Men det er interessant, at en organisation baseret på XP teoretisk kan opfylde kravene til et isoleret CMM niveau 5, ved at indlejre intensionerne ved niveau 5 i de allerede eksisterende kommunikative strukturer, som er til stede i en XP organisation. Når man bibeholder tilliden til den enkelte medarbejder ved at delegere og distribuere ansvar til de personer, der alligevel har den største kompetence inden for feltet, er det kun nødvendigt for den enkelte leder at sikre konsistens i organisationens standardiserede udviklingsproces. Luhmann udtrykker det således:

> *Tillid er overhovedet kun mulig, hvor sandhed er mulig, hvor mennesker med forpligtigelse overfor tredjepart kan komme til forståelse af det samme. Sandhed letter denne indbyrdes forståelse og dermed reduktion af kompleksitet, ved at det er underforstået, at også tredjeparten ville anse opfattelsen for rigtig.*
>
> [Luhmann, 1999, s. 99]

Dermed sikrer ledelsen, at de kommunikative redskaber i XP også fokuserer på procesforbedrende tiltag som en del af den daglige eller ugentlige kommunikation og kan derved opfylde samtlige intentioner med CMM modellens niveau 5. Herved mener jeg at kunne konkludere, at de kvantitative værktøjer, som primært er at finde på CMM modellens niveau 4 kan undværes i de organisationer, der anvender XP som udviklingsmetode og som organisatorisk redskab, hvormed XP intentionelt kan opfylde kravene til en organisation på CMM niveau 5.

Konklusion

No battle plan ever survives contact with the enemy
- en af Murphys militærlove

Som både den indledende test, den teoretiske analyse og det praktiske eksperiment illustrerer, er der en spændende og meget sigende sammenhæng mellem Extreme Programming og Capability Maturity Model; ikke mindst på det intentionelle plan. Den teoretiske analyse har i den grad vist, at XP indeholder størstedelen af de krav til en softwareudviklingsorganisation, som fordres af CMM modellen på niveau 3. De to områder, som ikke er indeholdt (Software Subcontract Management og Integrated Software Management), kan uden de store problemer indeholdes i XP, som det illustreres i det praktiske eksperiment, uden at det strider imod XP's grundlæggende værdier. De store problemer fremkommer ved de to øverste niveauer af CMM modellen, hvor de kvantitative redskaber er særdeles dominerende. XP kan ikke indeholde disse styringsmæssige mekanismer, som illustreret i diskussionen, idet indførelse af disse redskaber vi stå i skarp kontrast til XP's grundlæggende værdisæt, hvorfor det kan diskuteres (som i diskussionen), hvorvidt kvantitative værktøjer er vejen frem for en XP organisation.

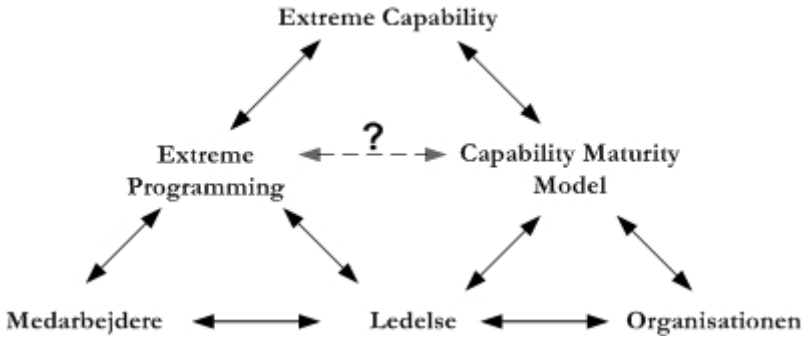
Der er ingen tvivl om at organisationer, som er certificeret på CMM niveau 5, gentagne gange er i stand til at udvikle funktionel software til kunder indenfor de angivne økonomiske rammer, hvilket stemmer overens med intensionen i CMM modellen. Men skiftet i intentionalitet fra CMM niveau 3 til CMM niveau 4, hvor tilliden til medarbejderne i den grad fjernes ved indførelsen af kvantitative styringsværktøjer, er særdeles problematisk; også i forhold til XP. Det er min opfattelse gennem arbejdet med XP og CMM modellen, at XP udmærket kan opfylde de intentionelle krav på CMM model 5 uden indførelse af kvantitative styringsredskaber. Intentionen med CMM modellen er i sidste instans at skabe en tilpasningsdygtig organisation, som er i stand til at tilpasse sig omverdensændringer, konstruere organisatoriske procesforbedrende tiltag på metodisk niveau og samtidig indføre teknologiske vurderingsredskaber i de enkelte projekter. Disse egenskaber for en organisation kan indføres med XP som organisatorisk og udviklingsmæssigt redskab uden at tilføje kvantitative redskaber. Det handler udelukkende for en organisation baseret på XP om at ville denne fremgangsmåde mere end det handler om at måle og veje den enkelte medarbejder igennem projekterne. Ansætter man som organisation en medarbejder, må man kort sagt have tillid til, at denne medarbejder kan opfylde sin rolle.

Og netop tilliden, eller mangel på samme, er den faktor som adskiller XP værdier fra CMM værdier, hvilket også er illustreret igennem indeværende speciale. Tillid er den kompleksitetsreducerende faktor, som tillader softwareorganisationen at give tillid til medarbejderne gennem distribuering af ansvar og dermed opnåelse af personlig motivation for organisationens projekter. Det er en særdeles problematisk indgangsvinkel for en softwareorganisation først at distribuere ansvar på CMM niveau 3 for derefter at fjerne dette ansvar på CMM niveau 4 og derigennem tilliden til den enkelte medarbejder. Der er ingen tvivl om, at denne fremgangsmåde kan effektueres, men hvorvidt der er tale om den samme skare af medarbejdere før og efter effektueringen er en helt anden sag. Jeg er personligt af den opfattelse, at medarbejdere i en organisation certificeret på CMM niveau 3 ikke nødvendigvis er indstillet på anvendelsen af de styringsmæssige redskaber og mekanismer, som indføres på ovenstående niveauer, men dette spørgsmål vil jeg lade stå ubesvaret hen, da det ikke muligt at

konkludere endeligt, dels på grund af specialets fokus, dels grundet den manglende understøttende empiri ud fra et komplet udviklingsforløb.

Som beslutningsunderstøttende middel er CMM modellen på alle niveauer et udmærket værktøj, som dog primært forholder sig til en substantiv rationalitet, hvor alt er beskrevet og hvor det antages, at ingen mangel på information gør sig gældende. I forhold til XP, som bygger på en kognitiv og proceduremæssig rationalitet, hvor beslutninger træffes kontekstafhængigt, er der stor divergens med CMM modellen, og dette er problematisk, hvis man anvender CMM modellen som organisationsudviklende værktøj. Anvendes CMM modellen med certificering som mål, kommer man ikke uden om de kvantitative værktøjer, som bevirker den substantive rationalitet, men anvender man den organisationsudviklende intention i CMM modellen, er den substantive rationalitet problematisk. Den substantive rationalitet virker som en forældet organisations- og beslutningsstruktur, der ikke tager højde for de kommunikative strukturer, som muliggør foranderlighed og tilpasning af organisationen, som eksemplificeret med XP som organisationsudviklende middel. Uden de kvantificerende værktøjer umuliggøres en CMM certificering, men med anvendelsen af kvantitative værktøjer vanskeliggøres en tilpasningsdygtig organisation. Dette er en problematik, som ikke adresseres i den anvendte litteratur, men som diskussionen i indeværende speciale pointerer. CMM modellen kan ganske vist anvendes til at opnå en tilpasningsdygtig organisation, men analysen vurderer, at der skal anvendes langt flere ressourcer, end hvis organisationen anvender kommunikative redskaber med samme formål.

Fokus i specialet har derimod hele vejen igennem været styret af en undren om, hvorvidt XP kan supportere de organisatoriske beslutninger, der er medvirkende til at konstituere en organisation på et givent CMM niveau, og hvorvidt XP i denne funktion kan anvendes som organisatorisk værktøj. Gennem den teoretiske analyse stod det klart, at XP indeholder en række elementer, som opfylder kriterier i CMM modellens niveau 3, dog med undtagelse af to KPA. Disse to KPA kunne dog igennem det praktiske eksperiment med få udvidelser i XP metodologien indeholdes indenfor XP's grundlæggende værdisæt, hvorfor konklusionen må være, at XP kan supportere organisatoriske beslutninger, som konstituerer en organisation på CMM niveau 3. Det står dog samtidig klart, at XP ikke kan indeholde kvantitative redskaber, hvorfor XP ikke kan supportere en organisatorisk certificering på CMM niveau 4 eller 5. Det interessante er dog imidlertid, som den efterfølgende diskussion illustrerede, at XP uden problemer kan efterleve intensionen med en organisation på CMM niveau 5 isoleret uden at indføre de kvantitative styringsredskaber. Derfor vil jeg afslutte med at beskrive den organisation, som gennem XP kan opfylde de intentionelle kriterier for en organisation på CMM niveau 5; en organisation som jeg vælger at betegne med *Extreme Capability*.



Figur 10 Extreme Capability

For at opnå Extreme Capability skal der først arbejdes særskilt med henholdsvis XP og CMM modellen. For XP's vedkommende handler det mest om samspillet mellem ledere og medarbejdere, som i den grad skal efterleve principperne i XP, hvorefter ledelsen tilretter organisationen, så den opfylder intensionen med CMM modellen som organisationsudviklende redskab og ikke de enkelte KPA i de øverste niveauer. Det kan dog ikke konkluderes endeligt, om der er redskaber, som forbinder XP og CMM direkte, for indeværende speciale kan kun konkludere, at der kan opstå et samspil, men kun igennem medarbejdere, ledelse og organisationen. Det interessante ligger i at udvikle eller konstruere kommunikative redskaber, som kobler CMM og XP med hensyntagen til begges præmisser, hvilket i dette speciale ikke er muligt. Der er dog ingen tvivl om, at en mere kommunikativ tilgang til repræsentation af viden i højere grad ekspliciterer viden, som i højere grad kommer tættere på "den virkelige verdens" repræsentationelle udtryk end den udelukkende kvantitative repræsentation.

Det betyder, at det praktiske eksperiment med den teoretiske analyse in mente illustrerer, at XP udmærket kan anvendes til at kvalificere og konstituere en organisation til at eksistere på CMM 3 og formodentlig også få den certificering, som udadtil er det bevis, som anvendes i kontraktforhandlinger, hvormed organisationen kan skille sig markant ud fra andre softwareudviklingsvirksomheder. Med XP som grundstruktur i organisationen og i softwareudviklingen kan organisationen dog ikke nærme sig en CMM certificering på højere niveauer, da de kvantitative redskaber ikke kan indføres i en XP organisation, idet det strider mod XP's grundlæggende værdier. XP tilbyder gennem verificerende instanser beslutningsunderstøttende redskaber, som grundlæggende kan medvirke til at konstituere en organisation. De tanker, som ligger til grund for CMM niveau 5, er exceptionelle, og burde være en eksplicit bestanddel af XP udvikling på et organisatorisk niveau, hvilket jeg i den grad anbefaler. XP kan sagtens som udviklingsmetode indeholde både DP, TCM og PCM uden at anvende kvantificerbare redskaber, hvilket betyder, at XP gennem de verificerende instanser i redskaberne og værdierne er i stand til at opfylde de intentionelle kriterier for CMM niveau 5 som organisationsudviklende værktøj, uden dog at kunne certificeres på dette niveau grundet de manglende kvantitative redskaber.

Denne vurdering tyder altså på, at XP udmærket kan anvendes som organisatorisk værktøj, hvor der dog skal konstrueres kommunikative verificerende redskaber, som kan koble CMM modellen og XP direkte. De kommunikative redskaber bør grundlæggende baseres på deling af viden på tværs af organisationen og gennem medarbejderne udnytte organisationens iboende viden; den viden som medvirker til, at organisationen opnår Extreme Capability.

Litteraturliste

Verden og indholdet af min hjerne kan ikke skilles ad.
- Svend Åge Madsen

Auer, Ken & Miller, Roy
Extreme programming Applied – playing to win
Addison-Wesley, 2000

Beck, Kent
Extreme programming explained
Addison-Wesley, 2000

Beck, Kent & Fowler, Matin
Planning Extreme Programming
Addison-Wesley, 2000

Cockburn, Alistair
Agile Software Development
Pearson Education Inc., 2002

Dahlbom, Bo og Mathiassen, Lars
Computers in Context – The Philosuphy and Practice og Systems Design
Blackwell, 1995

Davis, R. Schrobe, H. og Szolovits, P.
What is a Knowledge Representation?
AI Magazine, 14(1), 1993.

Dreyfus & Dreyfus
Mind over Machine
The free press, 1986

Føllesdal, Dagfinn, Walløe, Lars og Elster, Jon
Politikens bog om moderne videnskabsteori
Politikens forlag A/S, 1.udgave, 3.oplag, 1997

Jalote, Pankaj
CMM in Practice
Higher Edication Press, år ukendt

Latour, Bruno

A Collective of Humans and Nonhumans,
Pandoras´s Hope, Cap. 6.
Harvard University Press, 1999

Luhmann, Niklas

Tillid
Hans Reitzels forlag, 1999

Luhmann, Niklas

Sociale systemer – Grundrids til en almen teori
Hans Reitzels Forlag, 2000

Lytje, Inger

Software som tekst – en teori om systemudvikling
Aalborg Universitetsforlag, 2000

Mathiassen, Lars

Objektorienteret Analyse og design
2. udgave, Forlaget Marko Aps, 1998

March, James G.

A Primer on Decision Making
The Free Press, 1994

March, James G.

Fornuft og forandring – ledelse i en verden beriget med uklarhed
Samfundslitteratur, 1995

Marchesi, Michele, Succi, Giancarlo m.fl

Extreme Programming Perspectives
Addison-Wesley, 2003

Mathiassen, Lars, Pries-Heje, Jan og Ngwenyana, Ojelanki

Improving Software Organizations
Addison-Wesley, 2002

McConnell, Steve

Rapid Software Development
The Microsoft Press, 1996

McConnell, Steve

Software Project survival guide
The Microsoft Press, 1998

McConnell, Steve

After the Goldrush
The Microsoft Press, 1999

Newkirk, James og Martin, Robert C.

Extreme Programming in Practice
Addison-Wesley, 2001

Noorderhaven, Niels, G.

Strategic Decision Making
Addison-Wesley, 1995

Rosenstand, Claus A. Foss

Kreation af narrative multimediesystemer
Samfundslitteratur, 2002

Succi, Giancarlo og Marchesi, Mechele

Extreme Programming Examined
Addison-Wesley, 2001

Thyssen, Ole

Iagttagelse og paradoks
Gyldendal, 1997

Thyssen, Ole

Værdiledelse – om organisationer og etik
Gyldendal, 2000

Qvortrup, Lars

Mellem kedsomhed og dannelse – variationer over et tema af Pico
Odense Universitetsforlag, 1996

Wenger, Etienne

Communities of practice – Learning, Meaning and Identity.
Cambridge University Press. 1998

Åkerstrøm, Andersen, Niels

Kærlighed og omstilling

Nyt fra samfundsvidenskaberne, 2001

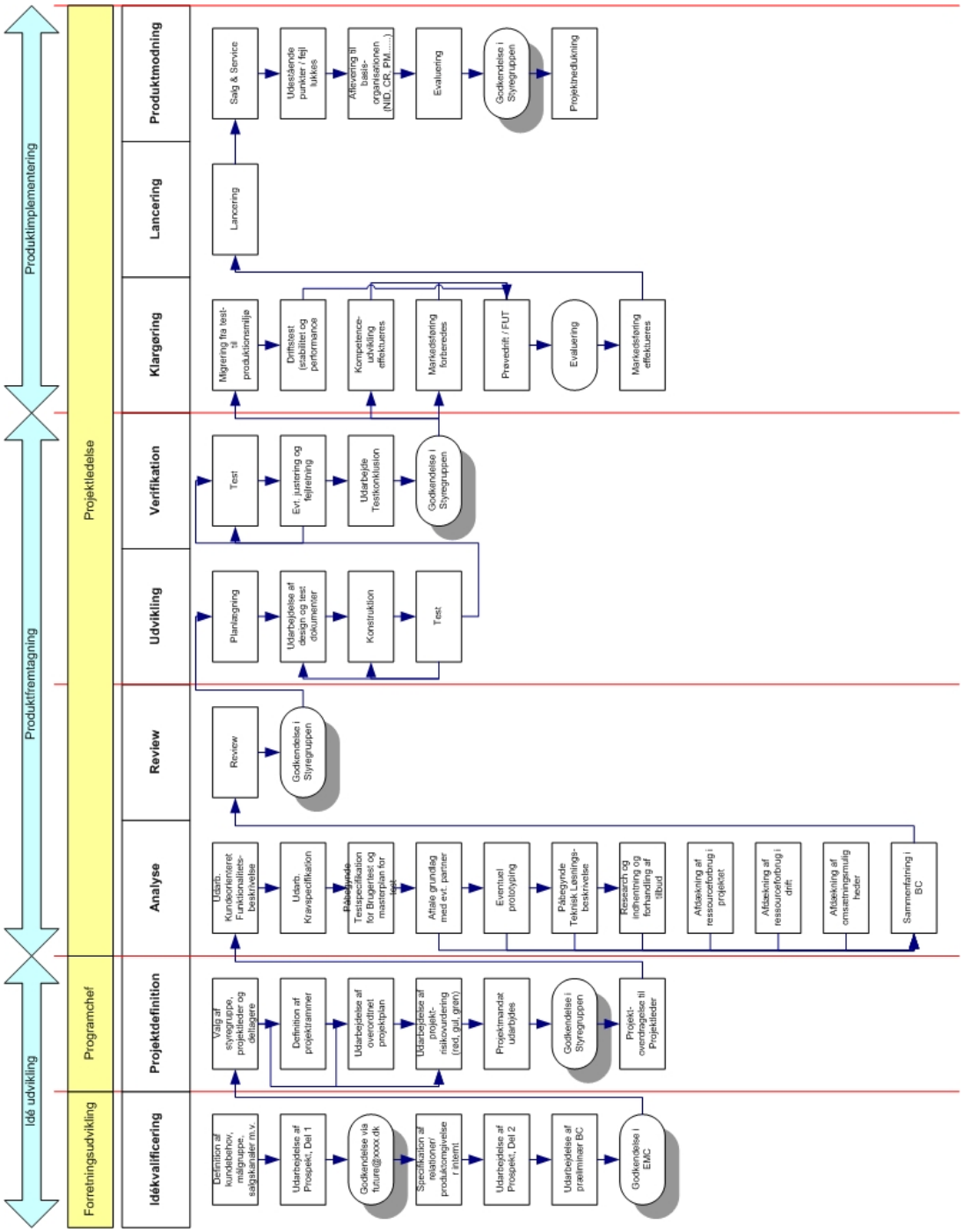
Forkortelser

Those are my principles. If you don't like them, I have others.

- Groucho Marx

BSC – Balanced Score Card
CMM – Capability Maturity Model
CMS – Content Management System
DP – Defect Prevention
IC – Intergroup Coordination
ISM – Integrated Software Management
KISS – Keep It Simple, Stupid
KPA – Key Process Area
OPF – Organization Process Focus
OPD - Organization Process Definition
PR – Peer Revies
PCM – Process Change Management
QPM – Qualitative Process Management
RM - Requirement Management
SCM - Software Configuration Management
SPI – Software Product Engeneering
SPP - Software Project Planning
SPTO - Software Project tracking and oversight
SQA – Software Quality Assurance
SQM – Software Quality Management
SSM - Software Subcontract Management
TCM – Technology Change Management
TP – Training Program
XP – Extreme Programming

Bilag 1- traditionel softwareudvikling



Bilag 2 – CMM oversigt

Capability Maturty Model Level 2 (Repeatable)

Key Process Area	Goals
Requirements Management	• Software requirements are controlled to establish a baseline for software engineering and management activities. • Software plans, products and activities are kept consistent with requirements.
Software Project Planning	• Estimates are documented for use in planning and tracking the project. • Project activities and commitments are planned and documented. • Affected groups and individuals agree to their commitments related to the project.
Software Project Tracking and Oversight	• Actual results and performances are tracked against the software plans. • Corrective actions are taken and managed to closure when actual results and performances deviate significantly from the software plans. • Affected groups and individuals agree with the changes to commitments.
Software Subcontract Management	• The prime contractor and the subcontractor agree to their commitments. • The prime contractor tracks the subcontractor's actual results against its commitments. • The prime contractor and the subcontractor maintain ongoing communication. • The prime contractor tracks the subcontractor's actual performance against its commitments.
Software Quality Assurance	• Software quality assurance activities are planned. • Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively. • Affected groups and individuals are informed of software quality assurance activities and results. • Noncompliance issues that cannot be resolved within the project are addressed by senior management.

Software Configuration Management	• Software configuration management activities are planned. • Selected software work products are identified, controlled and available. • Changes to identified software work products are controlled. • Affected groups and individuals are informed of the status and content of software baselines.

Capability Maturty Model Level 3 (Defined)

Key Process Area	Goals
Organization Process Focus	• Software process development and improvement activities are coordinated across the organization. • The strengths and weaknesses of the software processes used are defined. • Organization-level process development and improvement are planned.
Organization Process Definition	• A standard software process for the organization is developed and maintained. • Information related to the organization's standard software process by the software projects are collected and reviewed and made available.
Training Program	• Training activities are planned • Training for developing skills and knowledge needed to perform software management and technical roles is provided. • Individuals in the software engineering group and software-related groups receive the training necessary to perform their jobs.
Integrated Software Management	• The projects defined software process is a tailored version of the organization's standard software process. • The project is planned and managed according to the projects defined software process.
Software Product Engineering	• The software engineering task are defined, integrated and consistently performed to produce the software. • Software work products are kept consistent with each other.
Intergroup Coordination	• All affected groups agree to the costumer's requirements. • All groups agree to the commitments between different groups. • The groups identify, track and resolve intergroups issues.
Peer Reviws	• Peer review activities are planned. • Defects in the software work products are identified and removed.

Capability Maturty Model Level 4 (Managed)

Key Process Area	Goals
Quantitative Process Management	• The quantitative process management activities are planned. • The process performance of the project's defined software process is controlled and defined. • The process capability of the organization's standard software process is known in quantitative terms.
Software Quality Management	• The projects software quality activities are planned • Measurable goals for software product quality and their priorities are defined. • Actual process toward achieving the quality goals for the software products is quantified and managed.

Capability Maturty Model Level 5 (Optimizing)

Key Process Area	Goals
Defect prevention	• Defect prevention activities are planned • Common causes of defect are sought and identified. • Common causes for defect are prioritized and systematically eliminated.
Technology change management	• Incorporation of technology changes is planned. • New technologies are evaluated to determine their effect on quality and productivity. • Appropriate new technologies are transferred into normal practice across the organization.
Process change management	• Continuous process improvement is planned. • Participation in the organization's software process improvement activities is organization-wide. • The organization's standard software process and the project's defined software process are improved continuously.

Bilag 3 - Empiri fra workshop

Case: Videndeling
Metode: The planning game (TPG)
Deltagere: Nina, Lars, Dennis (kun den 20.), Nikolaj, Jonas og Kresten

Dagens forløb:

Mødte kl. 9
Casen var videndeling i en organisation, hvor den viden, der kan deles, skal ses som den viden, der kan repræsenteres/formaliseres, f.eks. på skrift.
Kresten påbegyndte et TPG forløb og sikrede, at vi fulgte metoden.
Lavede brainstorm på tre tavler, hvorved vi fik defineret syv opgaver inden for videndeling, samt dannet konsensus om begrebet videndeling og hvilken viden vi mente der var relevant.
Herefter meldte hver person ud hvilke opgaver han/hun fandt mest interessant, hvorved par blev dannet. Til en enkelt opgave manglede der en frivillig, så der måtte tildeles én.
Frokostpause (xp?)

Spørgsmål?

- Skal arbejdsopgaverne repræsenteres i systemet?
- Hvor meget skal repræsenteres?
- Skal alle tavlerne repræsenteres?

Første tavle: Hvad skal vi overveje ved videndeling i en organisation?:

Hvilken viden? – Hvor meget?
Til hvem?
Hvornår?
Hvordan? – Form?
Hvorfor?
Hvad kræver det af org.?
Personer
Strukturelt
Procedurelt
Ansvar (fordeling af)
Arbejdsmiljø
Kompetence

Formelt/uformelt

Anden tavle: Hvilken viden?

Faglig viden (indiv.)
Fagspecifikt
Faglige egenskaber
Faglige erfaringer
Uddannelse
Målbarhed?
Jobbeskrivelse
Personligt
Interesser
Hobby
Fritid
Erfaringer

Organisatorisk viden (indiv.)
Idegrundlag
Mål/koncept
Hvem laver hvad? Nu/tidligere
Tidligere gruppesammensætninger
Tidligere projekter/iterationer
Tidligere par (PP) – resultat/kvalitet
Mål (beslutninger) til opgavestyring

Organisationens erfaring (fælles)
Problemløsninger
Tidl. Projekter
Budget
Deadlines
Progression af nuværende projekter/iterationer
Formålsbeskrivelse (idegrundlag)
Org. Struktur
FAQ
Private forpligtelser
"social koncept/normer"

Tredje tavle: Opgaver

Firmaprofil (eksternt) – Nina, Kresten
Faglig dækning

Økonomiske nøgletal
Tidligere opgaver
Vi er/hedder…
Org. Struktur (overordnet)

Projektmæssigt niveau (eksternt) – Nikolaj, Kresten, Jonas
Status på igangv. Opgave
Tilknyttede pers.

Personlig profil (intern) – Nina, Jonas, Kresten
Se anden tavle
Faglig
Personlig

Administrativt modul (internt) – Nikolaj, Lars, Jonas, Kresten
Sammensætter grupper/par
Planlægning/ressourcer
Jobbeskrivelser
Overordnet statusniveau
(Kvalitetssikring)
Organisatorisk struktur (inkl. roller)

Projektmæssigt niveau (internt) – Nikolaj, Lars, Kresten
Personer
Opgaver
Dynamisk/status
Statisk/beslutningsgrundlag

Idegrundlag (internt) – Nina, Lars, Kresten
Procedurer
Best practices
Formål

Erfaringsopsamling (internt) – Jonas, Nikolaj, Kresten
Tidl. Erfaringer/projekter
FAQ

Story-indhold

Formålet med området
Hvilke problemer skal løses herunder?

I et operationelt perspektiv

Story 1 – Firmaprofil

Firmaprofilen skal give eksterne personer relevante oplysninger om organisationen, både på et overordnet organisatorisk niveau og et eksplicit fagligt niveau. Desuden skal interessenter få en beskrivelse af organisationens overordnede formål.

Opgave 1.1 Faglig dækning

Virksomhedens faglige profil skal præsenteres. Denne profil skal henvises til den enkelte medarbejder.

Opgave 1.2 Organisatorisk struktur

Virksomhedens faglige opdeling skal illustreres og begrundes. Lige fra styregruppe, bestyrelse, ledelse til medarbejdere og studentermedhjælpere.

Opgave 1.3 Økonomiske nøgletal

Virksomhedens økonomiske nøgletal skal være tilgængelige, dvs. årlige omsætning antal medarbejdere. Intet andet.

Opgave 1.4 Tidligere opgaver

Tidligere projekter, både interne og eksterne skal beskrives med mindre det er forretningsfortroligt. Det gælder både udviklingsopgaver og konsulentbistand.

Opgave 1.5 kort præsentation

Kort præsentation af virksomhedens formål, udviklingsopgaver, faglige områder, samarbejdspartnere mm.

Opgave 1.6 XP organisationen

Virksomhedens basering på XP. Beskrivelse af fordele ved XP. Omskriftelighed, effektivitet mm.

Story 3 Personlig profil

Denne story skal gøre det muligt at få et professionelt overblik over ansatte personer, så man som administrativ leder kan strukturere ressourcer, og så man som medarbejder kan få et overblik over kompetencer og ansvar gennem organisationen.

Opgave 3.1 Profil
Personlige oplysninger som navn, billede, alder (fødselsdag), ansættelsestidspunkt

Kompetencer, herunder konkrete fagområder, evner inden for disse, uddannelse, fritidsinteresser

Udførte opgaver i virksomheden

Private forpligtigelser der har indvirkning på planlægning, f.eks. familie, studie, fritidsinteresser

Jobbeskrivelse, indeholder de arbejdsopgaver medarbejderen er ansat til at opfylde.

Opgave 3.2 Administrativt databaseudtræk
Der skal oprettes en database, hvor man kan anvende nedenstående data til planlægning og administration af tid og gruppesammensætninger.

Private forpligtigelser

Kompetencer, herunder konkrete fagområder, evner inden for disse, uddannelse, fritidsinteresser

Udførte opgaver i virksomheden

Private forpligtigelser der har indvirkning på planlægning, f.eks. familie, studie, fritidsinteresser

Story 4 – administrativt modul

Formålet med det administrative modul er dels at skabe et øjebliksbillede over ressourcer, såvel allokerede som deallokerede, dels at givet mulighed for administration af disse i forhold til en langsigtet plan.

Hensigten med modulet er at skabe grundlag for fremtidige beslutninger for at opnå en udnyttelse af ressourcer, der både er realistisk og hensigtsmæssig i forhold til de overordnede mål.

Endvidere skal der være en beskrivelse/redegørelse af de styringsredskaber som virksomheden anvender, rent praktisk, for at opfylde det ovenfor beskrevne formål.

Opgaver

Overordnet status niveau

Begrebet "ressourcer" skal i denne forbindelse forstås i bredeste forstand og indeholder tid, kompetencer, lokaler, software og computere (flere…).

Story 5 – Projektmæssigt niveau

Den ansatte vil i starten af projekt have brug for at vide:

- hvilke opgaver der er knyttet til projektet, hvordan de er prioriteret,
- hvem der er sat til at lave de enkelte opgaver og fra og med hvornår
- hvilket tidsestimat der er på de enkelte opgaver.

Og på et praktisk niveau, hvilke klasser, metoder osv. der er aftalt ifm. de enkelte opgaver.
I løbet af projekt (efter morgenmøder) har den ansatte brug for følgende info:

- er der sket nogle ændringer af de ovennævnte vidensområder
- herunder om nogle opgaver er blevet delt op i mindre
- evt om der er opstået nogle helt nye opgaver
- er der sket ændringer i klasser/metoder
- viden om andre ansattes fremtidige mødetider

Viden relevant for kunden:
TPG-fasen:

- viden om de foreløbige tidsestimeringer som udviklingsafd. har inddelt tasks i indenfor hver iteration (samlet tidsestimat for hele projekt)
- antal mandetimer som udviklingsafd. mener der er rådighed indenfor iteration
- I forløbet
- status for opgaver indenfor iteration
- Overordnet status

Opgave 5.1 OpgaveTavler
Tre tavler med alle opgaver.
Opgavestatus repræsenteres vha. en farve: rød (ikke påbegyndt), gul (i gang), grøn (gennemtestet)
1. Postkasse tavlen: Yderligere opgaver er tilgængelige via opgave-tavle (røde9
2. Trafiklystavlen: Iterationstavle - Den indeholder de opgaver som kunden har prioriteret til den igangværende iteration. (rød, gul, grøn)
3. Agurketavlen: Afsluttede opgaver - indeholder færdige opgaver (grønne)
På de enkelte opgaver skal der fremgå:
Tidsestimat
fælles aftalte klasser/metoder
navne på par (når den er gul)
starttidspunkt (hvis den er gul)
Der kan desuden knyttes:
nye metoder
strege unødvendige

Opgave 5.2 Kalender
En tavle med en to ugers oversigt over de enkelte pars aftaler - og de opgaver de har aftalt.

Story 6: Forretningsgrundlag (internt)

Forretningsgrundlaget er et internt dokument der indeholder en beskrivelse af den filosofi, de principper, metoder som organisationen er opbygget efter og skal arbejde ud fra.

Forretningsgrundlaget foregår på et strategisk niveau og sætter rammen for en operationalisering i organisationen

Opgave 6.1 Idegrundlaget
Er en kort beskrivelse af virksomhedens overordnede filosofi, og har til formål at danne grundlag for den videre udarbejdelse af et koncept.

Opgave 6.2 Koncept (styreredskab)
Opgave 6.2.1 Ideologi
Præsentation af XP's ideologi og formål som ledende instanser (herunder XP's tolv principper)

Opgave 6.2.2 Organisatorisk struktur
Under dette afsnit i konceptet skal forefindes en visuel organisationsplan, hvor alle medarbejdere og deres indbyrdes placering i virksomheden fremgår.
Der skal også fremgå en ansvarsfordeling og en fordeling af beslutningskompetencer hos både afdelinger og medarbejdere.
Opgave 6.2.3 Procedurer
Der skal foreligge en plan for beslutningsprocedurer, f.eks. rækkefølge for processen og hvem der er involveret.
Der skal være en beskrivelse af arbejdspraksis for de enkelte medarbejdere, inklusiv en tidsplan.
Udarbejdelse af procedure for dokumentation
Opgave 6.2.4 Kommunikation
En beskrivelse af kommunikative strukturer (kanal, form, i henhold til PP)
Beskrivelse af en kommunikationsansvarligs funktion

Story 7: Erfaringsgrundlag

Gamle stories bruges til at sætte gang i ideerne i mødet ml. kunde og forretningsafd. Den story som kunde kommer frem til kan udviklings.afd. finde nogle kendetegn ved i en opgave database - derved kan de finde estimater fra lignende opgaver - muligvis finde refleksioner gemt fra tidl. projekter - om gode/dårlige oplevelser med koden. Ydermere kan de måske finde kode (7.3) der kan bruges
direkte -og dermed nedbringe estimater.
Manageren (7.4) kan lave en overall project- evaluering. Ved at se på tal for tidsestimater
kontra real tid, samt se på refaktorerereringsstatistikker, kan han sige noget om holdets svagheder i prog. fasen, samt om vurderingerne holder stik. (På person plan kunne han også holde øje med hvem der var langsom - intet fik programmeret, som ikke skulle refaktoreres - eller aldrig overholdt sine egne estimater - selvom de var længere end alle andres ...etc.)

Opgave 7.1 Stories
Vi forestiller os en database indeholdende alle stories. Der skal være sigende søgeord tilkoblet hver enkelt story.
Opgave 7.2 Opgaver
Databasen får opgaver registreret - med de reelle trackede tider for hver enkelt opgave.
Opgave 7.3 Kode
Kode stumper lægges i database [størrelsen heraf må afhænge af hvad Dennis finder en relevant størrelse - er det objekter, metoder ....] De registreres med dækkende betegnelser (muligvis def. i kode-standarden).
(En god javadoc kunne måske klare dette problem - her ville der være betegnelser på hver enkelt del, hvad den udførte etc - som man kunne søge på!! - teknisk problem? Hvordan henviser javadoc til koden - skal de være samme sted?)
Opgaver 7.4 Estimater vs. Real tid
Tracker indskriver reelt brugt tid på en opgave. Dette registreres med sigende titler (se 7.2) samt med link til tilhørende kode. Der vil desuden være mulighed for at finde diagrammer/statistikker op over hvor lang tid generaliserbare opgaver tager.
(Muligvis skal der bringes registrering af de enkelte programmører ind i sammenhæng hermed - så der er mulighed for at se hvem der arbejder e_ektivt sammen - og hvem der ikke gør... etc)
Opgaver 7.5 Refaktorering
Se refaktoreringsfaktor for de enkelte kode stykker. Kunne søge heri. Og ud fra tjek af kode, samt ændringer, da se hvor vanskelig den er at arbejde med.
Optimalt, så vil man kunne se hvor mange refaktoreringer der er sket pr. kode og med initialer på (så der vil være et billede af hvor ofte en persons kode bliver refaktoreret (og hvem der gør det..)).

Xp?
Enkeltpersoner
Økonomiærlighed

Bilag 4 – Samtale omkring XP

Denne transskription har udelukkende til formål at beskrive, hvad personerne siger, og ikke beskrive de pauser, tryk på ord eller andre sproglige områder. Transskriptionen er møntet på forståelse og mening af det sagte, som anvendes i analysen. Gruppen havde fået spørgsmålene på forhånd, så de havde fået tid til at forberede sig. Samtalen fandt sted den 7. maj.2004 på Dreamhouse.

Medvirkende fra unigruppen: Jonas, Nina og Lars

Kresten: Jeg vil egentlig godt starte ud med at høre, hvordan I har oplevet dette samarbejde, specielt de samarbejdsrelaterede roller. Har det været en problematisk afføre, given, spændende, oplysende, bare sådan lige fra hoften.

Jonas: Nu har vi jo talt lidt om det. Det vi blev enige om var, at det var fint; det var faktisk rigtig fint at få et indblik i tingene – også for vores XP kendskab. Det, der var rart var, at der var noget styrende, men måske skulle der have været mere styr på tingene, måske noget mere kontrol snakkede vi om. Vi havde faktisk for meget frihed, for vi havde meget frihed og et stort ansvar, men spørgsmålet var om vi kunne håndtere det.

Nina: Vi snakkede også om, at det var forskelligt som i forundersøgelsen var det måske fint nok, at der var plads til, at det var vores case der var inde over for eksempel. At vi havde den frihed var rart nok i stedet for at vi bare skulle hoppe efter din pibe, ikke. Men i XP workshoppen ville man godt føle, at nu gør vi det her, det her og så det her. PÅ den anden side var det også fint nok, at have det ansvar, at vi skulle skrive storys og det der. Det var fint at der var frihed til valg, men vi savnede det overblik over metoder, som beskrev, hvad vi skulle.

Jonas: Det kan måske forklares med, at vi kom og havde to dage og skulle nå det hele. Vi var ikke helt indstillede endnu.

Nina. Der var nogle principper vi bare ikke kendte til, og du havde sikkert ikke tid til at give os en dag med introduktion omkring eksempelvis Pair Programming mm.

Kresten: Det burde sådan set ikke være nødvendigt, at I skulle have en sådan introduktion. Der burde jeg kunne gå ind og skabe denne situation, sådan at den bare virker, men det var måske planlagt for laissez faire?

Jonas: Ja, hvor vi måske først får kontrollen et par dage senere.

Nina: Det du sagde på et tidspunkt, hvor du brød ind ved Lars og mig og så sagde du at vi skulle tage papiret fra hinanden eller bare skrive på det begge to. Det forstod vi selvfølgelig først da du sagde det; altså var det var. Det kunne man godt have startet med at introducere: Prøv det her, det her og det her. Nu har vi jo arbej-

det en del i grupper, så det at skrive kunne vi godt finde ud af, men det var alligevel noget andet at prøve fælles ejerskab.

Kresten: Det er en svær ting, og det er det alle siger: at Pair Programming er den sværeste disciplin indenfor XP og det var jo en anderledes vi anvendte Pair Programming på. Hvordan kunne man gøre det bedre? Ville det være at lave et introducerende forløb, eller?

Jonas: For os ville det var hjulpet, hvis vi havde snakket om hvad der helt præcist skulle ske, men kan man snakke om hvad der skal ske uden at gøre noget? Det ved jeg ikke helt.

Nina: Jeg synes godt man kunne have haft et introducerende forløb, der sagde: for at vi skal have et forløb bliver I nødt til at gøre det her, det her og det her, eller I bliver nødt til at efterleve de her metoder. Og ikke bare sige det, men også sige fordi I lave fælles ejerskab betyder det det her for systemet, for koden, eller noget andet. Et introducerende forløb, der forklarer det, så man sidder og tænker: ja, det må vi også prøve det der. Skabe det gode miljø.

Kresten: Hvordan oplevede I deling af viden i forbindelse med workhoppen? Primært i forbindelse med Pair Programming.

Jonas: Vi talte om at det var fint med begrebsafklaringen. At vi i starten have de indledende afklaringer på tavlerne. Det var faktisk en rigtig god måde at dele viden på; få det op på tavlen og snakke om det derudfra. Også til dem, der kom senere hen, at de kunne skabe overblik vha. tavlerne og tage udgangspunkt heri.

Nina: Selvom de så ikke helt forstod dem.

Jonas: De skulle lige følges igennem første gang. Så den videndeling var i hvert fald god.

Lars: Men det er også en arbejdsform man skulle lære mht. The Planing Game. Vi var sådan lidt da vi kom, at hvad skulle der ske, hvor er det vi kommer hen og sådan noget. Men det tror jeg vi mente ville komme nemmere når man blev rutineret, så det faldt naturligt, men sådan er det altid, når der er noget nyt man skal lære.

Kresten: Hvis man skulle prøve det igen ville det foregå på en anden måde? Hvis vi gjorde det igen ville det rykke et niveau op?

Lars: Ja.

Jonas: det kunne man godt forestille sig.

Nina: ja, det tror jeg også.

Kresten: Så må vi jo gøre det igen.

Jonas: Du skal bare presse på med principperne. Det var fint da du kom ind og sagde, at nu går I imod principperne og lige dreje det tilbage igen. Det var rart da du gjorde det.

Nina: Det virkede i hvert fald.

Kresten: Hvad med det ansvar I fik, for I fik jo en del ansvar for at kreere stories og tasks.

Jonas: Det er lidt med udgangspunkt i det vi talte om før, at vi måske havde for meget ansvar.

Nina: Både/og for på den anden side kunne jeg godt lide, at vi bare skulle sidde og skrive og man kunne skrive, men vi har også talt om forskellen på at udvikle stories og udvikle kode. Det kræver jo at man kan skrive, ikke? Det kræver at man har en måde at gøre tingene på, og nu da vi er 6. semester hum. Dat'ere så vi har en konsensus om måden at gøre tingene på, så jeg synes vi alle var kompetente nok til at kunne beskrive området, og gøre det godt. Men jeg kan ikke lige overskue om det ville være et for stort ansvar at lægge ud til alle. Man skal selvfølgelig bare have kompetencerne til det. Det er også anderledes med kode, for der er sikkert mere end en måde at gøre tingene på.

Kresten: Der er også flere måder at udvikle stories på. Selvom en story skal indeholde samme emner kan det jo godt udforme sig som to helt forskellige stories.

Nina: det er som om det er et meget bredere spektre at der kan ligge indenfor stories end inden for kode. Der er jo en del funktionalitetskrav, at det skal kunne det og det på en smart måde. Det ansvar der bliver lagt ud der er måske lettere at håndtere. Jeg tænker måske mere på dig end på os selv, om du kunne styre de kompetencer folk havde. Om du var sikker på det du fik.

Kresten: Det var jeg så selvfølgelig ikke. Det var jo primært for at afprøve værktøjerne og for at se, hvor lang kunne vi egentlig komme for at designe det videndelingssystem I ville designe. Og I kendte til XP, I kendte til videndeling og I havde en interesse for systemet. Det gør jeg kompetente til at kunne udtale jer om det. Så lige der så jeg ikke noget problem, og jeg har heller ikke set noget problem i de stories I har lavet. Jeg synes faktisk de er rigtigt gode og det er i hvert fald helt sikker at de kan anvendes til at udvikle et system. Det er jeg ikke i tvivl om.

Kresten: Hvordan ser i, at denne delegering af ansvar i forundersøgelsesfasen spiller sammen med XP? Nu røg i jo ud af huset for at lave denne forundersøgelse og sad ikke inde i huset.

Jonas: Jeg kan godt se ansvarsfordelingen som en stor del af XP, at ansvaret for den enkelte opgave bliver lagt over til udviklerne, men forundersøgelsen blev ikke betragtet som et XP projekt, i hvert fald ikke inde i mit hoved.

Kresten: Jeg så det som den del af CMM modellen som hedder Software Subcontract management, hvor I fik en opgave og den røg ud af huset så mit spørgsmål går også op hvor maget tilknytning I følte til virksomheden her i forhold til, at det var jeres egen forundersøgelsesfase, som skulle udføres.

Nina: I går talte vi om at vi mere havde tilknytning til dig som specialestuderende end som *m*2i3D fordi der er jo ingen struktur endnu. Du forsøger at være virksomhedsleder, men du er der ikke helt endnu. Du træffer ikke beslutninger for virksomhedens bedste. Det er ikke, at du sidder og siger: Nej det kan vi ikke fordi så vil det betyde det og det for hans arbejdstid eller økonomi. Det har så gjort for mit vedkommende at jeg mest har snakket med dig og du skulle skrive et speciale og at du så havde nogle visioner om en case. Selve *m*2i3D synes jeg ikke jeg har følt mig så meget tilknyttet til selvom vi har brugt ordet flittigt.

Jonas: Som vi talte om i spørgsmål 1 om vi følte det som en virksomhed eller som en person og det var nok hovedsageligt en person.

Kresten: Ændrede det sig igennem tiden, hvor vi gik mere over til projektsamarbejde?

Jonas: På en eller anden måde, så har vi en diskurs som siger, at nu har vi *m*2i3D i vores projekt, så vi får et eller andet forhold til det som en virksomhed nu. Men stadig så tænker jeg på Kresten, når jeg…. Men det er også svært at adskille inden i hovedet, når man tænker på virksomheden tænker man på Kresten.

Kresten: I forbindelse med workshoppen er der en rolle indenfor XP, hvor en person er kommunikationsdedikeret, dvs. personen sørger for at kommunikationen foregår på den rigtige måde, er konstruktiv og den fører til noget. Det var den rolle jeg forsøgte at gå ind og spille eller at tage på mig. Var det noget I oplevede under workshoppen?

Nina: Altså jeg oplevede det, da du brød ind og sagde, at I skulle forsøge at lave fælles ejerskab. Der gjorde du det og du sagde: snak lige med dem – bare spørg dem. Det husker jeg da i hvert fald du gjorde. Så ja, det husker jeg.

Kresten: Det er en essentiel person indenfor XP og jeg ved ikke om I har læst den artikel om det århusianske firma Systematic, som blev certificeret på CMM niveau 4, men de havde en beskrivelse af den kommunikationsdedikerede person. Det er ret sjovt at se teorien blive udfoldet i praksis.

Kresten: Sikring af kvalitet er et essentielt punkt i XP så jeg vil godt høre jeres input til hvordan i oplevede det dels i forundersøgelsesfasen, dels i workshoppen og dels under hele forløbet.

Jonas: Altså vi skrev at vi i vores forundersøgelsesfase savnede en styregruppe som sikrede kvalitet. Den blev aldrig rigtig helt etableret og igen så vi ikke forundersøgelsesfasen som et reelt XP projekt. Du var så i vores styregruppe dengang både i forundersøgelsen og i The Planning Game og sikrede hurtig respons og det kan man se som en kvalitetssikring. Vi fik verificeret vores ting og vores iagttagelser med det samme eller afverifi-

ceret. På den måde var der et hurtigt kontrolcheck. Det synes jeg fungerede godt. Hvis det er det du tænker på med sikring af kvalitet?

Kresten: Jo det er det.

Nina: Det under vorkshoppen forløb kun under to dage, og du læste vores stories bagefter, og jeg så en fare for at vi gik i fuldstændig modsatrettede retninger og som hurtigt kunne blive ukontrollerbart, og hvis man først tjekker op, når dagen er gået, så er det en hel dags arbejde der er spildt, hvis vi er gået i forkerte retninger fra starten af. Det så jeg så et problem i i forhold til stories, men hvis det er i forhold til parprogrammering vill det være noget andet. Men der har de i XP det her med tests af koden og det kan man ikke rigtig gøre med stories, så det kvalitetsmoment manglede.

Kresten: Men hjalp det i forhold til kvaliteten at vi havde den ene tavle fyldt op som et begrebsapparat over viden og så på den anden tavle de operationelle redskaber?

Nina: det gav i hvert fald et fælles udgangspunkt, som vi kunne vende tilbage til. Hvad var det nu lige vi skulle have med. I hvert fald som stikord, der skulle beskrives. Så måtte vi sætte os sammen og huske hvad vi havde snakket om under de forskellige ord.

Jonas: Jeg tror også det hjælper på kvaliteten, at der er noget eksternt; nogle ting man kan referere tilbage til alle sammen. Det her ligger under jeres område o.l.

Nina: Det er godt nok at det bliver skrevet ned, for ellers skulle man sidde og huske på det, og så kommer altid en masse fortolkninger ind over det.

Jonas: Både skrevet ned, men også synligt. Meget af det der bliver skrevet ned bliver aldrig brugt, men det kunne man jo bruge her.

Kresten: Kunne man bruge det i forbindelse med videndeling tænkte jeg på?

Jonas: Det tror jeg helt sikkert.

Nina: jeg synes helt sikker, at denne idé med at alle sidder i samme rum og der er denne fælles opslagstavle er en fantastisk god idé. Vi har faktisk selv indført en tavle i grupperummet; eller Jonas har gjort det og skabt overblik over afsnittene, hvem der laver hvad og sådan noget. Men gule grønne og røde tegnestifter. Man kunne godt gøre dette overfor begreberne og hvad der skulle være i og vi kunne godt gøre mere ud af det, ser jeg det som. At få systemdefinitionen op. Det er det her vi skal stræbe imod.

Kresten: Har I oplevet at processer er blevet ændret i The Planing Game eller er processerne blevet fulgt slavisk?

Jonas: Det var jo en lidt anden form for The Planning Game vi lavede med det indblik havde i Kent Becks bog, i forhold til at sidde og programmere, og der er bare nogen ting der ikke kører ind over, når man bare sidder og skriver.

Nina: Det var jo også en workshop personligt mødtet på os et eller andet sted, hvor der var nogle områder vi bare ikke kunne lave i The Planning Game. Så på den måde har det ikke været slavisk af naturlige årsager.

Kresten: Har I oplevet at det har været problematisk?

Jonas: Jeg har i hvert fald ikke oplevet det sådan at hov nu følger vi ikke metoden. Hvis jeg havde et bedre kendskab til metoden, så kan det godt havde været, at det var et problem.

Nina: Mit grundlag for at sige det er at jeg ikke kender metoden så godt på den måde, så jeg vil ikke kunne vide, hvor der mangler et eller andet helt vildt vigtigt. Jeg synes det var et fint forløb.

Kresten: Hvordan oplevede i min rolle i The Planning Game?

Jonas: Flink var et af de ord der blev nævnt. Igen vendte vi tilbage til, at vi måske gerne ville hav noget mere styring. For at er lidt vage om om vi vil have mere styring. PÅ en eller anden måde kan vi jo godt li, at du kommer med nogle svar, når vi bliver usikre. Måske var vi også for dårlige til at spørge. Det var i hvert fald et godt forhold der kom frem fra din rolle. Måske skulle vi have snakket noget mere om din rolle; om vi skulle have gjort din rolle mere klar fra starten.

Nina: Du havde mange ting ind over dig. Du er Kresten og du er 10. semester og du er *m*2i3D og du er også lige kommunikationsdedikeret og coach på samme tid og formidler af metoden. Du var mange ting på én gang. Det er ikke så man sidder og er forvirret i sit hoved, men jeg ved ikke rigtig hvordan jeg kunne følge, at jeg gerne ville have en klarer definition af det. Jeg ved ikke om det ville have gjort den store forskel.

Kresten: XP lægger jo også op til, at man delegerer ansvar og siger at det her område skal løses og jeg stoler simpelthen på, at den måde I gør det på er den bedste fordi I er de mest kompetente til at gøre det. Men det er måske også umiddelbart for meget?

Jonas: Det kan være at vi ikke helt følte at det var den kompetence vi havde.

Nina: Det tror jeg også er fordi vi ikke er uddannet til det her. Vi er det både og. Du har jo ikke sagt at I er programmører og I skal lave det her stykke kode. Vi skulle lave stories men det er ikke det vi laver til hverdag, selvom vi godt kunne finde ud af det, så er det ikke vores kompetenceområde.

Kresten: Ikke det at lave stories, men det at lave et videndelingssystem er jeres kompetence bl.a.

Nina og Jonas: Ja

Kresten: Lagde I mærke til det skift der skete under workshoppen? I starten da vi udfyldte den store tavle var I meget tavse og tilbagelænede og forventede at få noget, lidt ligesom til en forelæsning. Da vi begyndte på den anden tavle så skete der et skift hvor I begyndte at diskutere indbyrdes og med mig og I blev lidt mere frigjorte.

Jonas: Det var måske en opvarmningsfase for at finde ud af hvad der skulle ske

Nina: Jeg kan godt se det, når du siger det, men vi sad nok og forventede et introduktionsforløb e.l.

Kresten: Har I oplevet, at jeres evner er blevet øget ved anvendelsen af XP eller i Pair Programming i The Planning Game?

Jonas: Selvfølgelig så vi nye muligheder i den måde at samarbejde på, men ellers er det svært at sige at vi har oplevet et kvantespring i vores viden.

Kresten: Lærte i overhovedet af hinanden ved at arbejde på denne måde eller var det bare almindeligt gruppearbejde?

Nina: nej, der var en forskel til almindeligt gruppearbejde idet man i højere grad tvinger sig selv til at spørge de andre. Man kan godt føle ude på universitetet, at det gør vi ikke helt alligevel fordi jeg allerede har spurgt 50 gange om noget i dag, så nu holder jeg lige bøtte i 2 sekunder. Det skulle man bare og sådan var omgangstonen bare og på den måde var det anderledes at skulle skrive på det samme. Som Jonas siger så var det da et læringsforløb, hvor vi har fået et eller andet med, men om evnerne er øget er svær at svare på.

Jonas: Der er da både for og imod at man sidder to og skriver. Nogle gange når man sidder to kan det virkelig blive godt, og det snes jeg også dette forløb viser. Man får gennemtænkt nogle tanker og ideer og man får virkelig produceret noget. Men det er svært med sådan noget tekst i forhold til kode.

Nina: tekst har den personlige fortolkning af ord og det er svært at gå ind og rette: Nej det er det ord du mener. Man kunne hurtigt komme i clinch med nogle ord. Man skal kende hinanden bedre eller have større respekt for hinanden for at man kan skrive sammen.

Kresten: Dette spørgsmål rejser sig i forbindelse med CMM modellen fordi der er et krav som hedder Training Program (TP) og jeg mener Pair Programming indeholder en forøgelse af viden, men det er ikke noget I oplever?

Jonas: Det er svært på de to dage at sige, at man har fået så og så meget viden. Det ville være lettere over et længere forløb. Jeg tror helt sikker, at der ville ske meget der. I løbet af diskussionerne ville man da lære meget.

Nina: det tror jeg også. Jeg tror man kunne komme rimelig langt med at arbejde på den måde. Man ville så bare blive bedre til det. Men måske ikke med alt det andet. Jeg ved ikke om man kan bruge det i andre sammenhæng.

Kresten: Hvordan har I oplevet koordineringen mellem grupperne i The Planning Game. Kunne man anvende redskaber, så man fik større overblik over viden og de opgaver, som var i gang?

Jonas: Muligheden for at spørge synes vi var helt gode og at vi sad i samme rum var en dejlig måde at gøre det på.

Lars: Nogle gange kunne man godt have brug for at lukke døren. Der kunne godt være et sted til ro. Jeg har i hvert fald sværere ved at arbejde hvis der ikke er ro eller snak omkring mig.

Jonas: Men man blev også holdt i gang af dem, der sad og skrev. Der blev man motiveret.

Lars. Jeg er også af den mening at det gav en masse synergi. Man skulle så overveje om man skulle have muligheden for at træde uden for. Jeg ved ikke lige hvordan man kunne gøre det.

Nina: Mht. arbejdsopgaver havde vi jo ikke det store overblik over hvad de andre lavede udover de enkelte stikord på tavlen. Jeg kan ikke lige gennemskue noget redskab, der kunne gøre det bedre eller om det overhovedet ville blive bedre. Faren for overlap kunne jeg måske se, når man ikke har indsigt i det, de andre laver, men det må ligesom klares ved at lave overblik mellem elementerne i de enkelte opgaver. Det må være det, der sikrer, at der ikke kommer overlap mellem opgaverne. Man gad jo heller ikke sidde og læse de andres opgaver for at sikre at der ikke var overlap.

Kresten: Hvis vi nu prøvede at sætte et redskab på, hvor I bliver målt. Jeres evner som bliver til et tal, de stories I producerer bliver kvantificeret for at kunne måle på dem. Hvordan ville I have det med det hvis vi nu kørte den samme workshop igen med disse kvantitative redskaber?

Jonas: Og som så blev brugt til at parre os?

Kresten: Jo både til at parre jer og vurdere jeres evner

Jonas: Når man siger vurdere og måle, så er der noget negativt over det. Hvad er det du vil måle og hvordan vil du måle. Personligt ville jeg ikke have noget imod at der var noget som tjekkede op på mig. Men det kommer meget an på, hvordan det viser sig. Om jeg ser et barometer som viser at Jonas er på 95 i dag eller om jeg får et kvalitativt udsagn om at du skal arbejde med de og de ting. Men når der kommer det her kvantitative, så ved jeg ikke.

Kresten: Det kvantitative kan jo også være, at nu får I 1½ timer til at lave den her story og der skal være minimum 12 tasks. Det er jo meget estimering og ressourceforbrug kvantitative redskaber bruges til.

Nina: Jeg synes det er vildt svært at svare på for på den anden side er vi jo også vant til at få karakterer, så hvis man også blander det med noget kvantitativt, så får man jo også en vurdering. Det handler jo ikke om at man får 80 og er fyret, men det skal ikke være en stressfaktor, så man for alt i verden skal nå de 80, for ellers får jeg ikke den her opgave. Hvis der er en differentiering, hvor det ikke er alt for generelt, kan jeg ikke gennemskue om jeg så vil være tilfreds med det tal. Hvis der ikke bliver lagt for meget i det og det ikke er alt for stressende. Det kan jo være svært at sige at dig og dig klarede opgaven godt, så I skal være sammen næste gang. Jeg kan godt se ideen i det kvantitative til at kunne parre. Man skal prøve det for at kunne vurdere det. Kvalitetsbedømmelser kan både være sigende og knap så sigende.

Jonas: Det kan meget let vippe til noget man egentlig ikke har lyst til at høre. Nogle tal man ikke bryder sig om at se. Det kommer meget an på, hvordan de bliver brugt.

Kresten: Men med den erfaring I har fået igennem vores samarbejde, hvordan ser I så, at XP kan anvendes som organisatorisk strukturelt værktøj?

Jonas: Med den organisation, som vi har set eller i hvert fald snakket om, så kunne vi sagtens se det fungere. Altså at XP kørte på det her niveau. Vi har svært ved at se det større end de her 10-12 personer eller hvor mange det nu er.

Nina: Vi snakkede lidt om at se *m*2i3D som en udviklingsgruppe, men har svært ved at se det som flere udviklingsgrupper hvor der var flere udviklingsprojekter i gang og hvor der også var en ledelse. Så bliver det svært at vurdere om XP vil fungere på det, for skal alle udviklingsgrupper så have indsigt i hinanden og skal de så også sidde i samme rum og snakke og det kan godt blive lidt ekstremt at se det på den måde. Der må være nogle fravalg når man trækker det op på den måde.

Lars: det skægge ved XP er at den giver meget frihed og skal det fungere får du også en masse formalia som skal være på plads inden det kan fungere. Jeg tror ikke det er alle mennesker der kan arbejde på denne måde og det er ikke alle man kan bruge på den måde, og det skal man være bevidst om, når man ansætter folk. Der er nogen som skal prikkes til hver dag for ellers ender det i hat og briller. De skal have den der stramme styring – meget klart kommunikeret ud. Det er så kun i den indledende proces at der skal være styring og så kan man ellers begynde at bløde op og anvende XP's værdier derefter. Det handler om at få udstukket rammerne. Hvis man gør det kan jeg sagtens se det, men jeg tror der er et øvre loft på den i antallet af mennesker.

Jonas: Det handler om, hvordan man vælger at implementere det. Jeg tror der er nogen gode træk i XP som skal føres videre. Det er svært at se anvendelsen på et højere organisatorisk niveau.

Indeks