

Procesforbedring ved hjælp af selvforbedrende projektgrupper

Joakim Recht
Rasmus Walbum Jensen
Michael Lerskov Munk Nielsen

Institut for datalogi - Aalborg Universitet

Aalborg, d. 14. juni 2002



Aalborg, d. 14. juni 2002

Projekttitle:

Procesforbedring ved hjælp af
selvforbedrende projektgrupper

Fagområde:

Informationssystemer

Projektperiode:

4. februar 2002 -
14. juni 2002

Projektgruppe:

Joakim Recht
Rasmus Walbum Jensen
Michael Lerskov Munk Nielsen

Vejleder:

Ivan Aaen

Censor:

Kirsten Hjerrild Nielsen

Oplag:

8 eksemplarer

Antal sider:

128 sider

Afsluttet:

Aalborg, d. 14. juni 2002

Synopsis

I dette speciale undersøges muligheden for at forbedre en softwareorganisations udviklingsproces gennem en bottom-up tilgang til procesforbedring. Vi undersøger den traditionelle tilgang til procesforbedring og dens anvendte metoder med henblik på at identificere mangler. På baggrund af dette, samt kendte alternative tilgange til procesforbedring, udvikler vi en metode til at igangsætte og støtte op om selvforbedrende projektgrupper i en organisation. Metoden anvendes og valideres i en case-organisation, hvor vi gennem en intervention hjælper en projektgruppe til at udføre bottom-up procesforbedring. Selvforbedrende projektgrupper er baseret på ideen om, at ansvaret for procesforbedringen skal flyttes til organisationens projektgrupper, og at projektgrupperne skal have lov til at tilpasse deres udviklingsproces efter behov. Interventionen viser, at det er muligt at opnå dette, men at der opstår nye problemer. Selvom små forbedringer kan gennemføres hurtigt, formoder vi, at der vil være problemer med forbedringernes forankring og anvendelse i det daglige arbejde. Et andet problem kan være, at den overordnede forbedringsindsats ikke er koordineret, hvilket reducerer muligheden for at opnå en højere procesmodenhed. På trods af disse problemer mener vi, at specialet viser, at det er muligt at udføre bottom-up procesforbedring på en struktureret måde.



Aalborg, June 14th, 2002

Project title:

Software Process Improvement using Self-improving Teams

Line:

Information Systems

Project period:

February 4th, 2002 -
June 14th, 2002

Project group:

Joakim Recht
Rasmus Walbum Jensen
Michael Lerskov Munk Nielsen

Supervisor:

Ivan Aaen

External examiner:

Kirsten Hjerrild Nielsen

Number of prints:

8 copies

Number of pages:

128 pages

Finished:

Aalborg, June 14th, 2002

Synopsis

This master thesis examines the possibility of improving a software organisation's development processes using a bottom-up approach. We examine traditional process improvement methods and identify deficiencies in these. Based on this as well as known process improvement alternatives, we develop a method for implementing and supporting self-improving teams in an organisation. This method is applied and validated in a case organisation, where we through an intervention helps a team performing bottom-up process improvement.

Self-improving teams are based on the idea that responsibility should be delegated to the project teams, and that the teams themselves should be allowed to adjust their development processes as needed. The intervention shows this is possible to attain although new problems do arise. Even though small improvements can be implemented quickly, we suspect there will be problems with anchoring the changes and maintaining them in the daily practices. Another problem could be that the overall improvement effort is not coordinated, which reduces the possibilities of reaching a high process maturity. Despite these problems, we believe that this thesis shows it is possible to improve processes in a structured bottom-up manner.

Forord

“ Change is one thing, progress is another. Change is scientific, progress is ethical. Change is indubitable, whereas progress is a matter of controversy.” – Bertrand Russell

Dette projekt er blevet til på Institut for datalogi ved Aalborg Universitet i foråret 2002 på baggrund af en grundlæggende interesse i procesforbedring. Vores hensigt har været at undersøge, hvordan traditionel procesforbedring foregår i en konkret virksomhed, og på baggrund af det komme med anbefalinger til en forbedret fremgangsmåde.

Rapporten er skrevet på dansk, og fagudtryk er blevet oversat, hvor det har været muligt. Desværre kan nogle engelske ord miste deres fulde betydning ved en sådan oversættelse.

Vi vil gerne rette en stor tak til Logimatic for deres samarbejde i forbindelse med dette speciale. Specielt vil vi gerne takke Finn Hedehus, Dorte Pedersen, Annette W. Sørensen, Klaus S. Nielsen, Michael Gottlieb, Lars Børglum og Dennis Andersen for deres medvirken.

Aalborg, d. 14. juni 2002

Joakim Recht

Rasmus Walbum Jensen

Michael Lerskov Munk Nielsen

Indhold

1	Procesforbedring i Logimatic	1
1.1	Problemstilling	3
1.2	Selvforbedrende projektgrupper	3
1.3	Selvforbedring i praksis	4
1.4	SPI gruppens rolle	5
1.5	Problemdefinition	5
2	Traditional SPI	9
2.1	Baggrunden for SPI	9
2.2	Capability Maturity Model	10
2.3	IDEAL modellen	13
2.4	Læring og læreprocesser	15
2.5	Vurdering af traditionel SPI	21
3	Alternativ SPI	25
3.1	Bachs procesforbedringsmodel	25
3.2	End-user SPI	27
3.3	Personal Software Process	30
3.4	Team Software Process	32
3.5	Letvægtsmetoder	34
3.6	Vurdering af alternativ SPI	37
4	Selvforbedrende projektgrupper	39
4.1	Egenskaber ved selvforbedrende projektgrupper	39
4.2	Overvejelser om indførelse	42
4.3	En alternativ forbedringsmodel	45
4.4	Opsummering	48

5	Case beskrivelse	49
5.1	Logimatic Gruppen	49
5.2	Logimatic Software A/S	51
6	Logimatics SPI tilgang	63
6.1	SPI gruppen	64
6.2	Vurdering af SPI indsatsen	65
7	Interventionen - forberedelse	67
7.1	Forløbet hos Logimatic	67
7.2	Reviews	69
8	Interventionen - fase 1	73
8.1	Første review	73
8.2	Vurdering af første review	75
8.3	Opfølgning på forløbet	76
8.4	Vurdering af opfølgningen på første review	79
8.5	Opsummering på interventionen fase 1	80
9	Interventionen - fase 2	83
9.1	Andet review	83
9.2	Vurdering af andet review	84
9.3	Opfølgning på forløbet	85
9.4	Vurdering af opfølgningen på andet review	88
9.5	Opsummering på interventionen fase 2	89
9.6	Opsamling på hele interventionen	90
10	Review gruppen	93
10.1	Reviewgruppen	93
10.2	Vurdering af reviewgruppen	94

11 Analyse	97
11.1 Procesejerskab og problemerkendelse	97
11.2 Læring	98
11.3 Dokumentation og design	99
11.4 Planlægning	100
11.5 Normer	100
11.6 Processen	101
11.7 Fokus	102
11.8 SPI gruppens rolle	103
11.9 Faldgruber for selvforbedrende projektgrupper	106
12 Konklusion	109
12.1 Hvilke egenskaber karakteriserer selvforbedrende projektgrupper?	109
12.2 På hvilken måde kan en softwareorganisation fremme selvforbedrende projektgrupper?	110
12.3 Hvordan bør en SPI gruppe fungere i en softwareorganisation med selvforbedrende projektgrupper?	111
Litteraturliste	113
Figurliste	117
Tabelliste	119

Dette projekt tager udgangspunkt i Logimatic Software A/S¹, som igennem længere tid har haft fokus på SPI (Software Process Improvement). SPI indsatsen i Logimatic er dog løbet ind i problemer i kraft af, at indsatsen er begyndt at miste fokus i organisationen. Derfor vil vi gennem dette projekt beskæftige os med et alternativ til Logimatics nuværende indsats. Den nuværende indsats kommer primært til udtryk i virksomhedens ISO-9001:94 certificerede kvalitetssystem og virksomhedens nedsatte SPI gruppe.

I 1998 blev Logimatic certificeret i henhold til ISO-9001:94 standarden efter ønske om at forbedre kvaliteten af virksomhedens software og for at leve op til konkurrencen indenfor branchen. Baggrunden for denne certificering var indførelsen af et kvalitetssystem i virksomheden, hvorved virksomhedens forretningsgange, fx proceduren for udarbejdelse af projektplaner, blev beskrevet og dokumenteret. Ydermere medførte certificeringen, at der blev opstillet formelle krav til udvikling, implementering og servicering af software. Denne certificering har ifølge Logimatic betydet et løft i virksomhedens kvalitetssikring. [APH01]

På baggrund af et faldende fokus på kvalitetssikring og procesforbedring fra ledelsen og medarbejdernes side tog en projektgruppe bestående af tre medarbejdere i år 2000 fat på procesforbedring i Logimatic. Denne indsats tog udgangspunkt i projekthens deltagelse i et kursus i projektleddelse ved Aalborg Universitet. Resultatet af projekthens arbejde var, at der blev nedsat en SPI gruppe i virksomheden, hvis formål er at modne Logimatic i forhold til Capability Maturity Model² (CMM) normen [PCCW93]. Konkret skal SPI gruppen stå for at bringe Logimatic fra et CMM niveau mellem 1 og 2 i år 2001 til CMM niveau 3 indenfor 3 år. SPI gruppen, som består af 3-4 medarbejdere udpeget af ledelsen, har ansvaret for CMM målinger, anbefaling af tiltag, udarbejdelse af handlingsplaner og sikring af handlingsplanernes realisering. Dette sker bl.a. ved nedsættelse af forbedringsgrupper, der kan bestå af medarbejdere på tværs af projekthens grupperne, som får tildelt ansvaret for indførelse af et konkret tiltag. I forbindelse med projekthens arbejde er der blevet påbegyndt et konkret initiativ i form af kvalitetsmøder. Formålet med disse møder er at sikre kvaliteten af Logimatics produkter gennem formelle reviews. [APH01]

Procesforbedringsindsatsen i Logimatic har indtil nu været kendetegnet ved at være en *top-down* tilgang, som har taget udgangspunkt i ISO standarden og CMM normen. Ved *top-down* procesforbedring er det organisationens top, som igangsætter SPI initiativer. Fx skete indførelsen af kvalitetssystemet primært på baggrund af et ønske fra ledelsens side om at kunne leve op til konkurrencen indenfor branchen og derved sikre Logimatics fremtidige overlevelse. Forbedringsindsatsen i forhold til CMM normen er også præget af at være en *top-down* tilgang, fx har det indtil nu været

¹ I kapitel 5 vil vi præsentere og foretage en analyse af Logimatic, som vi har brugt som case i projektet. Derefter vil vi efterfølgende i kapitel 6 beskrive og vurdere Logimatics SPI indsats.

² Modellen vil blive nærmere gennemgået i afsnit 2.2.

SPI gruppens ansvar at træffe beslutninger om hvilke initiativer, der skal igangsættes, og hvor de skal igangsættes i organisationen.

Gennem vores tidligere samarbejde med Logimatic [RJN01] har vi identificeret en række problemer ved en top-down tilgang til procesforbedring i virksomheden. I forhold til kvalitetssystemet erfarede vi gennem en række interviews, at kvalitetssystemet indenfor nogle områder kun i ringe grad benyttes i virksomheden. Dette skyldes bl.a., at nogle af standarderne ikke afspejler den faktiske udvikling i virksomheden, fx er en række standarder forældede, og andre standarder passer ikke til alle afdelingers arbejdsprocesser, da der blev taget udgangspunkt i en enkelt afdeling ved udarbejdelsen. Desuden er standarderne i kvalitetssystemet udarbejdet af enkelte medarbejdere i virksomheden og derefter forsøgt udrullet. Denne udrulning er udpræget top-down og herved risikeres fremmedgørelse overfor kvalitetssystemet, da systemet ikke afspejler, hvordan udvikling foregår i praksis i organisationen.

Med hensyn til Logimatics forsøg på at få etableret formelle reviews i form af kvalitetsmøder, har de ikke haft stor succes i virksomheden. Kvalitetsmøderne er endnu ikke nået længere end til et par pilotprojekter, som der dog har være positiv respons på. En af grundene til den manglende udvikling er, at det er SPI gruppen, som har været drivkraften bag kvalitetsmøderne. Da SPI gruppen består af ledende medarbejdere, har der ikke været plads til at afsætte tilstrækkeligt med ressourcer, og derved er initiativet gået i sig selv igen.

Denne udsultning af SPI indsatsen har været nævnt i litteraturen som en af grundene til, at normbaserede SPI indsatser fejler i nogle organisationer [HZG⁺97]. Andre grunde til fiasko, som har været nævnt, er territorialisering, kynisme pga. tidligere mislykkedes SPI forsøg og overbevisning om, at SPI kommer i vejen for det rigtige arbejde [HZG⁺97]. Fremmedgørelse overfor SPI indsatsen kan også være en væsentlig faktor for fiasko, hvor medarbejderne opfatter indsatsen som ledelsens forsøg på at gribe ind i deres personlig frihed [DM93], eller hvor medarbejderne ikke opfatter de problemer, som indsatsen tager fat i, som relevante [Bac98].

Om Logimatics nuværende problemer skyldes ovenstående faktorer er svært at svare på. Vi mener, at Logimatics problemer kan bunde i de grundlæggende antagelser, der ligger bag en top-down tilgang til procesforbedring, som fx normbaseret SPI. Dette er antagelser om, at initiativet til SPI skal komme fra toppen af i organisationen, at toppen ved hvilke problemer, der er i organisationens udviklingsproces og at adfærden hos medarbejderne kan ændres gennem design af nye processer. Vi vurderer derfor, at en anden tilgang til procesforbedring i virksomheden bør overvejes, da der med den nuværende tilgang ikke er tilstrækkelig fokus på forbedringsindsatsen.

Et alternativ kunne være at fokusere på en *bottom-up* tilgang også kaldet *End-user SPI* [Aae02]. Ved End-user SPI benyttes tilgange til procesforbedring, som sigter imod at få etableret softwareprocesser på gruppe- eller individniveau. Fokus i tilgangene ligger på procesforbedring gennem inddragelse af medarbejderne, og ikke på definerings og formalisering af arbejdsprocesser som det kendes fra traditionel SPI. Desuden adskiller End-user SPI sig fra traditionel SPI ved, at fokus er på at forbedre dele af organisationen først, og derefter lade disse forbedringer sprede sig til resten af organisationen lige så stille.

1.1 Problemstilling

Den primære problemstilling i dette projekt er, hvordan en bottom-up tilgang til procesforbedring kan anvendes i praksis. Der har været flere bud på, hvordan bottom-up tilgange kan opnås, og hvilke fordele der er ved dem [Bac95] [Bac98] [Jak98], men endnu er der ikke etableret nogen metode for, hvordan en bottom-up tilgang kan tilstræbes i praksis. I stedet bygger de fleste forslag på ideer, der ikke har været efterprøvet i praksis, eller de bygger på erfaringer, der ikke er bakket op teoretisk. Et af hovedmålene i projektet er derfor at forsøge at etablere en række tiltag, en organisation kan tage for at fremme bottom-up procesforbedring, der både er baseret på praktiske erfaringer og teoretiske overvejelser.

Vi har valgt at basere bottom-up tilgangen på *selvforbedrende projektgrupper*. Selvforbedrende projektgrupper tager udgangspunkt i de projektgrupper, der allerede eksisterer i organisationen, og disse grupper har selv ansvaret for at procesforbedre i modsætning til den traditionelle tilgang, hvor en SPI gruppe sidder med ansvaret. Vores tidligere arbejde i Logimatic [RJN01] har vist, at projektgrupperne har en interesse i at forbedre sig, både af hensyn til personlige interesser og ambitioner, og på grund af et ønske om at forbedre virksomheden som helhed. Dette gør, at vi ser selvforbedrende projektgrupper som en reel mulighed i Logimatic.

En af de fordele, vi forudser ved selvforbedrende projektgrupper er, at SPI indsatsen bliver spredt ud over organisationen, hvilket betyder, at indsatsen ikke længere er afhængig af en enkelt gruppes arbejde. Det kan hjælpe i den situation, som Logimatic er i for tiden, hvor SPI indsatsen er blevet udsultet på grund af manglende ressourcer i SPI gruppen.

En anden konsekvens vi forudser ved at lægge ansvaret for procesforbedringen ud i de enkelte projektgrupper er, at projektgruppen og dens medlemmer føler sig mere motiverede til at foretage forbedringerne, da forbedringerne vil tage udgangspunkt i gruppens eget arbejde og egen hverdag. Det betyder, at forbedringerne har større chance for at blive fastholdt i gruppens videre arbejde, og at gruppen dermed får et større udbytte af forbedringerne, end den ville have fået ved en traditionel SPI tilgang.

I dette projekt vil vi opstille en metode for, hvordan en selvforbedrende projektgruppe kan sættes i gang og komme med forslag til tiltag, en organisation kan tage for at fremme selvforbedrende projektgrupper. Disse vil blive opstillet på baggrund af en intervention i Logimatic, hvor vi afprøver nogle af vores ideer baseret på teoretiske overvejelser om, hvilke egenskaber selvforbedrende projektgrupper bør have. Dette leder os frem til vores undersøgelsesspørgsmål:

Hvordan kan en softwareorganisation foretage procesforbedring ved hjælp af selvforbedrende projektgrupper?

Undersøgelsesspørgsmålet kan deles op i nogle mindre delspørgsmål, der konkretiserer hovedspørgsmålet.

1.2 Selvforbedrende projektgrupper

Det første underspørgsmål beskæftiger sig med de teoretiske aspekter af selvforbedrende projektgrupper, hvor vi igennem en analyse af egenskaberne for traditionel og alternativ SPI opstiller

modsætninger for derigennem at komme frem til en karakteristik af selvforbedrende projektgrupper. Det leder os frem til vores første undersøgelsesspørgsmål:

Hvilke egenskaber kendetegner selvforbedrende projektgrupper?

Karakteristikken af hvilke egenskaber der kendetegner selvforbedrende projektgrupper, er nødvendig for, at der kan blive etableret en fælles forståelse af, hvad begrebet indebærer. Det er også nødvendigt for at identificere de områder, hvor det kan være nødvendigt at gribe ind i forhold til de projektgrupper, der eksisterer i Logimatic. Hvis selvforbedrende projektgrupper skal blive en realitet, kan det være nødvendigt at ændre ledelsens opfattelse af, hvordan traditionel SPI foregår.

Det drejer sig om, at vi ønsker at undersøge, hvilke egenskaber selvforbedrende projektgrupper bør have. Karakteristiken af selvforbedrende projektgrupper vil være baseret på en teoretisk tilgang af traditionel og alternativ SPI, men også på vores egen opfattelse af, hvordan en projektgruppe kan organiseres, hvis den skal have medansvaret for at gennemføre forbedringer i organisationen.

Vi har en klar forventning om, at der er nogle egenskaber, der er væsentligt anderledes end for normale projektgrupper. Det handler især om evnen til at reflektere over egen proces, som er en nødvendig egenskab i selvforbedrende grupper, og som der ikke lægges nær så meget vægt på i normale projektgrupper. Derudover drejer det sig om at flytte ansvaret for forbedringerne fra SPI gruppen ud i projektgrupperne.

1.3 Selvforbedring i praksis

Det andet spørgsmål beskæftiger sig med de praktiske aspekter af selvforbedrende projektgrupper. Med udgangspunkt i de egenskaber der karakteriserer en selvforbedrende projektgrupper, vil vi opstille en forbedringsmodel, som vi ønsker at afprøve i Logimatic, hvilket leder os frem til vores andet undersøgelsesspørgsmål:

På hvilken måde kan en softwareorganisation fremme selvforbedrende projektgrupper?

Det sker gennem en intervention i Logimatic, hvor vi forsøger at påvirke en projektgruppe til at være selvforbedrende. Vi opstiller en metode for, hvordan selvforbedring kan opnås. Metoden opstilles på baggrund af de egenskaber, selvforbedrende projektgrupper har og på baggrund af en analyse af de modsætninger, der ligger i forholdet mellem en traditionel og en alternativ SPI tilgang. På baggrund af analysen kan vi identificere hvilke tiltag, softwareorganisationer bør tage, når de ønsker at fremme selvforbedrende projektgrupper.

For at foretage en validering af vores metode vil vi derfor afprøve modellen i praksis i Logimatic, hvorefter vi laver en opsamling på resultatet for at vurdere, hvilke perspektiver der ligger i selvforbedrende projektgrupper.

1.4 SPI gruppens rolle

Det sidste underspørgsmål omhandler SPI gruppen og dens forhold til selvforbedrende projektgrupper. Her vil vi, på baggrund af erfaringerne fra interventionen, karakterisere SPI gruppens rolle og opgaver i softwareorganisationer med selvforbedrende projektgrupper. Det leder os frem til vores tredje undersøgelsesspørgsmål:

Hvordan bør en SPI gruppe fungere i en softwareorganisation med selvforbedrende projektgrupper?

SPI gruppen har traditionelt haft ansvaret for procesforbedringen i en organisation og har siddet tungt på det, der i litteraturen kaldes procesejerskabet. Det vil sige, at det har været SPI gruppens ansvar og opgave at overvåge og vedligeholde processerne. Dette er ikke længere muligt med selvforbedrende projektgrupper, da en del af ansvaret skubbes ned i de enkelte projektgrupper, og dermed flyttes procesejerskabet også. SPI gruppen bliver på den måde fritaget for nogle opgaver, men på den anden side, bliver det SPI gruppens ansvar at sørge for, at organisationen som helhed bevæger sig i den ønskede retning, og de skal i stedet for støtte op omkring de enkelte projektgrupper.

Derudover vil vi undersøge, hvordan SPI gruppen kan interagere med de selvforbedrende projektgrupper, da den normale SPI fremgangsmåde med SPI gruppen, som den styrende enhed ikke længere fungerer, når der er tale om selvforbedrende projektgrupper. SPI gruppen kan ikke diktere, hvordan processerne skal se ud, men skal i stedet for komme med råd og vejledning til, hvordan de enkelte grupper kan udarbejde deres egne processer. Dvs. vi ønsker at besvare hvilke nye opgaver, SPI gruppen får som resultat af selvforbedrende projektgrupper og hvilken indvirkning, dette har på SPI gruppens arbejde.

1.5 Problemdefinition

Igennem dette projekt ønsker vi at besvare følgende problemstilling:

Hvordan kan en softwareorganisation foretage procesforbedring ved hjælp af selvforbedrende projektgrupper?

- A) Hvilke egenskaber kendetegner selvforbedrende projektgrupper?
- B) På hvilken måde kan en softwareorganisation fremme selvforbedrende projektgrupper?
- C) Hvordan bør en SPI gruppe fungere i en softwareorganisation med selvforbedrende projektgrupper?

For at svare på problemstillingen vil vi teoretisk overveje, hvordan software organisationer kan procesforbedre ved hjælp af selvforbedrende projektgrupper. Herunder vil vi udvikle en model for indførelse af selvforbedrende projektgrupper, og denne model vil vi afprøve i praksis i en projektgruppe i Logimatic. Dette projekt, som dokumenterer disse teoretiske overvejelser og den praktiske anvendelse er inddelt i følgende dele:

1. *Teoretisk SPI*. Vurdering af teori omkring SPI og dannelse af syntes om selvforbedrende projektgrupper.
2. *Praktisk SPI*. Casestudie af Logimatic og anvendelse af syntes i praksis gennem intervention.
3. *Analyse og konklusion*. Med udgangspunkt i interventionen analyseres selvforbedrende projektgrupper og der konkluderes på problemstillingens undersøgelsesspørgsmål.

I det følgende vi de kapitler som projektet er inddelt i kort blive beskrevet.

Del 1: Teoretisk SPI

Kap. 2 - Traditionel SPI. Dette kapitel gennemgår, med udgangspunkt i CMM og IDEAL modellen, den traditionelle tilgang til SPI. Denne tilgang holdes desuden op imod læring, som også er beskrevet i kapitlet. Herved får læseren en forståelse af traditionel SPI, og de tanker der ligger bag.

Kap. 3 - Alternativ SPI. I dette kapitel beskrives og vurderes en række forskellige alternativ tilgange til SPI. Formålet er at præsentere nogle af de alternative tilgange, som er blevet foreslået og herigennem hente inspiration til selvforbedrende projektgrupper. Læseren vil efter kapitlet have en grundlæggende forståelse for de alternative tilgange og tankerne bag.

Kap. 4 - Selvførelsende projektgrupper. Præsenterer vores forslag til en alternativ SPI tilgang funderet i begrebet selvforbedrende projektgrupper. I kapitlet vil hovedideerne og tankegangene bag selvforbedrende projektgrupper blive præsenteret, og der vil blive præsenteret et forslag til en metode, hvorved selvforbedrende projektgrupper kan igangsættes i en organisation. Efter kapitlet vil læseren have en forståelse for begrebet selvforbedrende projektgrupper, og de tanker der ligger bag begrebet.

Del 2: Praktisk SPI

Kap. 5 - Case beskrivelse. Beskriver casevirksomheden Logimatics historie, forretningsområder og organisatoriske opbygning. Desuden vurderes Logimatic i forhold til Mintzbergs organisationsteori. Formålet er at give læseren en forståelse for den kontekst, hvori selvforbedrende projektgrupper skal afprøves.

Kap. 6 - Logimatics SPI tilgang. Giver en beskrivelse af den SPI tilgang, som Logimatic indtil nu har anvendt. Denne tilgang vil i kapitlet blive vurderet i forhold til den traditionelle tilgang. Formålet med kapitlet er, inden interventionen, at give læsere en forståelse for, hvordan Logimatic indtil nu har grebet SPI an.

Kap. 7 - Interventionen - forberedelse. Dette kapitel dokumenterer de indledende forberedelser til interventionen i Logimatic. I kapitlet vil selve interventionsforløbet, projektgruppen som vi skulle samarbejde med i forbindelse med interventionen og teorien omkring forbedringsområdet blive præsenteret. Efter kapitlet vil læseren have en overordnet forståelse for interventionen.

Kap. 8 - Interventionen - fase 1. Dokumenterer første fase af interventionen, hvor projektgruppen identificerer hvad, og hvordan den ønsker at forbedre sin praksis.

Kap. 9 - Interventionen - fase 2. Dokumenterer anden fase af interventionen, hvor projektgruppen foretager forbedringerne og efterfølgende reflekterer over forbedringsinitiativet. Desuden foretager projektgruppen en vurdering af selvforbedrende projektgrupper.

Kap. 10 - Reviewgruppen. Med udgangspunkt i vores observationer i forbindelse med interventionen bliver projektgruppen beskrevet og vurderet. Formålet er at give læseren en forståelse for, hvordan gruppen samarbejder.

Del 3: Analyse og konklusion

Kap. 11 - Analyse. I dette kapitel analyserer vi vores ideer omkring selvforbedrende projektgrupper ud fra vores erfaringer i forbindelse med interventionen i Logimatic. Herunder præsenteres en række faldgruber i forbindelse med indførelse af selvforbedrende projektgrupper. Formålet er at give en vurdering af, om selvforbedrende projektgrupper er en reel mulighed som alternativ forbedringsmetode.

Kap. 12 - Konklusion. I dette kapitel konkluderer vi på spørgsmålene, der er stillet op i problemdefinitionen. Formålet er at give en kort opsummering af vores svar på spørgsmålene.

I dette kapitel vil vi foretage en vurdering af den traditionelle måde at lave SPI på, som udmøntet igennem CMM og IDEAL modellen. Derudover vil vi beskrive to forskellige læringstilgange og holde den op imod den traditionelle SPI tankegang. Det vil være med til at begrunde vores holdning til traditionel SPI, og bruges til at identificere problemer ved denne tilgang til procesforbedring. Efter kapitlet vil læseren have en grundlæggende forståelse af den traditionelle måde at foretage procesforbedring på, og hvilke tanker der ligger bag fremgangsmåden.

2.1 Baggrunden for SPI

I midten af 1980 begyndte software industrien at opdage, at virksomhederne havde behov for at fokusere på, hvordan de på en mere effektiv måde kunne styre udviklingsprocessen. Baggrunden var, at den hidtidige måde at effektivisere på var foregået ved at indføre ny teknologi som den drivende faktor, men med moderat succes. Den erfaring medførte en ændret opfattelse af, at hvis ønsket var at forbedre produktiviteten og kvaliteten af produkterne, så var der behov for en bedre styring af udviklingsprocesserne [CKO92, s. 85]. Det betyder ikke, at teknologi ikke er vigtigt, men det betyder blot, at det først er nødvendigt at tage sig af dårligt definerede og styrede processer, og først derefter vil indførelse af ny teknologi have en positiv indvirkning [Zah98].

Netop på baggrund af ovenstående problemstilling begyndte Watts Humphrey og nogle af hans kollegaer at interessere sig for SPI og i løbet af anden halvdel af 1980erne, begyndte de at udvikle deres ideer, hvilket har udmøntet sig i CMM, som vi kender den i dag. Men for at forstå Humphreys ideer, bag traditionel SPI, er det vigtigt at have kendskab til, hvilke strømninger der eksisterede dengang i 80erne.

I starten af [Aae02] sættes Humphreys tanker og ideer bag CMM i et historisk perspektiv. Da Humphrey udviklede CMM, var forskningsmiljøet indenfor softwareudvikling koncentreret omkring, hvordan CASE værktøjer, til understøttelse af procesmodellering, kunne udvikles. Formålet med værktøjerne var, at de skulle understøtte specifikation, tilpasning og udvidelse af metoder og dermed underbygge en individuel tilpasset softwareproces på baggrund af foruddefinerede modeller.

Humphrey er blevet kraftig inspireret af antagelsen om, at en softwareproces kan betragtes på samme måde som et stykke software og er på linje med Osterweil, som i [Ost87] argumenterer for, at software processer har de samme karakteristika som software, og dermed også kræver de samme discipliner og metoder. Osterweil lægger op til, at procesbeskrivelser opbygges af programmoduler, som beskriver, hvordan den givne proces skal udføres. Dermed benyttes programmeringsteknikker til procesprogrammering, og frihedsgraden for den enkelte udvikler indsnævres kraftigt. Dette er baggrunden for Humphreys ideer og tanker omkring, hvordan software procesforbedring skal foregå. I næste afsnit vil vi kort opridse, hvordan CMM normen er opbygget.

2.2 Capability Maturity Model

CMM er baseret på Watts Humphreys arbejde [Hum90] med software processer, og er blevet udviklet af Software Engineering Institute (SEI), som gennem praktisk erfaring med procesforbedring i større offentlige og private amerikanske virksomheder har udviklet en generel model for modning af en software organisations udviklingsproces. Den nuværende version af CMM, version 1.1, er dokumenteret af SEI i de to tekniske rapporter [PCCW93] og [PWG⁺93].

Grundideen bag CMM er, at procesforbedring sker bedst i små evolutionære skridt fremfor revolutionerende innovationer. Modellen opstiller en ramme for disse skridt, som sikrer, at organisation fokuserer på nogle enkelte skridt ad gangen. Det er vigtigt at understrege, at CMM ikke i sig selv fortæller, hvad en organisation skal gøre for at øge dens modenhed, men kun opstiller en ramme for, hvordan organisationen kan modnes, samt en række kriterier til at vurdere, hvornår en organisation har en given modenhed ¹. Selve modellen er struktureret i 5 dele, kaldet modenhedsniveauer, og ved at bruge denne model som ramme for organisationens procesforbedring, sikres en effektiv etablering af procesforbedringsinitiativer.

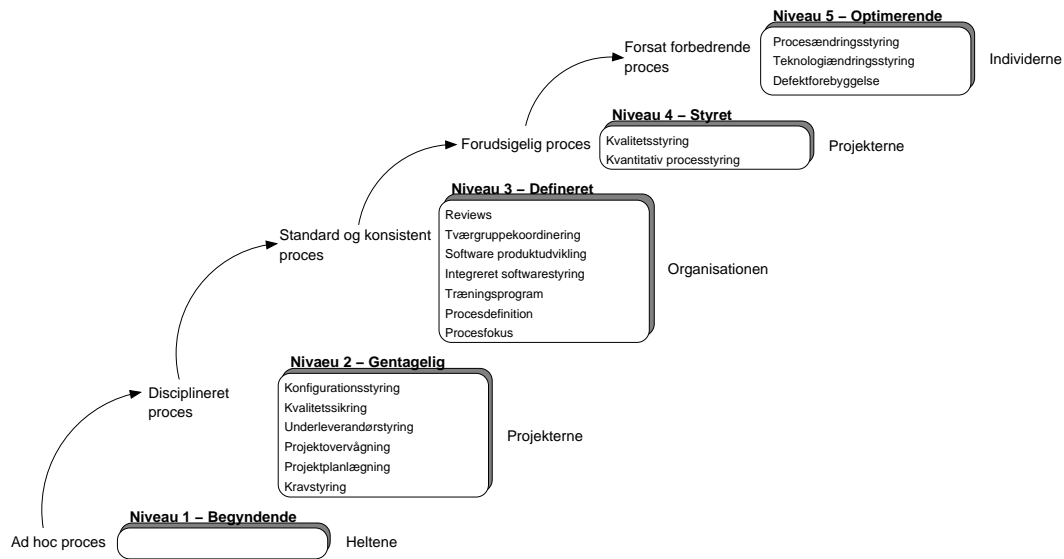
Modenhedsniveauer

Modenhedsmodellen er, som før nævnt, delt op i 5 overordnede modenhedsniveauer. Hvert af disse niveauer repræsenterer en basis for organisationer, som er i færd med at modne deres udviklingsproces. På figur 2-1 fremgår denne overordnede struktur af CMM. Hver kasse på figuren, som repræsenterer et niveau i modellen, er givet en betegnelse, som beskriver organisationens modenhedsstadiet. Desuden er hvert niveau opbygget af en række indsatsområder, Key Process Areas. Mellem kasserne er der en række pile, som fortæller hvilken udvikling i processen, der medfører en ændring i organisationens modenhed. Til højre på figuren er der ud for hvert modenhedsniveau angivet, hvad der fokuseres på i organisationen, når den bevæger sig mod et højere niveau. I takt med bevægelsen op gennem modellen øges organisationens evne til at styre planer, omkostninger og kvalitet. Dvs. organisationen forbedrer dens evne til, på en forudsigelig måde, at levere de aftalte produkter, med den aftalte funktionalitet, til tiden.

Når en organisation befinder sig på et givet modenhedsniveau, er der en række karakteristika, som kendetegner den. Disse karakteristika kan kort opsummeres til:

Niveau 1 - Begyndende Organisationen benytter en ad hoc proces, og udviklingsforløbet er ofte kaotisk. Succes afhænger i høj grad af dygtige medarbejdere og ledere. Det væsentligste karakteristika er dog, at problemer takles, som de opstår også kaldet brandslukning. I pressede situationer droppes udviklingsprocessen, og alt sættes ind på at gøre produktet færdigt. På dette niveau fokuseres der på heltene i organisationen. Disse helte er dem, som er med til at sikre, at projekterne i organisationen lykkes i kraft af heltenes erfaring og kompetence.

¹En *moden* organisation er karakteriseret ved evnen til at lede og vedligeholde udviklingsprocessen. Der eksisterer en defineret proces, som beskriver, hvordan arbejdet reelt udføres, og som følges af størstedelen af projekterne. Aktiviteterne i processen følges som planlagt, og processen bliver løbende evalueret og forbedret. Den *umodne* organisation er karakteriseret ved ikke at have en forudsigelig udviklingsproces. Udviklingsprocessen bliver ofte improviseret af ledelsen og udviklerne, og kriser løses som de opstår.



Figur 2-1: Den overordnede struktur af CMM, som kan danne basis for organisationers modningsproces.

Niveau 2 - Gentagelig Det, der kendetegner organisationer på dette niveau, er evnen til at gentage succes. Der eksisterer procedurer og værktøjer til styring af udviklingsprojekter, og projektplaner er baseret på definerede projektstandarder og erfaringer fra tidligere projekter. Ledelsen holder løbende øje med omkostninger, planer og funktionalitet, og problemer opdages hurtigt og løses. Fokus på dette niveau er på de enkelte projekter, og forbedringsinitiativerne sigter mod at forbedre deres præstationer.

Niveau 3 - Defineret På dette niveau dokumenteres, standardiseres og integreres ledelses- og udviklingsaktiviteter i en standardproces for organisationen. Denne standardproces bruges som udgangspunkt for de enkelte projekter, hvor den tilpasses til det enkeltes projekts behov. Processen sikrer en effektiv udnyttelse af organisationens ressourcer og en høj grad af produktkvalitet. På dette niveau er fokus rettet mod organisationen som helhed, og forbedringerne sigter imod at skabe større ensartethed og koordinering mellem projekterne. Derudover skal projekterne i organisationen trække på erfaringer fra andre projekter.

Niveau 4 - Styret Organisationer på dette niveau opsamler data vedrørende processen og produktet systematisk fra projektet. Hermed kan kvaliteten for processen og produktet løbende måles, vurderes og styres. Opsamlede data er tilgængelige for hele organisationen, og projekter kan drage nytte af tidligere erfaringer. Udnyttelse af data fra tidligere projekter gør, at organisationen er i stand til at producere produkter af høj kvalitet indenfor den afsatte tid. Fokus flyttes på dette niveau tilbage til de enkelte projekter, og indsatsen for niveauet sigter imod at forbedre kvaliteten af de enkelte projekters produkter samt at foretage dataopsamling fra projekterne.

Niveau 5 - Optimerende På dette niveau fortsætter organisationen med at optimere processen. Dette gøres ved løbende at evaluere processen for at identificere styrker og svagheder ved

processen, og ved at afprøve nye ideer og teknologier gennem pilotprojekter. Organisationen er i stand til at lære af tidligere fejl ved at analysere defekter og identificere årsager til defekterne. Fokus på dette niveau er igen flyttet tilbage på de enkelte medarbejdere i organisationen. Forbedringsindsatsen på dette niveau sigter imod at forøge den enkelte medarbejders læring. Dvs. at medarbejderne skal lære af deres tidligere fejl og være i stand til at vurdere udviklingsprocessen med henblik på en løbende optimering af processen.

Modenhedsniveauerne i CMM er den højeste abstraktion over modellen. Hvert niveau definerer en række mål for, hvornår en organisation har nået et givet niveau. Disse mål kan bruges til at vurdere, fx gennem et spørgeskema, om en organisation har en modenhed svarende til niveauet. Vigtigst af alt kan målene bruges som fokuspunkter på vejen mod den modne organisation.

2.2.1 Vurdering af CMM

CMM er en af de mest anerkendte procesforbedringsmodeller i softwareindustrien [Zah98, s. 58]. I CMM ligger vægten på kontrol af processen som vejen til forbedring af en softwareorganisations produkter. Der fokuseres på, hvordan udviklingsprocesserne kan kontrolleres, hvilket gør, at normen i højere grad henvender sig til ledelsen i softwareorganisationerne end mod de udviklere, som rent faktisk skal udføre processerne [Bac95]. Det er et af de kritikpunkter, som James Bach rejser, og han mener, at organisationerne i stedet bør fokusere på at få sine helte til at trives, dvs. de medarbejdere, som ofte er med til at gennemføre succesfulde projekter i organisationer med en lav modenhed, set i forhold til CMM. Bach argumenterer for, at en organisation kan have nok så formaliserede procesbeskrivelser, men det er kun i hænderne på "heltemodige" udviklere, at beskrivelserne kommer til deres fulde udfoldelse.

Dette henviser til, at innovation er en personlig egenskab, som en organisation kun kan få gennem enkeltpersoner. Ligemeget hvor mange processer der defineres og nedskrives, falder det stadig tilbage på, at der sidder innovative medarbejdere og udviklere i organisationen. Her kommer forholdet mellem en organisations mål og behov og CMMs krav ind. I mange tilfælde kræver CMM, at der nedsættes forskellige grupper til at lede og kontrollere procesforbedringsarbejdet. I en virksomhed med mange medarbejdere er det muligvis en lille del af medarbejderne dette drejer sig om, men i mindre virksomheder, hvor der er færre medarbejdere, kan det være en helt anden sag at afsætte folk til at holde øje med, at processerne forløber planmæssigt. Det er også let for ledelsen at stirre sig blind på CMM og de medfølgende krav, hvormed den samlede produktivitet i virksomheden går ned, på trods af, at det burde gå fremad ifølge CMM.

Endvidere er det tydeligt, at CMM er kraftig inspireret af Osterweils ideer om programmering af processen [Ost87], hvilket kommer til udtryk i CMMs vægt på formalisering [Aae02]. Bl.a. igennem de enkelte niveauer som organisationer skal igennem, hvor de modnes efterhånden, som de definerer, implementerer, måler, kontrollerer og forbedrer deres softwareprocesser. Fokus ligger helt klar på, hvordan de enkelte processer kan formaliseres, og der er ikke meget vægt på, hvem der rent faktisk skal bruge processerne i sidste ende. Dvs. der er fokus på processerne og ikke på udviklerne.

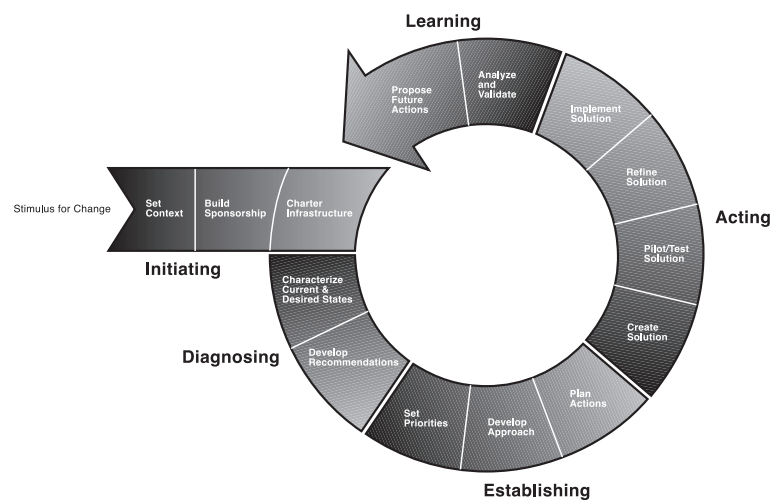
I dette afsnit har vi kort gennemgået principperne i CMM normen. CMM opstiller nogle rammer for, hvordan softwareorganisationer kan modnes, men kommer ikke med noget bud på, hvordan det

rent faktisk skal gøres. I næste afsnit vil vi præsentere en model for, hvordan forbedringsinitiativer i praksis kan indføres.

2.3 IDEAL modellen

IDEAL modellen er, hvad der betragtes som de facto standarden indenfor procesforbedringsmodeller, og selv hvis organisationer ikke formelt følger IDEAL modellen, så er det alligevel ofte lignende trin, der bliver foretaget. Følgende afsnit er baseret på [McF96] og [Zah98].

IDEAL modellen sigter mod at igangsætte procesforbedringsinitiativer i en organisation, som ønsker at måle sig op imod CMM normen. Modellen beskriver de faser, aktiviteter og ressourcer, som der er behov for, for at opnå en succesfuld procesforbedringseffekt. Modellen er opdelt i fem faser, som udføres cyklisk, se figur 2-2, dvs. det er en proces som i princippet aldrig stopper. De fem faser er **initiering**, **diagnosticering**, **etablering**, **aktion** og **læring**.



Figur 2-2: Ideal modellen.

Initiering

Dette er den indledende fase, hvor det erkendes, at der er behov for at etablere et forbedringsprojekt. Erkendelsen af behovet kan udspringe fra ledelsen eller fra medarbejderne, men for at et forbedringsprojekt kan gennemføres, er det nødvendigt at få ledelsens opbakning. Ledelsen skal dels stå for at allokere ressourcer til gennemførelse af projektet, men også som garant for, at der er politisk opbakning til projektet, da det uundgåeligt vil støde på modstand i organisationen. Derudover skal der sættes mål for forbedringsprojektet ud fra de behov, som er erkendt, og der skal etableres en plan gældende for de første tre faser. Til sidst skal forbedringsinfrastrukturen etableres, som regel i form af oprettelse af en styregruppe og en SPI gruppe. Aktiviteterne i initieringsfasen er:

- Erkend behov for forandring.
- Opnå opbakning fra ledelsens side.
- Etabler den nødvendige infrastruktur.

Diagnosticering

I diagnosticeringsfasen etableres et udgangspunkt på baggrund af en modenhedsmåling. Ud fra dette udgangspunkt udarbejdes en handlingsplan, som, foruden den pågældende måling, også inkluderer tidligere erfaringer fra forbedringsprojekter. Nye mål sættes for hvor langt, organisationen skal være nået inden næste iteration. Aktiviteterne i diagnosticeringsfasen er:

- Vurder resultatet af undersøgelsen.
- Udarbejd anbefalinger og dokumenter resultater.

Etablering

I etableringsfasen udarbejdes procesforbedringsstrategien med et forslag over mulige forbedringer samt strategier for, hvorledes de prioriterede mål skal nås. Herefter skal de mål, som blev fastlagt i initieringsfasen omformuleres i nye målbare mål, og metrikker udarbejdes for at kunne følge procesforbedringsprocessen. Aktiviteterne i etableringsfasen er:

- Sæt mål og prioriter blandt anbefalingerne fra diagnosticeringsfasen.
- Planlæg procesforbedringen.
- Etabler forbedringsgrupper.

Aktion

I tredje fase, aktionsfasen, iværksættes deciderede procesforbedringsinitiativer med det formål at ændre organisationen. Ændringerne iværksættes i pilotprojekter, inden de udrulles i organisationen. Derfor skal den ændrede proces testes og evalueres i pilotprojekter for at få fastslået, om løsningerne vil fungere. Først derefter vil resten af organisationen få glæde af forbedringerne. Aktiviteterne i aktionsfasen er:

- Definer processer og metrikker.
- Planlæg og udfør pilotprojekter.
- Etabler plan for at overvåge effektiviteten af de nye processer i pilotprojekterne.
- Udbred de nye processer og metrikker til resten af organisationen.

Læring

I læringsfasen gælder det om at få samlet op på hele forløbet, for at organisationen næste gang IDEAL modellen tages i anvendelse kan gøre processen mere effektiv, og gentagelse af tidligere fejl kan undgås. De erfaringer, som er gjort ved gennemførelse af pilotprojekterne og fra indsatsen under forbedringsaktiviteterne, opsamles, analyseres og dokumenteres. Til sidst evalueres modenhedsprojektets strategi, metoder og infrastruktur, og IDEAL modellen gentages. Til forskel fra første iteration springes initieringsfasen i fremtiden over. Aktiviteterne i læringsfasen er:

- Dokumenter og analyser procesforbedringen.
- Revider den organisatoriske fremgangsmåde inden næste iteration.

2.3.1 Vurdering af IDEAL modellen

Gennem de ovenstående fem trin planlægges og udføres forbedringsprocessen nøje for at forebygge uventede komplikationer undervejs, og desuden sikres det, at erfaringer fra forbedringsaktiviteterne bliver opsamlet.

IDEAL modellen lægger sig op af CMM, og er beregnet på, at en iteration hæver modenhedsniveauet trin for trin. Det betyder, at iterationerne bliver store og kræver meget detaljeret planlægning, og tager mange kræfter at gennemføre. Denne nøje planlagte proces medfører, at det er svært at korrigere indsatsen undervejs, hvis der opstår uventede problemer. Det betyder også, at forbedringsinitiativerne hurtigt kan blive udsat for politisk pres for at få dem gennemført, indenfor en vis tidsramme, uden at der tages hensyn til, om forbedringerne når at forankre sig i organisationen.

Endnu vigtigere er det, at IDEAL modellen har en meget formel tilgang til procesforbedring. Det antages, at forbedringer kan planlægges i detaljer, og at alt, eller i hvert fald det meste af forbedringsforløbet, vil opføre sig forudsigeligt. Desværre er dette ikke altid tilfældet, og derfor kan IDEAL modellens tilgangsvinkel nogle gange vise sig at svigte.

Derudover fokuserer IDEAL modellen meget på at trække processer ned over hovedet på andre end dem, der udarbejdede dem i første omgang. Det giver sig udtryk i, at det er en SPI gruppe, som designer og dokumenterer processerne, hvorefter de bliver udrullet i organisationen. Det kan bevirke at udviklerne føler en fremmedgørelse overfor processerne, og de vil ikke være særlig motiverede for at bruge dem. Derudover kommer IDEAL modellen ikke med konkrete bud på, hvordan forbedringsinitiativer føres ud i livet, den specificerer blot nogle faser, forbedringen skal igennem for at blive implementeret, og er dermed en meget bred og generel model. En egenskab ved IDEAL modellen er også, at den i sin struktur ligner vandfaldsmodellen for softwareudvikling, hvor de enkelte faser kommer i trin.

Opsummerende er IDEAL modellen kendetegnet ved at foretage en grundig analyse af gældende praksis, designe og dokumentere de nye processer for dernæst at afprøve dem igennem pilotprojekter. Til sidst implementeres de nyudviklede processer i hele organisationen på en gang. Udviklingen af processerne foregår typisk i en SPI gruppe, der kan være sammensat på forskellige måder. Nogle steder består SPI gruppen af ledende medarbejdere, andre steder består den af medarbejdere, der har en interesse for SPI. Fælles for dem er, at det oftest er SPI gruppen, der er initiativtager for hvilket problem, der skal fokuseres på. De er ansvarlige for at designe en løsning, dokumentere denne løsningen i procesmodeller, afprøve den, for til sidst at udbrede den i organisationen. Det er derfor SPI gruppen, som sidder med hele ansvaret for, at der sker forbedringer. Igen kan tankegangen føres tilbage til programmeringsopfattelsen, hvor der er fokus på processerne, og der foregår en klar adskillelse mellem dem, der udarbejder processerne, og dem som senere skal bruge dem.

2.4 Læring og læreprocesser

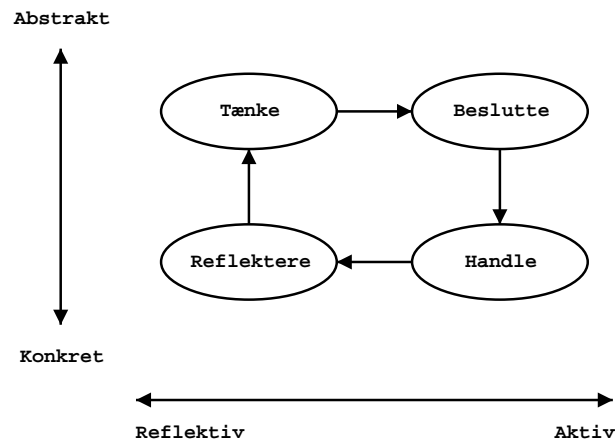
En egenskab ved traditionel SPI handler om synet på, hvor læring bør foregå. Følgende afsnit er hovedsageligt baseret på [JF97] og [FvH95]. Inspirationen i [FvH95] stammer fra Joop Swieringa og André Wierdsma "Becoming a Learning Organization". Disse to forfattere er selv blevet inspireret af Argyris & Schön [AS96] og D.A. Kolb [Kol84]. I denne sammenhæng kan det bemærkes,

at begrebet triple loop læring, som vi bruger, er det samme som deuterolearning som Argyris og Schön benytter.

Læring handler om at udfordre og overskride egne tænkemåder dvs. evnen til refleksion. At give slip på det, der ikke længere giver mening eller har effekt, kræver mod, og det kræver, at deltagerne er åbne overfor at tilegne sig alternative tænke- og handlemåder. Traditionel undervisning er baseret på indlæring og fokuserer på at øge deltagerens kompetencer. Begrebet læring er skridtet videre, hvor det også kræves, at deltagerne har vilje og evne til handling. Bevidst læring betyder, at de eksisterende kompetencer udvides, og den bevidste læring udvikler læringspotentialer, hvilket medfører evnen til at kontrollere ens egen læring og dermed sætte den enkelte i stand til at lære at lære.

2.4.1 Læring

I figur 2-3 har vi illustreret Kolbs læringscyklus [JF97], som beskriver den bevidste læring som en kontinuerlig proces med fire elementer. Selve processen startes med et spørgsmål eller problem, hvilket, i sig selv, er et tegn på, at der allerede er nogen, som har foretaget en refleksion. Det første element er *tænke*, hvor det forsøges at forstå eller skabe overblik over problemet, og via analyse skabes der struktur i problemet og dets bestanddele. Næste element er *beslutte*, hvor der på baggrund af analysen træffes beslutning om, hvordan problemet skal løses. Ved det tredje element, *handle*, føres den tænkte problemløsning ud i livet via nogle konkrete handlinger. Til sidst er der elementet *reflektere*, som er den systematiske evaluering af handlingens konsekvens, og som er med til at kaste lys over værdien og effekten af indholdet i de foregående elementer. Spørgsmål som hvad gik godt, mindre godt, hvad skal vi være opmærksomme på, og hvad bør vi ændre er typiske spørgsmål under denne fase. Refleksionen skaber evne til udvikling, og uden refleksion ville indlæringsprocessen have karakter af at være famlende.

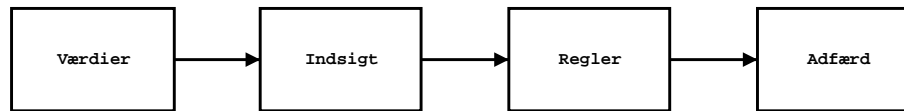


Figur 2-3: Kolbs læringscyklus.

Ud over læring er det vigtigt at være i stand til at aflære, dvs. at kassere forældet og overflødig praksis. Som regel vil aflæring være den største forhindring i at opnå højere kompetence og for at skabe ægte læring og forandring [JF97].

2.4.2 Organisatorisk adfærd

Før vi kan tale om organisatorisk læring, må vi først forstå hvad organisatorisk adfærd er, hvilket vi har illustreret i figur 2-4. De tre faktorer som styrer organisatorisk adfærd er regler, indsigt og værdier [FvH95]. *Regler* er det første, som styrer en organisations adfærd, og det kan være både nedskrevne og uskrevne regler. Reglerne styrer rutiner og giver udtryk for, hvad individerne i organisationen må og ikke må, og de vil som regel være dokumenteret i personalehåndbogen eller gennem kvalitetsprocedurer i et kvalitetssystem [FvH95].



Figur 2-4: Organisatorisk adfærd.

Kvalitetsprocedurerne er baseret på viden og *indsigt*, og de er udarbejdet, fordi organisationen på et eller andet tidspunkt har lært, hvad der skal til for at opnå en god kvalitet. Det er derfor den kollektive viden og indsigt, som er bestemmende for, hvilke regler der eksisterer. Den kollektive indsigt handler også om kendskab til udviklingsmetoder, branchekendskab, markedskendskab og er dermed organisationens forretningsgrundlag.

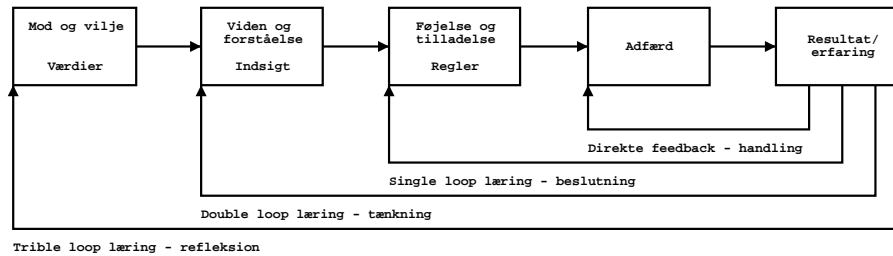
Hvis organisationen på et eller andet tidspunkt erkender, at forretningsgrundlaget er væk, må organisationen gå tilbage til de grundlæggende *værdier* og principper, som forretningen er bygget på for at genoverveje, om de skal ændres. En eksempel kunne være IBM, der i sin tid havde et forretningsgrundlag, der byggede på at sælge kontor-maskiner, og som senere blev ændret til, at IBM sælger edb-løsninger.

Samlet kan vi konkludere, at organisatorisk adfærd består af en kombination af regler, indsigt og værdier. Hvis organisationens adfærd skal ændres, er det nødvendigt at påvirke organisationen på et af disse tre niveauer.

2.4.3 Læreprocesser

Kolbs læringscyklus er baseret på individuel læring, og det der er interessant i denne sammenhæng, er læring som en kollektiv proces. Organisatorisk læring er mere end individuel læring, og det er først en realitet, når individernes samlede kompetence er forankret i organisationens værdier og faktiske handlemåde [JF97]. Vi siger, at organisationer lærer, når de ændrer adfærd, hvilket også gælder den anden vej. Dvs. når organisationer forandrer sig eller ændrer adfærd, er det fordi, der er sket en kollektiv læring [FvH95]. Derfor er det væsentligt, for at forstå traditionel SPI, at undersøge, hvor læringen foregår, når organisationen skal forandres igennem IDEAL modellen.

Organisatorisk læring kan udtrykkes i dimensionerne progression og dybde. Progression er en proces, der starter med individet, forsætter med gruppen for til sidst at slutte med organisationen. Dybden i læring er udtrykt ved direkte feedback, single-, double- og triple loop læring. På figur 2-5 har vi koblet Kolbs læringscyklus sammen med de enkelte læringsloop og den organisatoriske adfærd jvf. [JF97] og [FvH95].



Figur 2-5: Læringsprocesser.

Direkte feedback er den enkleste form for læring, hvor personen gør sine erfaringer og indretter sine handlinger på baggrund af resultatet af tidligere handlinger. Dvs. at indlæringen giver anledning til en korrektion af den handling, der førte til erfaringen. Det karakteristiske er, at indlæringen er individuel, og hvor individet blot har behov for at være bedre i stand til at følge reglerne, hvilket bl.a. kan ske gennem bedre træning. Et eksempel på direkte feedback er, når en udvikler kompilerer sit program og får en syntaks fejl. Vedkommende finder stedet i programmet, hvor fejlen opstod, og opdager, at syntaksen ikke er fulgt. Fejlen rettes, og programmet kan nu kompilere, og udvikleren er klar over, at problemet skyldtes, at vedkommende ikke fulgte reglerne. Der sker en justering af den individuelle adfærd, og det typiske spørgsmål er, har jeg fulgt reglerne?

Single-loop læring træder et skridt tilbage og handler ikke længere om indlæring på individniveau, men om kollektiv læring. Single loop handler om at lære af de indtrufne resultater. Der stilles typisk ikke spørgsmål ved, hvorfor resultaterne blev som de blev blot om, hvordan de fremkom. Det handler om, når erfaring ikke længere er nok til at ændre sine handlinger inde for organisationens gældende regler. Reglerne må derfor ændres, eller der må laves nogle nye, men det fundamentale spørgsmål er hvordan? Et eksempel på single-loop læring er kvalitetssystemer, som handler om at ændre organisationens adfærd gennem regelændringer. Hvis en organisation fx på et tidligere tidspunkt har erfaret, at produkterne lider af for ringe kvalitet, og svaret er at skabe et kvalitetssystem, der kan sikre en bedre og mere ensartet kvalitet er der tale om single-loop læring. Der stilles ikke spørgsmål ved, om de bagvedliggende antagelser om at styring af adfærd gennem regler og standarder giver bedre kvalitet, men det handler om, hvordan vi kan ændre eller lave nye regler, som kan give os en bedre kvalitet af vores produkter. Temaet bag single-loop læring er mere af det samme, bare bedre, og handlinger styres af begivenheder, og ikke af indsigt og refleksion.

Double-loop læring træder endnu et skridt tilbage, når en erfaring gøres, og det er ikke længere nok at ændre reglerne inden for gældende viden og kutymer. Her holdes de procedurer og regler, der var bestemmende for handlingen og dermed det resultatmæssige udfald, op imod, hvorfor resultatet blev, som det blev. Spørgsmålet for forståelse bliver, hvorfor gør vi det overhovedet på den måde, som vi gør, hvorfor vælger vi ikke en ny strategi? double-loop læring handler om, at når en organisation drager konklusionen, at vidensgrundlaget ikke længere fører til hensigtsmæssige strategier, at så må viden fornys. For at udvide eksemplet fra før, så kan det være, at det gennem double loop læring opdages, at bedre kvalitet ikke nås gennem regler og procedurer i et kvalitetssystem. Lige meget hvor mange regler, der udformes og lægges ind, ændrer det ikke ved det

faktum, at kvaliteten af organisationens produkter stadigvæk ikke bliver bedre. Det drejer sig fundamentalt om, at der er brug for en fornyelse i form af en ny metode til at tage hånd om problemet. double-loop læring er svært, da det kan være vanskeligt at skulle beskæftige sig med organisationens indsigt og viden. Reaktionen fra organisationens medlemmer vil ofte være at undgå denne form for diskussioner og løbe fra problemet med undskyldningen, at de har travlt, og at der ikke er tid til det [FvH95]. Double-loop læring handler om, at når organisationens viden udfordres, vil organisationens adfærd blive ændret, og vi står over for en kollektiv læringsproces, der er startet af behovet for fornyelse.

Triple-loop læring betyder, at der trædes endnu et skridt tilbage, og der stilles spørgsmålstejn ved hvilke grundlæggende faktorer, som var bestemmende for, at reglerne ser ud, som de gør, og som derfor forklarer, hvorfor handlingen blev, som den blev. Ved triple-loop læring drejer det sig om, når refleksion over sine egne erfaringer fører til, at der uddrages den lære, at individets eller organisationens grundlæggende værdier og identitet ikke holder længere, og at de ikke mere fører til hensigtsmæssige modeller og indsamling af viden. Spørgsmålet er koblet til, hvad står jeg eller vi for, og det er en læringsproces, som er startet af behovet for udvikling.

Vi har opsummeret de forskellige læringsloops i tabel 2-1. Allerede nu kan vi se at det, der er relevant i forhold til traditionel SPI er at vurdere, hvor i organisationen henholdsvis single- og double-loop læring foregår, da SPI ikke handler om at ændre på organisationens grundlæggende værdier, men om at gøre organisationen mere effektiv og forudsigelig.

Indlæringsloop	Single-loop	Double-loop	Triple-loop
<i>Læringsområde</i>	Regler	Indsigt	Værdier
<i>Læringsniveau</i>	Føjelse og tilladelse	Viden og forståelse	Mod og vilje
<i>Læring</i>	Beslutning	Tænkning	Refleksion
<i>Problemspørgsmål</i>	Hvordan?	Hvorfor?	Hvad?
<i>Læringsresultat</i>	Forbedring	Fornyelse	Udvikling
<i>Involverede</i>	Medarbejdere	Mellemlidelse og medarbejdere	Topledelsen, mellemlidelse og medarbejdere
<i>Påvirker</i>	Operation	Koordination	Omgivelser

Tabel 2-1: Forskellige indlæringsloop.

2.4.4 Vurdering af læring

Organisatorisk læring starter som regel ved, at det først er individet, som lærer. Når det enkelte individ har lært, og det dernæst handler og beslutter ud fra de kollektive regler, det har lært, taler vi om organisatorisk adfærd. Dog er der ikke tale om, at individuel læring automatisk vil medføre organisatorisk læring, da organisationen ikke nødvendigvis ved alt, hvad den enkelte medarbejder ved. Medarbejdernes viden kommer ikke til udtryk i deres handlinger, og derved får organisationen ikke gavn af denne viden. Netop spørgsmålet om hvor viden, i forbindelse med traditionel SPI, læres er vigtig. Kigger vi tilbage på IDEAL modellen, er det især i SPI gruppen, at viden om processerne opsamles. Det i sig selv er et problem, at der på den måde sker en adskillelse mellem

SPI gruppens viden og organisationens viden. Hvis denne viden ikke udmønter sig i organisatorisk læring, vil organisationen heller ikke forandre sig.

Ofte forbindes læring med uddannelse og oplæring, men dette er som regel en fejltagtig antagelse, da der herved ikke sker en refleksion, men blot single-loop læring. I forhold til SPI kan vi se, at processerne, ifølge IDEAL modellen, skal udrulles fra SPI gruppen og ud i resten af organisationen. Det er et problem, da refleksionen er sket i SPI gruppen og derfor er løsrevet fra den praksis, som læringen skal forandre. Eller sagt på en anden måde, den viden, der skal til at forandre og stimulere en refleksion, skal placeres ude i organisationen, hvor processerne benyttes og ikke nødvendigvis i SPI gruppen.

Det er som om, at den traditionelle SPI tilgang betragter forandringer og dermed læring snævert og instrumentelt som ren og skær problemløsning. Hvis målet er at korrigere fejl, i forhold til organisationens regler, er det udmærket, men formålet med SPI er netop, at organisationen bl.a. skal forandre sig imod højere CMM niveau. Hvis de organisationer, som stiler mod højere CMM niveau, lever i en forholdsvis stabil verden, er der ikke noget problem ved den måde traditionel SPI foregår på. Så ville det instrumentelle udgangspunkt levere effektive løsninger på organisationens problemstillinger med entydige årsags- og virkningssammenhænge. single-loop læring vil være tilstrækkelig, når det handler om at udføre rutinearbejde, hvor udviklerne gentager, hvad de altid har gjort, og organisationen er dermed i stand til at opretholde en stabil effektivitet. Under stabile forhold vil organisationen ved single-loop læring nå sine mål, men da softwarefirmaer ofte løser komplekse problemstillinger, vil det ikke være nok med single-loop læring.

For at organisationer er i stand til at ændre adfærd og forny sig, er der behov for, at den problemløsning, der finder sted ude i organisationen, også forbindes med evnen til at identificere problemer, og forstå hvorfor problemerne opstår. Det skal ske i sammenhæng med den kontekst hvori problemer opleves. Netop det er problematisk i forhold til traditionel SPI, da problemerne ikke ses i den kontekst, hvori problemerne opleves, og selve problemidentifikationen og problemløsningen finder sted et andet sted nemlig i SPI gruppen. Hvis vi skal have fornyelse og forandring i organisationen, må vi have flyttet problemløsningen ud i organisationen til der, hvor problemerne opleves. Det gælder om at få flyttet læringstilgangen fra problemløsning, single loop læring, til problemforståelse, double-loop læring, og ud hvor problemerne opleves. Et af vores kritikpunkter ved traditionel SPI er, at double-loop læringen foregår i SPI gruppen dvs. væk fra, hvor problemerne opleves. Det er vigtigt, at refleksion flyttes ud i organisationen, da der ellers vil være en fare for, at der ikke sker organisatorisk læring, og læringen alene vil foregå i SPI gruppen.

Argyris og Schön [AS96] bruger to begreber omkring læring, som er teori-i-tanke, espoused theory, og teori-i-brug, theory-in-use. Teori-i-brug styrer individets konkrete handlinger og kan ses som de bagvedliggende normer, som individet rent faktisk følger, når det udfører en handling, fx foretager et review. Teori-i-tanke er det sæt af regler, som er udtrykt i organisationens procedurebeskrivelser i fx et kvalitetssystem, og som det forventes, at det enkelte individ følger. Det der kan hæmme en læringsproces er, hvis der eksisterer et skel mellem det, som individet siger det gør, og det som individet rent faktisk gør. En forudsætning for læring er, at individet er i stand til at sætte spørgsmålstejn ved den viden, som ligger til grund for dets handlinger. Individet må derfor først blive bevidst om hvad det faktisk gør, og det som det siger, det gør, ellers kan individet ikke ændre adfærd, og dermed kan organisationen heller ikke ændre adfærd. Ifølge Argyris og Schön har der

ikke fundet organisatorisk læring, dvs. double-loop læring, sted medmindre der er overensstemmelse mellem organisationens teori-i-tanke og teori-i-brug. Det er derfor også kun i forbindelse med double-loop læring, at det er nødvendigt at blive bevidst om forskellen i organisationens teori-i-brug og teori-i-tanke.

Traditionel SPI bygger på, at teori-i-tanke udvikles i en selvstændig SPI gruppe, og det er derfor også her, at double-loop læring finder sted. Derefter skal den viden, der er udviklet, udbredes i organisationen. Dvs. at opgaven er at få de enkelte udviklere ude i organisation til at ændre deres teori-i-brug så den stemmer overens med den teori-i-tanke, som er udviklet af SPI gruppen. Den måde som traditionel SPI gør det på er ved undervisning eller udformning af regelsæt, som udtrykkes gennem et kvalitetssystem. Det forventes, at den enkelte følger de nye procedurer, men problemet er, at der herved kun sker single-loop læring. Det som Argyris og Schön netop udtrykker er, at for at der kan ske overensstemmelse mellem teori-i-brug og teori-i-tanke, skal der ske double-loop læring hos den enkelte, ellers sker der ikke læring og dermed organisatorisk forandring.

Når individer og organisationer handler, gøres det ud fra et sæt af regler eller en teori-i-tanke, som de bruger til at udforme deres egen adfærd, og som benyttes til at forstå andres. Teori-i-tanke ligger ofte så dybt forankret i os, at vi sjældent er bevidste om, at vi bruger den. Paradokset er, at den teori-i-tanke, som vi tror, vi benytter, sjældent er den teori-i-brug, som vi rent faktisk bruger. Derfor bør SPI måske i højere grad basere sig på, i første omgang, at ændre teori-i-brug gennem double-loop læring for først derefter at dokumentere, hvordan den nye teori-i-brug er og omsætte den nye praksis til organisationens regler og viden til teori-i-tanke. Dermed skabes der overensstemmelse mellem det, vi siger, vi gør, og det vi faktisk gør. Traditionel SPI bygger på den modsatte antagelse, hvor der først skabes teori-i-tanke, som udbredes til organisationen. Hermed opstår paradokset, at der er uoverensstemmelse mellem teori-i-brug og teori-i-tanke, og der har ikke fundet organisatorisk læring sted, som er forudsætningen for organisatorisk adfærdsændring. Derved kan organisationen heller ikke forvente, at den vil opnå et højere CMM niveau, da målinger blot vil afsløre, at praksis er en anden end den, som SPI gruppen forventer.

2.5 Vurdering af traditionel SPI

Vi har i dette kapitel gennemgået den traditionelle SPI tilgang, og i det efterfølgende vil vi gennemgå nogle alternativer til traditionel SPI. For at kunne vurdere tilgangene op mod hinanden har vi identificeret en række egenskaber, som vi mener er væsentlige i forhold til dette projekt. Disse egenskaber er vist i tabel 2-2, hvor kendetegnene er opstillet som spørgsmål. Det er disse spørgsmål, vi vil forsøge at svare på i forhold til traditionel SPI i dette kapitel, og i senere kapitler i forhold til andre tilgange.

Traditionel SPI lægger op til, at bedre processer kun kan opnås gennem detaljeret planlægning, analyse og design, og at det er SPI gruppen, der har ansvaret for, at det sker. Når SPI gruppen sidder med ansvaret for, at organisationen får forbedret sine processer, så betyder det også, at det er SPI gruppen, der specificerer problemerne. Det er SPI gruppen, der identificerer problemer i udviklingsprocessen og igangsætter forbedringsinitiativer på baggrund af det, lige som det er SPI gruppen, der sætter de overordnede mål for organisationen, fx i forhold til CMM.

Det betyder, at *problemerkendelsen* foregår i SPI gruppen. Det kan medføre, at problemerne enten er meget overordnede, så de kan passe på alle dele af organisationen, eller at problemerne kun er

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Hvem erkender problemerne?
<i>Procesejerskab</i>	Hvem kommer med løsninger?
<i>Planlægning</i>	I hvor høj grad er SPI forløbet planlagt?
<i>Design</i>	Hvor meget fokus er der på design af processer?
<i>Dokumentation</i>	Hvor meget af processerne dokumenteres, og hvordan?
<i>Normer</i>	Hvilken holdning har tilgangen til normer?
<i>Fokus</i>	Hvad er tilgangens primære fokus?
<i>Læring</i>	Hvilken type læring foregår der i de forskellige dele af organisationen?

Tabel 2-2: Egenskaber for SPI tilgange.

gældende for dele af organisationen, hvilket betyder, at resten af organisationen ikke umiddelbart har nogen interesse i at løse problemerne. Samtidig er der den fare, at problemerne kan være formuleret på baggrund af en norm som CMM, uden at det rent faktisk er et problem i organisationen. På den anden side, så kan det være nødvendigt, at problemerkendelsen ligger i SPI gruppen, da det er SPI gruppen, der har kompetencerne til at identificere problemerne i organisationens udviklingsprocesser. Derved sikres det, at der ikke arbejdes på at forbedre processer, der ikke er vigtige for den samlede udviklingsproces.

Procesejerskabet har betydning for, hvem der kommer med løsninger på problemerne. I traditionel SPI er det SPI gruppen, der har ansvaret for organisationens procesforbedring, og dermed er det også SPI gruppen, der sidder med procesejerskabet. Det kan være praktisk i forhold til, at problemerne er identificeret af SPI gruppen selv, hvorfor der er en vis sandsynlighed for, at den også har forslag til, hvordan problemerne kan løses. Desværre kan det medføre, at SPI gruppen får for meget at lave, da den både skal erkende problemer og løse dem. Derfor inddrages andre dele af organisationen ofte i problemløsningsfasen, hvor der nedsættes arbejdsgrupper, der skal komme med forslag til løsninger, som SPI gruppen kan forholde sig til. Det ændrer dog ikke på, at procesejerskabet ligger i SPI gruppen, da det stadig er SPI gruppen, der sidder med den endelige beslutning om, hvordan problemerne bedst løses. En af ulemperne ved at have procesejerskabet liggende i SPI gruppen er, at der er en adskillelse mellem dem, der laver processerne og dem, der skal bruge processerne. Det betyder, at der er en risiko for, at de nye og forbedrede processer ikke vil blive accepteret, og at de dermed ikke vil blive fulgt.

Som allerede nævnt, så lægger traditionel SPI op til, at SPI forløb *planlægges* detaljeret før de føres ud i livet. Det bygger på ideen om, at alt kan kontrolleres og forudses, og ved at lave en grundig planlægning er der større sandsynlighed for, at forbedringsindsatsen bliver en succes. Med en detaljeret plan bliver det muligt for ledelsen at afsætte midler til forbedringerne, da planen siger, hvor mange ressourcer, det vil tage at gennemføre forbedringerne. Det bliver også muligt at følge forbedringerne løbende og at opstille milepæle, som kan bruges som delmål. Desværre gør denne planlægning forløbet mindre fleksibelt, da det bliver sværere at imødekomme forandringer undervejs uden at skulle revidere hele planen. Den betyder desuden, at det bliver svært at koordinere forbedringsarbejdet med de konkrete udviklingsprojekters aktiviteter, hvilket vil sige, at der er en

risiko for, at procesforbedringerne sker på områder, der ikke er relevante for udviklingsgrupperne på forbedringstidspunktet. Det kan lede til, at de forbedrede processer ikke får nogen indflydelse, da de er glemt, når udviklingsgrupperne når til det punkt, hvor de forbedrede processer er relevante.

En afledning af planlægningen er, at der med traditionel SPI også er fokus på *design* af processer. Processerne forsøges designet til at passe på flest mulige dele af organisationen, så de er brugbare for så mange som muligt. Ofte tages der udgangspunkt i de nuværende processer, men da de ofte er meget forskellige, er der en tendens til, at traditionel SPI prøver at lave en samlet proces, der gælder for alle. Det resulterer i en proces, der enten er meget overordnet, eller en proces der kun passer på dele af organisationen. Når processerne designes er der også en fare for, at brugerne af processerne føler sig fremmedgjorte, især hvis processerne også udsættes for politisk pres. Det kan være tilfældet hvis ledelsen prøver at styre forbedringerne med henblik på at sikre virksomhedens fremtid eller omdømme.

Når processerne er designet, skal de *dokumenteres*, og det er blot et spørgsmål om at nedskrive designet. Hvis processerne skal dokumenteres, er det nødvendigt at fastfryse processerne i et stykke tid mens denne dokumentation sker. Det er ikke svært, hvis processerne er blevet designet, men det er til gengæld svært, hvis der er tale om en løbende udvikling af processerne, da det vil resultere i, at dokumentationen næsten øjeblikkeligt er forældet. Hermed er et af argumenterne for at designe processer også givet. Hvis der stilles store krav til dokumentation, vil det betyde, at processerne ikke får lov til at ændre sig særlig meget. Det betyder, at det er vigtigt at ramme rigtigt hurtigt, da de beskrevne processer ellers vil være ubrugelige.

Kravet til dokumentation skyldes den traditionelle tilgangs forhold til *normer*, hvor tilgangen bygger på en norm, og formålet med tilgangen er at opfylde denne norm, men en norm er først opfyldt, når det kan påvises, at processerne eksisterer og bliver fulgt. Derfor er det nødvendigt med dokumentation. Normerne har til formål at gøre udviklingsprocesserne mere forudsigelige og effektive, men det kan på den anden side give anledning til, at processerne bliver opbygget på den forkerte baggrund, som allerede nævnt. Et kraftigt fokus på normer gør det muligt for organisationen at måle organisationens ydelse og sammenligne den med andres, men der er på den anden side den risiko, at forbedringerne kommer til at virke overflødige.

Generelt har traditionel SPI et kraftigt *fokus* på processer. Det kan ledes tilbage til Osterweils syn på processer som programmerbare. Hvis processer kan programmeres må det betyde, at de mennesker, der skal bruges processerne, opfører sig rationelt og som programmerbare enheder, der til enhver tid kan udskiftes. Dette fokus kan bidrage til at bagatellisere udviklernes rolle i forhold til procesforbedring og dermed skabe en yderligere adskillelse af processernes formål og processernes faktiske brug.

Endeligt er der tilgangens forhold til *læring*. Vi har i afsnit 2.4 argumenteret for, at double-loop læring er en nødvendighed for, at en SPI indsats kan blive en succes. Det er et problem med traditionel SPI, da der netop er en adskillelse af SPI gruppen som procesudviklere og udviklerne som procesbrugere. Double-loop læringen foregår i SPI gruppen og når ikke ned til udviklerne, hvorved der ikke sker nogen reel forandring.

Egenskaberne for traditionel SPI er opsummeret i tabel 2-3. Vi ser mange problemer ved disse egenskaber, desværre er der ikke et sted, der kan sættes ind for at løse problemerne. Det er nød-

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	SPI gruppen kommer med problemer.
<i>Procesejerskab</i>	SPI gruppen kommer med løsninger.
<i>Planlægning</i>	SPI forløb er planlagt i detaljer.
<i>Design</i>	Design af processer.
<i>Dokumentation</i>	Konstrueres på et bestemt tidspunkt.
<i>Normer</i>	Normer er udgangspunktet for SPI.
<i>Fokus</i>	Fokus er på processer.
<i>Læring</i>	Double-loop læring hos SPI gruppen.

Tabel 2-3: Traditionel SPI tilgang.

vendigt at lave mere fundamentale ændringer, hvilket kan være svært. Ved at have fokus på processerne fremfor på dem, der skal bruge processerne, bliver det svært at ændre på procesejerskabet og ved at bygge tilgangen op omkring normer, bliver der lagt mere vægt på at designe og dokumentere fremfor at udvikle processer, der kan bruges i praksis.

I dette kapitel vil vi foretage en vurdering af andre måder at udføre procesforbedring på. Vi vil tage udgangspunkt i artikler, som forholder sig kritisk til traditionel procesforbedring og foretage en vurdering af deres synspunkter. Dernæst vil vi undersøge alternative procesforbedringsmetoder for at se, om vi kan bruge nogle af principperne. Efter kapitlet vil læseren have en grundlæggende forståelse for alternative måder at foretage procesforbedring på samt hvilke tanker, der ligger bag fremgangsmåderne.

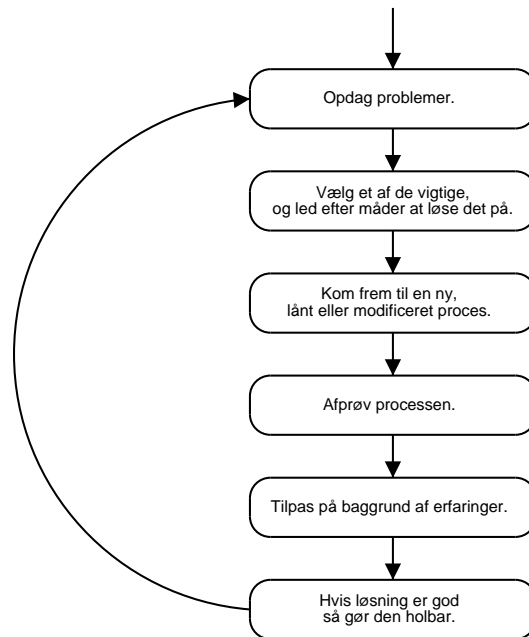
3.1 Bachs procesforbedringsmodel

James Bach kritiserer den traditionelle tilgang til procesforbedring for at fokusere for meget på processer og metoder i stedet for på at få organisationens helte til at trives. Organisationens helte er de medarbejdere, der ofte er med til at gennemføre succesfulde softwareprojekter [Bac95]. Bach præsenterer i [Bac98] sit alternativ til de traditionelle procesforbedringsmodeller.

Ifølge Bach adskiller hans alternativ sig fra den traditionelle procesforbedring ved, at procesforbedring betragtes ud fra en pragmatisk synsvinkel, hvor der anvendes en problemløsningsstilgang. Dvs. at det er de personer, der beskæftiger sig med procesforbedring, skal identificere problemerne i forbindelse med softwareudvikling og derefter lede efter realiserbare løsninger, som der også er ressourcer til, for at forbedre problemer.

Den grundlæggende tanke bag forbedringsmodellen er, at forbedringsaktiviteter ikke skal separeres fra praksis som selvstændige projekter. Herved kan det ifølge Bach undgås, at praktikerne saboterer forbedringsindsatsen, fordi de hellere vil lave rigtigt arbejde. Ved ikke at adskille procesforbedring fra praksis, kan det også undgås, at fokus skifter til processen, når der skal foretages procesforbedring. Dette fokusskifte kan føre til, at de personer, der skal procesforbedre, tænker på de anvendte fremgangsmåder som enten gode eller dårlige i stedet for at tænke på dem som gode eller dårlige i forhold til den specifikke situation. Ved at væve procesforbedringen ind i udviklingsprojekterne opnås til gengæld, at forbedringsaktiviteterne placeres i en motiverende kontekst. Dvs. der fokuseres på faktiske problemer, som der gøres noget ved, og derved forbedres resultatet af det konkrete udviklingsprojekt.

Bachs procesforbedringsmodel består af 6 skridt, som det fremgår af figur 3-1. Det første skridt består i at identificere reelle problemer i produkter eller resultater. Blandt disse problemer udvælges et vigtigt eller kronisk problem, og der ledes efter en måde at løse problemet på, enten hele problemet eller en del af problemet. Dernæst udvikles en ny proces, som kan løse problemet indenfor en acceptabel tidsramme og med acceptable omkostninger, eller en eksisterende proces lånes eller modificeres. Den nye proces afprøves på et rigtig projekt, og der foretages tilpasninger af processen på baggrund af erfaringer med processen og nye problemer, som opstår på baggrund



Figur 3-1: James Bachs forslag til en alternativ procesforbedringsmodel.

af processen. Hvis løsningen er værd at bevare, gøres den holdbar i organisationen, fx ved dokumentation, oplæring og værktøjsunderstøttelse, ellers gøres der forsøg på at forstå, hvorfor den ikke virkede. Herefter starter modellen forfra med et nyt problem.

For at modellen kan blive en succes kræves det, at de involverede personer er i stand til at identificere problemer, og at der kan opnås enighed om, hvilke problemer der er vigtige. I forhold til CMM og ISO så adskiller modellen sig væsentligt fra disse tilgange ved, at der ikke skal indføres bestemte processer og etableres bestemte institutioner, ligegyldigt hvilke problemer organisationen og projekterne står med. Grunden til denne forskel er, at Bach betrager normernes forsøg på at diktere bestemte processer som farlige. Det kan være med til at fremmedgøre udviklerne overfor processen, da det kan være svært at forholde sig til noget, der ikke anses som et problem, eller som udvikleren ikke forstår formålet med. Desuden fokuserer normerne på godkendelse af udviklingsarbejdet i forhold til et statisk sammenligningsgrundlag, som er defineret af en procesekspert, der ikke har ansvaret for de enkelte projekters resultater, i stedet for at fokusere på kritisk tænkning og uddannelse af medarbejderne. Derfor kan det være svært for udvikleren at vide, om han udfører arbejdet godt nok i forhold til sammenligningsgrundlaget.

3.1.1 Vurdering af Bachs procesforbedringsmodel

Selvom Bachs procesforbedringsmodel i sin opbygning ligner IDEAL modellen, mener vi, at den adskiller sig væsentligt i sin fremgangsmåde, da den lægger op til, at de evolutionære skridt, der tages er mindre end ved IDEAL modellen. Dette bunder i, at der tages fat i mindre problemer, som er specifikke for et konkret projekt, fx hvordan kan vi estimere bedre eller gøre vores møder mere effektive, og disse problemer forsøges løst, hvorefter løsningerne gøres tilgængelige for organisationen. Ved IDEAL modellen tages der derimod fat på store problemer som fx projektplanlægning

eller kvalitetssikring. Ved at tage mindre skidt opnås der hurtigere iterationer i forbedringsmodellen. Dette ser vi som en fordel, da resultaterne af arbejdet hurtigere bliver tydeligt for medarbejderne, hvilket kan være med til at skabe større motivation i forhold til procesforbedring. Der kan dog være ulemper ved en model, der benytter hurtige iterationer. For eksempel kan det ske, at tidligere forbedringer ikke bliver fasttømret godt nok i organisationen, så de bliver glemt igen, når næste iteration er i gang. Derudover kan små ændringer være svære at måle effekten af, og det kan skabe usikkerhed om, hvorvidt det overhovedet kan betale sig at fortsætte forbedringsprocessen.

Ud fra Bachs synspunkt om, at forbedringsindsatsen ikke bør separeres fra praksis, hans tidligere kritik af traditionelle forbedringsmetoders udprægede fokus på processer [Bac95] og hans synspunkt om, at procesforbedring drejer sig om at finde praktiske løsninger, kan vi konkludere, at Bachs forbedringsmodel lægger op til, at der fokuseres på udvikling i stedet for processen. Dette medfører, at der lægges op til at overdrage procesejerskabet og ansvaret for problemerkendelse til organisationens projektgrupper. Herved sikres, at læringen foregår på gruppe niveau, hvorved forståelsen for formålet med forbedringerne øges på dette niveau i organisationen.

På baggrund af ideen om, at procesforbedring skal tage udgangspunkt i projektgruppernes konkrete problemer, som de opstår, må det antages, at SPI indsatsen ikke kan planlægges i samme grad som ved traditionel procesforbedring. I stedet bliver forløbet præget af at være improviseret, da der tages udgangspunkt i de problemer, der eksisterer. Dette faktum gør, at det ikke er muligt for proceseksperter at sidde i baglokalet og designe nye processer. Derfor lægger Bachs forbedringsmodel op til, at der primært er fokus på etablering af en forbedret praksis og ikke på at få designet en proces. Dette kommer også til udtryk i modellen, da afprøvningen af processen kommer før dokumentationen af processen. I tabel 3-1 er vores vurdering af Bachs alternative forbedringsmodel opsummeret.

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen erkender problemerne ved deres udviklingsproces.
<i>Procesejerskab</i>	Projektgruppen finder frem til løsninger på problemerne.
<i>Planlægning</i>	Improviseret SPI forløb.
<i>Design</i>	Primært fokus på etablering af praksis, derefter modelleres resultatet.
<i>Dokumentation</i>	Rekonstrueres løbende.
<i>Normer</i>	Normer kan være hæmmende for SPI.
<i>Fokus</i>	Fokus på udviklerne.
<i>Læring</i>	Double-loop læring hos projektgruppen.

Tabel 3-1: Bachs procesforbedringsmodel i forhold til de otte egenskaber og deres kendetegn.

3.2 End-user SPI

I artiklen [Aae02] præsenterer Ivan Aaen begrebet *End-user SPI* som et alternativ til den traditionelle procesforbedring. Ifølge Aaen fokuserer den traditionelle SPI tilgang for meget på design,

i form af konstruktion og videreudvikling af procesmodeller som middel til forbedring af udviklingsprocessen. Denne fokusering medfører, at hovedvægten i indsatsen lægges på formalisering og eksternalisering¹, samt at udviklingen af procesmodellerne bliver separeret fra anvendelsen. Han anser det som et problem, at denne fokusering anlægges og mener, at den i værste fald kan være hæmmende for forbedringsindsatsens succes. Gennem begrebet End-user SPI ønsker Aaen derfor at dæmme op for fokuseringen på eksternalisering og adskillelse ved at anlægge et andet syn på procesforbedring. For at underbygge sine synspunkter henter han inspiration i de to forskningstraditioner: Vidensledelse og organisatorisk design.

SPI som vidensledelse

Aaen ser det som et fundamentalt problem ved SPI at få sikret en fælles forståelse for udviklingsprocesserne blandt praktikerne, som følger dem. I den henseende drejer SPI sig om vidensledelse, hvor der skal konstrueres og forandres procesviden hos individer, grupper og organisationer. Ved at overføre teori fra vidensledelse til SPI påpeger Aaen en række problemer ved adskillelse af design fra praksis og eksternalisering af procesviden.

Netop adskillelsen af design fra praksis er et typisk problem ved vidensledelse, da det har tendens til at nedtone vigtigheden af dialog, interaktion, udveksling og læring af procesviden og i stedet fokusere på teknologi som midlet til vidensdeling. Separationen vil sandsynligvis også hindre den reflekterende dialog mellem procesbrugere og procesdesignere, hvorved der ikke stilles spørgsmålstejn ved hvorfor, hvordan og relevansen af de designede processer. Det kan føre til, at processerne fastfryses, og der ikke sker nogen videreudvikling, da de mere er ideologier, der skal overholdes, end artifakter, som er til diskussion og løbende forandring. Procesbrugere er i sidste ende både kilde til og genstand for procesviden, derfor skal organisationen have opbygget og vedligeholdt en fælles kontekst, som udvikler og ensretter procesviden. En sådan kontekst opbygges fx af en forståelse for de specielle markedsvilkår, kunderelationer, tidligere udviklingsprojekter, procesinformation og procesnormer, og er nødvendig for at opnå en dialog om rationalet bag og forbedring af processerne og for at opnå en fælles praksis. Ved adskillelse af design fra praksis er det en implikation fra IDEAL modellen, at eksperimentering med nye processer er en del af procesdesignet og adskiller sig fra den regelmæssig anvendelse af processerne. Best practice som udvikles gennem IDEAL modellen indkapsler anvendt procesviden, og gør denne viden tilgængelig for praktikerne. Sådanne best practices kan være medvirkende til at fastfryse procesvidenen og undertrykke innovationer og forbedringer skabt af praktikerne, som har afvejet fra den eksisterende best practice. Herved går der viden tabt, som ikke er fremkommet ved officielle pilotprojekter.

Et andet typisk problem er forsøget på at eksternalisere praktikernes erfaring og viden om processerne i formaliserede procesbeskrivelser. Ved traditionel SPI opfattes procesviden som noget, der kan eksternaliseres i procesbeskrivelser. Dette syn medfører, at SPI indsatsen får et fokus på data og informationer, hvilket kan være på bekostning af opmærksomhed på processer, opbygning af viden og kompetence blandt praktikerne samt deling af praktikernes procesviden. At adgang til eksternaliseret viden, i form af data og informationer, er det samme som adgang til organisatorisk viden påpeges som en misforstået antagelse blandt mange softwareorganisationer. For topledelsen

¹Eksternalisering er oversat fra det engelske ord *Externalization*, som i forbindelse med SPI betyder formalisering af, hvilke mål organisationen har for dens produkter og af hvilken proces, den ønsker at anvende for at opnå disse mål.

giver eksternalisering af processer derimod ofte mening, da der herved er mulighed for at foretage statistiske analyser og vurderinger af udviklingsprocessen. Problemet er, at procesviden herved betragtes som noget, der eksisterer udenfor hovederne af praktikerne, som derved får en sekundær rolle i SPI indsatsen, hvor det primære er eksternalisering af praktikernes viden. Dermed bliver SPI indsatsen til et projekt, som udføres i baglokalet, hvorved opmærksomheden flyttes fra praktikerne, som både er kilden og målet for procesvidenen. Dette kan føre til, at der opstår to former for viden i organisationen: Den eksplicit formulerede og den implicite, som eksisterer i hovederne på praktikerne. En sådan opdeling kan give bagslag, hvis organisationen honorerer brugen af eksplicit viden og afviser brugen af implicit viden. Praktikere kan finde på at tilsidesætte eksplicit viden, hvis den er modstridende med deres implicite viden, og procesdesignere kan nedtone den implicite viden i deres iver for at formalisere den eksplicite viden.

SPI som organisatorisk design

Aaen mener også, at SPI kan betragtes som et forsøg på et foretage organisatorisk design, da SPI handler om at ændre måden, hvorpå softwareorganisationer udvikler og vedligeholder software. Med udgangspunkt i Karl Weicks opfattelse af organisatorisk design kommer Aaen frem til tre alternative ideer til SPI:

- Procesmodellering og SPI er en fortløbende aktivitet.
- Ansvar for redesign af processen spredes ud i organisationen.
- Softwareprocesser skal være simple og åbne for fortolkning.

Ud fra anskuelserne af SPI som vidensledelse og organisatorisk design foreslår Aaen en alternativ tilgang til procesforbedring, som er funderet i begrebet End-user SPI.

Ideer til End-user SPI

Ved End-user SPI er det væsentligt, at der fokuseres på forandring via læring og fokusering på den reflekterende praktiker, at der gives tilladelse til improvisation og at praktikerne sættes i centrum for SPI indsatsen. Hvordan dette konkret skal gøres, og hvordan en alternativ tilgang kan se ud, kommer Aaen ikke med et konkret bud på, men han præsenterer dog en række grundlæggende ideer:

- *Anvend tilgange til etablering af processer på gruppe niveau, fx Team Software Process og Extreme Programming.*
- *Anvend processpecialister som trænere og vejledere.* Hvis processer skal udvikles for, med og af praktikerne, så har de muligvis behov for instruktion og vejledning fra processpecialister. Desuden kan specialisterne være med til at udbrede procesideer i organisationen.
- *Betragt processen som en opskrift snarere end en arkitekturtegning.* Evnen og friheden til at improvisere under organisationens fælles standarder er essentielt for professionelt arbejde. Softwareprocesser er noget, der opstår, og som er åbne for improvisation.
- *Forbedre først, modeller derefter resultatet.* Dvs. integrere SPI indsatsen i de igangværende projekter og modellere den daglige praksis og ikke, hvordan det skal gøres i fremtiden.
- *Med en smule struktur kommer man langt.* Definer de essentielle proceselementer i stedet for alle elementerne. Det bør overvejes, om der ikke kan anvendes simple og fleksible procesmodeller af motiverede og kompetente mennesker.

- *Anvend indlejrede og løbende procesvurderinger på projektniveau.* Der eksisterer i dag en række lette vurderingsmetoder, der kan anvendes på projektniveau til at give hurtig feedback til praktikerne om, hvorvidt arbejdsprocessen er i overensstemmelse med normer, som fx CMM.

Ifølge Aaen er nutidens procesdesignere besat af hangen til kontrol, og de tror, at processer skal kontrolleres på samme måde som software. Denne overbevisning har ikke megen berettigelse i forhold til SPI, og det er derfor på tide, at softwareindustrien forkaster denne antagelse. I stedet bør der fokuseres på kompetence og ansvar som vejen til rigtig procesforbedring.

3.2.1 Vurdering af End-user SPI

Det primære i End-user SPI er, at praktikerne skal sættes i centrum for SPI indsatsen, og at forandring skal ske gennem læring hos praktikerne. Dette medfører, at SPI indsatsen skal tage udgangspunkt i konkrete problemstillinger i organisationen, hvorved det er svært at planlægge SPI indsatsen i samme grad som ved traditionel SPI. Derfor vil en SPI tilgang, der tager udgangspunkt i End-user SPI være præget af improvisation. Tilgangen sikrer, at sandsynligheden for fremmedgørelse mindskes i forhold til traditionel SPI, da forandring sker gennem læring hos den enkelte praktiker. Herved sker forbedringen ved, at praktikerne reflekterer over processen og indser dens problemer, hvilket bør føre til, at han forstår formålet med at foretage forbedringer.

Ved at anvende denne tilgang bliver ansvaret for udvikling og redesign spredt ud til praktikerne i organisationen, hvorved det er projektgrupperne, som får ansvaret for at erkende hvilke problemer, der er i udviklingsprocessen, og de får ansvaret for processen, dvs. de får procesejerskabet. For at understøtte dette ansvar foreslår Aaen, at der blandt andet kan anvendes såkaldte lette vurderingsmetoder i forbindelse med projekterne, hvorved praktikerne kan få hurtigt feedback omkring deres forbedringsindsats. En hurtig feedback er noget, som vi mener kan være en medvirkende faktor til at opnå erkendelse af problemer i processen og virke motiverende for praktikerne, da de hurtigt kan se resultatet af deres forbedringsindsats. Det kræver dog, at praktikerne er interesserede i at forbedre sig i forhold til en norm.

I stedet for at designe procesbeskrivelserne lægges der i End-user SPI op til, at der først skal etableres en praksis. Når praksis er etableret, modelleres resultatet, dvs. hvad praktikerne faktisk gør. Efterfølgende skal procesbeskrivelserne løbende rekonstrueres, da procesmodellering og SPI er en fortløbende aktivitet. I tabel 3-2 er End-user SPI, i forhold til de otte egenskaber, kort opsummeret.

3.3 Personal Software Process

Personal Software Process (PSP) er en metode udviklet af Humphrey [Hum00a] til at udvikle den enkelte softwareudviklers disciplin. Den bygger på, at hver udvikler er ansvarlig for sin egen proces, og at en proces ikke kan forbedres, før udvikleren har indset fejlene i processen. PSP hjælper udviklere med at indse fejl i deres nuværende proces, og gør det dermed muligt at forbedre processerne. Dette sker primært gennem omfattende dataopsamling og -behandling, der gør det muligt at identificere, hvor tiden bruges, og dermed hvor problemerne ligger.

Kort fortalt er grundprincipperne, der tilstræbes gennem PSP, som følger:

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen erkender problemerne ved deres udviklingsproces.
<i>Procesejerskab</i>	Ansvar for processen udbredes til projektgrupperne.
<i>Planlægning</i>	SPI forløbet improviseres.
<i>Design</i>	Primært fokus på etablering af praksis, derefter modelleres resultatet.
<i>Dokumentation</i>	Procesbeskrivelser skal rekonstrueres løbende.
<i>Normer</i>	Normer er nødvendige for at opnå en dialog om processer og en fælles praksis.
<i>Fokus</i>	Fokus på udviklerne.
<i>Læring</i>	Double-loop læring hos projektgruppen.

Tabel 3-2: End-user SPI i forhold til de otte egenskaber og deres kendetegn.

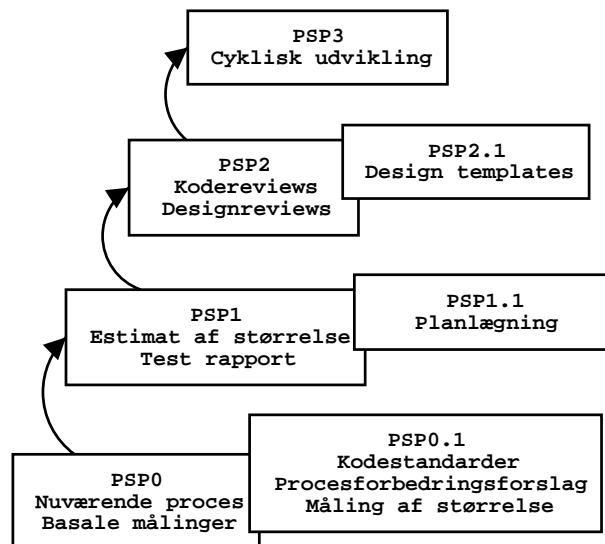
- For at være effektiv skal hver udvikler planlægge arbejdet, og planerne skal være baseret på personlige data.
- For at forbedre ydelsen skal hver udvikler følge en defineret proces, som også måles.
- For at kunne lave produkter af høj kvalitet skal udvikleren føle sig ansvarlig for kvaliteten af produkterne, og udviklerne skal stræbe mod at udføre arbejde af højere kvalitet.
- Jo tidligere en fejl findes og rettes, jo billigere er det at rette dem.
- Det er billigere at forebygge fejl end at rette dem.
- Den korrekte måde er altid den hurtigste og billigste.

Disse principper forsøger PSP at lære den enkelte udvikler gennem syv trin, som er vist på figur 3-2. Hvert af de syv trin er understøttet af et processcript, der fortæller hvilke aktiviteter, der skal udføres på det pågældende trin, samt et antal templates til at hjælpe med den løbende dataopsamling. Gennem trinene lærer udviklerne de grundlæggende teknikker, der er nødvendige for, at PSP kan fungere. Det drejer sig primært om planlægning, måling og strukturering af arbejdet.

3.3.1 Vurdering af PSP

Der hersker ikke tvivl om, at PSP kan forbedre udviklingsprocessen væsentligt. En række undersøgelser peger på, at i flere tilfælde er antallet af fejl blevet reduceret kraftigt, og tidsestimater er blevet meget mere præcise, og på trods af den omfattende dataopsamling er produktiviteten, målt i antal linjers kode produceret i timen, kun faldet minimalt [PUG97] [FHK⁺97]. Baggrunden for dette er, at muligheden for at lave realistiske estimater bliver større, når der er historiske data til rådighed. Desuden gør de historiske data også, at det bliver lettere at lære af tidligere erfaringer.

Som nævnt består en stor del af PSP af målinger og databehandling. I en af de få rapporter om PSP, der ikke er skrevet af Humphrey selv, [PUG97], nævnes dette som en ulempe. Især hvis databehandlingen ikke understøttes tilstrækkeligt af værktøjer, er der meget stor risiko for, at dataopsamlingen og -behandlingen ikke kommer til at foregå, og derved kan udvikleren ikke opfylde målet med PSP. Et andet kritikpunkt er, at PSP kræver stor selvdisciplin og motivation hos udviklerne. [PUG97] peger netop på dette som det største problem ved PSP. Dette skyldes i høj



Figur 3-2: PSP struktur.

grad den store mængde databehandling, der er involveret, og som ikke bruges til noget før senere i processen. Det betyder, at det kan være en tiltrækkende mulighed at optimere processen lidt ved at skære de dele væk, der umiddelbart virker unødvendige, især hvis tiden er presset.

PSP lægger op til, at procesejerskabet skal flyttes ned til de enkelte udviklere i organisationen. Ved hjælp af PSP skal udvikleren blive i stand til at identificere problemerne i sin udviklingsproces og på baggrund af disse komme med procesforbedringsforslag. Derved motiverer PSP at læringen foregår hos den enkelte udvikler gennem indsigt i hvilke områder, der kan forbedres.

Opbygningen af forbedringsmetoden PSP dikterer i høj grad hvilke områder, der skal forbedres og hvilken rækkefølge, de skal forbedres i. Derfor betragter vi et SPI forløb med udgangspunkt i PSP som planlagt, da det er muligt at forudsige, hvad udvikleren skal forsøge at forbedre i fremtiden.

Fokus i PSP ligger på processen i kraft af PSPs opbygning. De processcripts, som PSP består af, fokuserer på at beskrive hvordan udvikleren skal udføre sit arbejde. Udvikleren skal også sideløbende med det rigtige arbejde foretage dataopsamlingen og udregne forskellige statistikker og nøgletal. Alt dette er med til at flytte fokus fra den egentlige udvikling over på processerne. Desuden kan den omfattende registrering efter vores mening betyde, at PSP bliver opfattet som værende for bureaukratisk. Processcriptene fungerer som prædefinerede procesbeskrivelser, som efter udfyldelse udgør udviklerens proces. Når udvikleren er nået igennem PSP modellen har vedkommende en defineret proces for, hvordan software skal udvikles. Herved lægger PSP op til, at procesbeskrivelserne konstrueres på et bestemt tidspunkt og fremtidigt arbejde tager udgangspunkt i beskrivelserne. I tabel 3-3 er PSP holdt op imod de otte egenskaber.

3.4 Team Software Process

Team Software Process er Humphreys [Hum00b] bud på, hvordan PSP kan løftes op til at dække grupper i stedet for individer og sigter mod at hjælpe grupper af udviklere med konsistent at lave

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Individet indser problemerne i udviklingsprocessen.
<i>Procesejerskab</i>	Individet har ansvaret for processen og kommer med forbedringer.
<i>Planlægning</i>	SPI indsatsen er planlagt.
<i>Design</i>	Design af proces.
<i>Dokumentation</i>	Konstrueres på et bestemt tidspunkt.
<i>Normer</i>	Normer er vigtige, da PSP bygger på samme tankegang som CMM.
<i>Fokus</i>	Fokus på proces.
<i>Læring</i>	Double-loop læring hos det enkelte individ.

Tabel 3-3: PSP i forhold til de otte egenskaber og deres kendetegn.

produkter af høj kvalitet. Lige som PSP prøver TSP at opnå dette gennem definerede processer og disciplineret arbejde, og derfor er det naturligt, at TSP ligger som en forlængelse af PSP, da de enkelte udviklere lærer disse kvalifikationer gennem PSP. De elementer TSP beskæftiger sig med er især projektplanlægning, samt hvad der kan påvirke en projektplan. Gennem TSP tvinges udviklerne til at lave realistiske estimater og på den baggrund opstille realistiske tidsplaner, som kan forsvares overfor ledelsen. Gennem jævnlige genplanlægningsmøder revideres projektplanen og arbejdsfordelingen, hvorved alle har noget at lave, og planen altid er konsistent med den faktiske udvikling.

Lige som med PSP er det også meningen, at grupperne gennem TSP laver en omfattende dataopsamling. Det drejer sig især om nøgletal omkring hvor lang tid, der er blevet brugt på de enkelte elementer, og hvordan den tid er i forhold til den estimerede tid.

3.4.1 Vurdering af TSP

Betydningen af TSP er endnu uafklaret. Meget peger på, at PSP er anvendeligt i mange tilfælde, og stort set alle undersøgelser har vist, at fejlraten falder kraftigt, samtidig med at estimaterne bliver mere præcise. Om denne trend fortsætter, når TSP indføres, er der ingen undersøgelser, der har vist noget om endnu. I [Got00] nævnes adskillige gode erfaringer med TSP, men igen er der ingen garanti for, at PSP alene ikke kan gøre det.

Svaghederne ved TSP er at på trods af, at der genplanlægges jævnligt spiller de indledende estimater stadig en meget stor rolle for den efterfølgende projektopfølgning, og mange nøgletal udregnes på baggrund af disse estimater. Problemet er, at hvis TSP følges, så er estimaterne lavet kun en dag efter, projektet er blevet startet op, og på det tidspunkt kan det ikke forventes, at udviklerne kender til alle detaljerne i projektet. Selvfølgelig er det aldrig muligt at kende til alt, men når estimater laves på så tidligt et tidspunkt, kan de kun i få tilfælde være særlig præcise.

TSP har lige som PSP et problem med den omfattende dataopsamling. Der er nogle enkelte værktøjer til at hjælpe, men der er stadig stor mangel på værktøjer, der er praktiske og lette at bruge uden, at det forstyrrer andet arbejde alt for meget, hvilket betyder, at TSP som PSP kan blive opfattet som for bureaukratisk [Got00].

Ifølge tabel 3-4 har TSP de samme kendetegn som PSP i forhold til de otte egenskaber. Den eneste forskel er at problemerkendelsen, procesejerskabet og læringen er løftet fra individ niveau til projektgruppe niveau.

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen indser problemerne i udviklingsprocessen.
<i>Procesejerskab</i>	Projektgruppen har ansvaret for processen og kommer med forbedringer.
<i>Planlægning</i>	SPI indsatsen er planlagt.
<i>Design</i>	Design af proces.
<i>Dokumentation</i>	Konstrueres på et bestemt tidspunkt.
<i>Normer</i>	Normer er vigtige, da TSP bygger på samme tankegang som CMM.
<i>Fokus</i>	Fokus på proces.
<i>Læring</i>	Double-loop læring hos projektgruppen.

Tabel 3-4: TSP i forhold til de otte egenskaber og deres kendetegn.

3.5 Letvægtsmetoder

Det sidste sted vi vil hente inspiration til en alternativ procesforbedringstilgang er fra softwareudvikling. Indenfor softwareudvikling er der dukket en række alternative udviklingsmetoder op - de såkaldte *letvægtsmetoder*². Disse metoder udspringer af en kritik imod de traditionelle udviklingsmetoder, som fx spiral- og vandfaldsmodellen, for at være for tunge, bureaukratiske og for lidt fokuserede på de mennesker, der skal anvende metoderne i praksis. Den mest kendte letvægtsmetode er nok *Extreme Programming*, men der eksisterer en hel del andre fx *Crystal Methodologies* og *SCRUM*; [Fow00] indeholder en kort oversigt over disse og andre metoder. I dette afsnit vil tankerne bag letvægtsmetoderne blive gennemgået, da vi mener, det er muligt at overføre nogle af ideerne til SPI, selvom der er tale om to forskellige domæner indenfor datalogien.

Letvægtsmetoderne følger en meget praktisk tilgang til softwareudvikling. Alle former for unødvendigt bureaukrati er skåret væk, og der er ingen processer, der følges for processernes skyld. Det betyder, at følgende dele er sat i centrum:

- Individ og interaktion fremfor processer og værktøjer.
- Fungerende software fremfor omfattende dokumentation.
- Samarbejde med kunder fremfor kontraktforhandlinger.
- Ændringsdygtighed fremfor at følge en fast plan.

Den del, der umiddelbart er mest tydelig er, at dokumentationsaspektet er nedtonet alvorligt, hvor det i stedet siges, at kildekoden er en vigtig del af dokumentationen. Det betyder dog ikke, at letvægtsmetoderne ikke tager dokumentation alvorligt, det er nærmere et udtryk for, at letvægtsmetoderne ikke prøver at planlægge alt forud. Modsat de tunge procesmodeller prøver letvægtsmetoderne hele tiden at tilpasse sig den nuværende situation og er dermed resistente overfor ændringer.

²Oversat fra det engelske *agile processes*.

Derudover er letvægtsprocesserne orienteret mod personer og ikke mod processer. Der lægges vægt på, at softwareudvikling bør være en aktivitet, der ikke påtvinges, men som udføres af lyst.

3.5.1 Forudsigelighed og tilpasning

Design har altid været en meget vigtig del af tunge procesmodeller. Dette skyldes, at disse modeller er baseret på langsigtede planer og faste kontrakter, og derfor er det nødvendigt at have alt klart og designet, før programmeringsarbejdet kan gå i gang. Her har letvægtsmetoderne en anden tilgang, for de siger, at design rent faktisk er en del af programmeringsarbejdet, og at det er forkert at splitte de to ting op, for uanset hvor godt et design er, og hvor meget der er udspecificeret, vil der altid dukke nye ting op under programmeringsfasen.

Alt dette skyldes en udbredt trang til at forudsige processernes forløb. Der kan ikke være nogen tvivl om, at hvis det kan lade sig gøre, er det det mest fordelagtige, da det gør det muligt at opstille detaljerede planer og indgå kontrakter med kunder på faste beløb. Dette kræver, at kravene til det program, der skal udvikles, ligger helt fast fra starten, hvilket kun meget sjældent er tilfældet, og det er også langt fra altid tilfældet, at kravene kan lægges helt fast gennem en udførlig designfase. Det skyldes, at det er stort set umuligt at sige noget om et produkts anvendelighed, før det rent faktisk har været i brug. Derudover kan kravene også ændre sig som følge af eksterne hændelser, for verden står ikke stille, mens udviklingen foregår, og en kundes behov kan ændre sig med markedsituationen efterhånden, som tiden går. Der er selvfølgelig nogle tilfælde, hvor det er muligt at forudsige hvilke krav, der stilles til et produkt, men det er oftest i forbindelse med militære projekter, hvor alle krav lægges fast i god tid.

Det problematiske er at bruge en metode, der prøver at være forudsigelig, i en situation, som ikke er mulig at forudsige. Det betyder ofte, at der udarbejdes en plan, som langsomt begynder at falde fra hinanden, men som alle alligevel prøver at følge, indtil det tidspunkt hvor planen er fuldstændig uholdbar, hvilket kan være en smertefuld oplevelse for alle involverede.

Et projekt uden en detaljeret plan er for mange lig med komplet ukontrolleret kaos, men det er her, letvægtsmetoderne går ind og giver kontrol over denne mangel på forudsigelighed. Det sker ved hjælp af iterativ udvikling, hvilket ikke er noget nyt fænomen, fx er spiralmodellen en velkendt model. Fordelen ved iterativ udvikling er, at det gør det muligt med jævne mellemrum at få en pålidelig indikation af, hvordan situationen ser ud. Det sker ved at levere fungerende versioner af det færdige produkt, som kun indeholder en delmængde af den krævede funktionalitet, men som bortset fra denne mangel på funktionalitet stadig lever op til kravene til det endelige system. Disse mellemliverancer skal være testede og integrerede, som det færdige system ville være, og derved kan fejl og mangler hurtigt identificeres, da systemet udsættes for slutbrugerne og deres forventninger. Denne iterative udvikling betyder, at langsigtede planer er meget vage, og at det kun er planen for den næste iteration, der er nogenlunde endelig.

Denne udviklingstilgang, der er baseret på løbende tilpasning, stiller også nogle andre krav til aftagerne af produkterne. Med de tunge metoder er det let, her skal der blot udstikkes nogle krav, indhentes tilbud, udvælges en leverandør, og så er det ellers leverandøren, der sidder med resten af ansvaret for at gøre produktet færdigt. Dette lægger normalt også op til en kontrakt med fast pris, men for at kunne estimere prisen er processen nødt til at være forudsigelig, hvilket igen er lig med stabile krav. Som allerede nævnt, er dette sjældent tilfældet, og ved at sætte fast pris på

en uforudsigelig proces løbes der en stor risiko. Hvis det går galt, og planen ikke overholdes, hvilket ofte sker, er det ikke bare leverandøren, der taber penge, det er også kunden, for kunden ville næppe have bestilt et system, hvis der ikke var brug for det. Derfor er kunden også nødt til at arbejde anderledes, når der er tale om en proces baseret på løbende tilpasning.

I en sådan proces har kunden meget større kontrol over udviklingen. Ved hver iteration er der mulighed for at komme med nye krav eller ændre på de eksisterende, og det er muligt at afprøve systemet, som det ser ud. Det betyder også, at et system kan komme i produktion tidligt i processen, selv om det ikke er helt komplet endnu.

3.5.2 Udviklerne i fokus

En af motivationsfaktorerne bag de mere traditionelle modeller er, at ved at beskrive processerne, bliver det også lettere at sætte nye udviklere ind i processerne. Dermed behandles udviklerne som udskiftelige komponenter, og det er ikke så meget udviklerne, men derimod de roller, de udfylder, der er vigtige. Ifølge Alistair Cockburn [Coc99], er det et fejlagtigt synspunkt. Han skriver, at for at lave forudsigelige processer skal processens komponenter være forudsigelige, hvilket ikke er tilfældet med mennesker. Han skriver også, at han gennem sine studier er kommet frem til den konklusion, at den vigtigste faktor i softwareudvikling ikke er processen, men derimod de mennesker, der skal stå for udviklingen. Ved at behandle udviklerne som udskiftelige dele bliver de ikke respekteret som individer, og det fører til lavere motivation og produktivitet, og de kreative udviklere søger til andre steder. Teorien om, at en separat afdeling, fx i form af en SPI gruppe placeret i teknostrukturen, kan planlægge kompetente menneskers tekniske arbejde afvises fuldstændigt i letvægtsmetoderne, for her er opfattelsen den, at dem, der bedst ved hvordan arbejdet udføres, er dem, der rent faktisk udfører arbejdet.

Det betyder, at de eneste, der kan vælge at følge en tilpassende proces, er udviklerne selv, da det er den eneste måde, hvorpå det er muligt at komme ud over modstanden over initiativer, der bliver presset ned ovenfra. Et andet centralt element er, at udviklerne skal være i stand til selv at tage alle tekniske beslutninger. Det kan kræve store ændringer i ledelsens opfattelse af, hvor ansvaret skal placeres, for med denne tilgang har ledelsen ikke længere magten over alt, men deler i stedet ansvaret ligeligt med udviklerne.

3.5.3 Vurdering af letvægtsmetoder

Letvægtsmetoderne har i længere tid været anset som useriøse og uden hold i virkeligheden, men det har efterhånden ændret sig. Umiddelbart har letvægtsmetoderne mange fordele, og virker meget tiltalende, men der er stadig problemer. Problemerne er dog af en anden karakter end dem, der præger PSP, TSP og lignende metoder. Hos dem er problemet bureaukrati og formalisering, hos letvægtsmetoderne er det det modsatte, nemlig at styringen i høj grad er overladt til udviklerne. Det kræver både evner og disciplin hos udviklerne at følge en letvægtsmetode, og lykkes det ikke, kan det føre til et alvorligt tillidsknæk overfor ledelsen, og troen på at letvægtsmetoder kan fungere forsvinder måske helt.

Er der tiltro til, at medarbejderne i en virksomhed kan arbejde selvstændigt og uden topstyring, er letvægtsmetoderne helt klart et alternativ, der bør overvejes. Det er i mange virksomheder urealistisk at implementere en letvægtsmetode fuldt ud, og helt gå væk fra de gamle, mere bureaukratiske metoder på grund af konkurrencehensyn og på grund af, at letvægtsmetoderne stadig ikke

er anerkendte alle steder. På trods af det kan nogle af principperne benyttes, fx at bureaukratiet skæres ned til et minimum, og at udviklerne kommer i centrum fremfor processerne. Det kan også være svært for ledelsen at undvære den følelse af kontrol, som andre metoder giver. Letvægtsmetoderne går ud fra, at meget få situationer kan forudsiges fuldt ud, og at det kan være meget farligt at anvende metoder, der bygger på forudsigelse, når det ikke er muligt at forudsige situationen.

Letvægtsmetoderne sætter ord på mange af de problemer, som traditionel procesforbedring har, og deler vores kritik af, at de grundlæggende antagelser i de traditionelle tilgange ganske enkelt er forkerte. Selv om letvægtsmetoderne ikke sigter direkte mod procesforbedring, men mere taler om at udvikle software, er de grundlæggende antagelser væsentlig anderledes end dem, der ligger bag traditionelle SPI tilgange. Udviklingen af processer er stort set det samme som udvikling af software. Det er ikke muligt at designe processerne fuldstændigt, og det er nødvendigt for procesforbedringen at være klar til, at der kan ske ændringer undervejs.

3.6 Vurdering af alternativ SPI

I dette kapitel har vi præsenteret to artikler, der kritisk forholder sig til den traditionelle tilgang til SPI. Deres synspunkter er, at softwareindustrien bør ændre fokus i forbindelse med SPI og fokusere mere på udviklerne end på processerne. I stedet for at adskille udviklingen af procesmodeller fra den faktiske anvendelse, bør disse aktiviteter samles hos praktikerne. For at gøre dette foreslås det, at SPI indsatsen flyttes til praktikerne i organisationerne, så de selv får ansvaret for processerne og for selv at erkende problemerne i deres udviklingsproces. Herved sikres det, at læringen i organisationen foregår hos praktikerne og ikke i baglokalet hos processpecialisterne. Desuden foreslås det, at der ikke fokuseres på design af processer og planlægning af SPI forløbet, men i stedet fokuseres på etablering af praksis, og at der tages udgangspunkt i konkrete problemstillinger, hvorved SPI forløbet får et større præg af improvisation. Sidst men ikke mindst, er SPI en løbende aktivitet, og organisationens udviklingsprocesser skal løbende tages op til revurdering. Gennem PSP og TSP har vi set konkrete eksempler på metoder, der sætter større fokus på den enkelte praktiker og projektgruppe, og som med succes har været afprøvet i praksis.

Selvom letvægtsmetoderne kommer fra softwareudvikling, mener vi, at en række elementer kan overføres til SPI. Dette er synspunktet om, at individet skal sættes i fokus frem for processer og værktøjer, at fungerende software er mere væsentligt end en omfattende dokumentation, og at organisationen skal være i stand til at omstille sig fremfor at følge en fast plan.

I det følgende kapitel vil vi præsentere vores forslag til, hvordan en konkret forbedringsmodel til SPI kan opbygges. Baggrunden for at vi har udviklet en alternativ procesmodel er, at selvom Bach og Aaen præsenterer en masse tanker omkring alternativ procesforbedring, kommer de ikke med et konkret bud på, hvordan denne procesforbedring skal udføres i praksis. For at indføre Bach og Aaens tanker mener vi ikke, at metoder som PSP og TSP finder anvendelse, da de er for fokuserede på en foruddefineret proces og efterfølgende dataopsamling og ikke så meget på etablering af praksis. Desuden lægger de to metoder op til, at praktikerne eller projektgruppen skal indse problemerne ved udviklingsprocessen og ikke til, at de skal erkende problemerne gennem refleksion. Dette skyldes at metoderne påpeger, hvor problemerne er.

I dette kapitel vil vi præsentere en alternativ SPI tilgang, som vi har valgt at fundere i begrebet *selvforbedrende projektgrupper*. I det følgende vil de hovedideer og tankegange, der lægger bag begrebet blive præsenteret gennem en relatering til traditionel og alternativ SPI. Desuden vil vi illustre ideerne til en forbedringsmodel, hvorved selvforbedring kan igangsættes i en organisations projektgrupper. De indledende ideer, tankegange og metoden danner grundlaget for interventionen i Logimatic, hvor de vil blive forsøgt overført til praksis. Efterfølgende vil vi vende tilbage til vores foreslåede tilgang med henblik på en evaluering.

4.1 Egenskaber ved selvforbedrende projektgrupper

En selvforbedrende projektgruppe er kort fortalt en gruppe, der tager udgangspunkt i de allerede etablerede projektgrupper i organisationen, og som har interesse for, evnen til og ansvaret for at forbedre dens eksisterende udviklingsproces. I det følgende vil der udfra de otte egenskaber, som vi har holdt den traditionelle og alternative SPI tilgang op imod blive givet en karakteristik af selvforbedrende projektgrupper:

Problemerkendelse

En væsentligt forskel mellem den traditionelle og den alternative SPI er, at der ved den traditionelle tilgang er en adskillelse af procesdesign og procesanvendelse. Dette kommer bl.a. til udtryk ved, hvor i organisationen problemerkendelsen foregår. Ved den traditionelle tilgang sker det i SPI gruppen enten på baggrund af gruppens viden og erfaring om udviklingen i organisationen eller på baggrund af målinger i forhold til normer som fx CMM. Dette nævner både Bach og Aaen som et problem, og de ønsker derfor, at problemerkendelsen flyttes til dem, som skal anvende processen i praksis.

Ved selvforbedrende projektgrupper mener vi også, at det er væsentligt, at problemerkendelsen sker hos praktikerne. Det kan være med til at mindske sandsynligheden for fremmedgørelse overfor procesforbedringen, da de problemer, der bliver fokuseret på, anses som relevante af praktikerne. Desuden kan det være med til at sikre en større motivation i forbindelse med SPI, da projektgrupperne bliver en aktiv del af indsatsen.

Procesejerskab

I den traditionelle SPI ligger procesejerskabet hos SPI gruppen, hvilket anses som et problem hos både Bach og Aaen. De ønsker i stedet at flytte ejerskabet ud til de enkelte projektgrupper. Det samme gør PSP og TSP, men dette sker udfra en antagelse om, at forbedringerne skal ske på andre niveauer i organisationen, og ikke fordi det er et problem, at procesejerskabet ikke ligger

hos praktikerne. Ved at flytte procesejerskabet er det praktikerne, der udarbejder de løsninger, som de ønsker at anvende for at løse de problemer, som de i øvrigt selv har erkendt.

Netop flytningen af procesejerskabet fra SPI gruppen til den enkelte projektgruppe er en af hovedideerne bag selvforbedrende projektgrupper. Dette er efter vores mening med til at sikre en mindre topstyring af SPI indsatsen, da ejerskabet flyttes fra det taktiske niveau til det operationelle niveau, dvs. fra mellemliderne og teknostrukturen til praktikerne. Som ved problemerkendelsen mener vi, at flytningen af procesejerskabet kan være en medvirkende faktor til at mindske fremmedgørelse blandt praktikerne overfor SPI indsatsen. Ved at det er praktikerne, der selv er med til at udarbejde løsningerne, vil de også have nemmere ved at kunne relatere sig til dem samt vide hvilket ræsonnement der ligger bag dem.

Planlægning

Til forskel fra den traditionelle SPI tilgang, hvor SPI forløbet planlægges detaljeret mener både Bach og Aaen, at SPI forløbet godt må bære præg af at være improviseret. Det betyder, at i stedet for at SPI gruppen sidder og planlægger hvilke initiativer, der skal indføres i fremtiden, og hvordan de skal indføres, er det de enkelte projekter, som tager problemerne op alt efter hvilke problemer, de står med i den nuværende situation. Dette synspunkt kan også overføres fra letvægtsmetoderne, hvor der tales om, at det ikke er muligt at planlægge langt ud i fremtiden.

At SPI forløbet er improviseret er en egenskab som vi også finder anvendelig hos selvforbedrende projektgrupper. Da det er den enkelte projektgruppe, der selv erkender problemerne og kommer med løsningen, kan det være svært at lave en overordnet organisatorisk planlægning af hvilke områder, der skal forbedres. Dette skyldes, at det nok ikke er de samme problemer, som organisationens projektgrupper sidder med, og at projektgruppen ofte nedsættes ad-hoc og opløses efter, at gruppens mål er opfyldt.

Design

Traditionel SPI lægger stor vægt på formalisering og design af procesbeskrivelser med mere fokus på ledelsens end praktikernes behov. Dette gør, at designet af procesbeskrivelserne anses som det væsentligste element i SPI og ikke indførelsen af en bedre praksis. I den alternativ SPI tilgang sættes etablering af praksis til gengæld før udarbejdelse af procesbeskrivelserne. Herved er det praksis, der er i fokus. Desuden mener hverken Bacj eller Aaen, at procesbeskrivelser skal designes, de skal i stedet afspejle den faktiske praksis i organisationen.

Netop etableringen af praksis er et af de områder, vi mener, letvægtsmetodernes principper om at fokusere på fungerende software fremfor omfattende dokumentation kan overføres til SPI. Ved SPI bliver ræsonnementet i stedet, at der skal fokuseres på fungerende praksis fremfor omfattende dokumentation af praksis. Dette synspunkt understøttes af både Aaen og Bach, der i deres artikler skriver, at de finder etablering af praksis væsentligst. Ved at fokusere på etablering, anvendes der ikke unødigt tid på at designe processer, som ikke finder anvendelse i virkeligheden. Desuden gør det også, at der sendes et signal til praktikerne om, at procesforbedring drejer sig om at forbedre praksis, og ikke blot om at få nedskrevet processerne for at sikre normopfyldelse. Vi mener dog stadig, at det er vigtigt, at praksis nedskrives, hvorved gruppen kan dokumentere dens udviklingsproces og så eksempler på praksis kan gøres tilgængelige for resten af organisationen.

Dokumentation

Ved den traditionelle tilgang til SPI foregår dokumentation og derved konstruktion af procesbeskrivelserne på ét bestemt tidspunkt. Dokumentationen sker ved, at en procesbeskrivelse designes og derefter udrulles i organisationen. I forbindelse med udrulningen kan der ske en tilpasning af beskrivelsen, men når iterationen i IDEAL modellen er færdig, foreligger der en færdig procesbeskrivelse, som dokumenterer den anvendte praksis i organisationen. Herefter kan der gå lang tid, inden procesbeskrivelsen ændres igen. En sådan ændring kan ske på baggrund af en ændret praksis, eller fordi procesdesignerne får indsigt i nye måder at udføre processen på. Både PSP og TSP anlægger det samme syn på dokumentation som den traditionelle SPI, hvorimod End-user SPI lægger op til, at procesbeskrivelserne skal rekonstrueres løbende.

Vi mener ligesom ved End-user SPI, at ved selvforbedrende projektgrupper skal procesbeskrivelserne løbende rekonstrueres. Baggrunden for dette er, at vi mener, det er væsentligt, at dokumentationen af processen afspejler den faktiske proces, som anvendes i projektgruppen, ellers opfylder procesbeskrivelserne ikke deres formål. Desuden gør en løbende rekonstruering, at det hele tiden overvejes, om den anvendte proces er den mest optimale og hensigtsmæssige for projektgruppen. Herved kan der opnås en løbende tilpasning med små ændringer, som hele tiden forbedrer processen.

Normer

Normer er en væsentlig del af traditionel SPI, da normer sætter fokus på hvilke områder, der skal forbedres. Hvorvidt normer er en væsentlig del af SPI, er de alternative tilgang ikke enige om. Bach afviser brugen af normer med henvisning til, at det kan være direkte farligt for forbedringsindsatsen. Baggrunden for dette er ifølge Bach, at normer kan være med til at fremmedgøre praktikerne overfor SPI indsatsen. PSP og TSP bygger derimod på de samme tanker som CMM og er i princippet tænkt til at skulle gøre procesforbedring i forhold til CMM lettere. Derfor er der i disse to tilgange lagt op til, at normer er nødvendige for SPI. Aaen mener også, at normer er nødvendige for SPI, dette er dog udfra ræsonnementet om, at normerne er med til at få etableret en dialog omkring forbedring af processer, rationalet bag processerne og etablering af en fælles praksis på tværs af projektgrupper.

Hvorvidt normer er nødvendige for selvforbedrende projektgrupper er et godt spørgsmål. Som ved End-user SPI mener vi, at normer muligvis kan være en nødvendighed for at opnå en fælles praksis i organisationen. På den anden side kan brugen af normer også være et problem, da de overfor den selvforbedrende projektgruppe dikterer hvilke problemer, der er i deres udviklingsproces. Herved bliver fokus flyttet fra, at det er projektgruppen selv, som skal erkende problemerne til, at det nu er de personer, som har udviklet normerne, der dikterer problemerne i gruppens udviklingsproces. Desuden kan en fokusering på normer betyde, at den selvforbedrende projektgruppe ikke opdager de faktiske problemer i deres udviklingsproces, da de stirrer sig blind på de områder, normerne dikterer. Normerne kan dog også hjælpe den selvforbedrende projektgruppe, hvis den ikke er i stand til at identificere nogle problemer i udviklingsprocessen. Her kan såkaldte lette vurderingsmetoder anvendes til at hjælpe projektgruppen med at indse potentielle områder for en forbedringsindsats. Desuden er normerne ofte et væsentligt aspekt af procesforbedring i industrien, da forbedringsindsatsen sigter mod at give organisationerne et kvalitetsstempel i form af fx en ISO eller CMM

certificering. Derfor vil en SPI indsats, som ikke tager højde for normerne sikkert ikke have de store anvendelsesmuligheder i praksis. Vi mener dog, at problematikken omkring normer ligger udenfor projektets rammer, da vi gennem dette projekt fokuserer på afprøvelse af bottom-up SPI i praksis.

Fokus

I den traditionelle SPI ligger fokus på at få designet og formaliseret processer, som sikrer opfyldelse af organisationens mål. Efterfølgende skal disse processer udrulles, så organisationens praktikere kan anvende dem. Ved den alternative SPI, bortset fra TSP og PSP, bliver dette vendt på hovedet. I stedet for at fokusere på processerne bliver der fokuseret på udviklerne. Dvs. for alternativ SPI er det primære at få etableret en god praksis hos udviklerne. Her kan princippet fra letvægtsmetoder om, at der skal udvikles fungerende software fremfor omfattende dokumentation, også overføres til SPI.

Som både Bach og Aaen, mener vi også, det er væsentligt, at fokus bliver flyttet fra processerne og hen på praktikerne. Denne fokusering er endnu et element fra letvægtsmetoderne, som vi mener kan overføres til SPI. Ved at fokusere mere på udviklerne og hvad de anser som problemer i udviklingsprocessen, bliver de en del af SPI indsatsen. Derved undgås det at procesdesignerne sidder og designer processer, som ikke tager højde for den enkelte medarbejders kompetencer, men i stedet betragter medarbejderen som en udskiftelig komponent. Som Cockburn [Coc99] skriver, så bliver udviklerne ikke respekteret som individer, hvis de behandles som udskiftelige dele, og det kan føre til lavere motivation og produktivitet. For at undgå dette er det væsentligt, at der bliver fokuseret på udviklingen.

Læring

Adskillelsen af design fra praksis medfører i traditionel SPI, at læringen foregår to forskellige steder i organisationen. SPI gruppen foretager double-loop læring, hvorimod praktikerne kun foretager single-loop læring. Vi mener det er væsentligt, at læringen samles hos praktikerne. Uden den nødvendige indsigt i hvorfor nye processer foreslås, mindskes sandsynligheden for, at de nye processer bliver anvendt, og at der dermed sker ændringer i praktikernes adfærd.

For selvforbedrende projektgrupper er det derfor væsentligt, at de besidder evnen og viljen til at foretage double-loop læring. I kraft af at gruppen, som før nævnt, sidder med ansvaret for problemerkendelse og procesejerskabet, skal gruppen derfor være i stand til gennem diskussion at opnå konsensus omkring problemer ved udviklingsprocessen, komme med løsningsforslag og overføre forslagene til praksis.

Ovenstående otte egenskaber og deres kendetegn for selvforbedrende projektgrupper er kort opsummeret i tabel 4-1.

4.2 Overvejelser om indførelse

I det foregående afsnit har vi præsenteret de grundlæggende tanker og ideer bag selvforbedrende projektgrupper. Spørgsmålet er, hvordan ideerne kan indføres i praksis. Udfra de kendetegn ved selvforbedrende projektgrupper, som vi har beskrevet, har vi to hovedområder for indførelse af

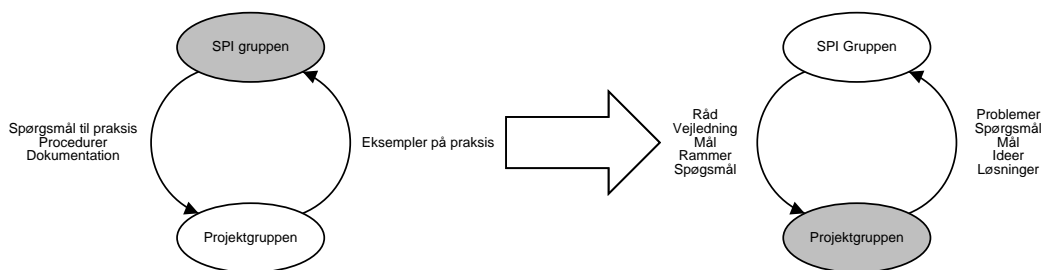
Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen erkender problemerne ved deres udviklingsproces.
<i>Procesejerskab</i>	Ansvar for processen ligger hos projektgruppen.
<i>Planlægning</i>	SPI forløbet improviseres.
<i>Design</i>	Primært fokus på etablering af praksis, derefter modelleres resultatet.
<i>Dokumentation</i>	Procesbeskrivelser skal rekonstrueres løbende.
<i>Normer</i>	-
<i>Fokus</i>	Fokus på udvikling.
<i>Læring</i>	Double-loop læring hos projektgruppen.

Tabel 4-1: Selvforbedrende projektgrupper i forhold til de otte egenskaber og deres kendetegn.

selvforbedrende projektgrupper: 1) Samling af udvikling og anvendelse af processerne og 2) fokusering på etablering af praksis. I det følgende vil vi komme med et bud på, hvordan dette kan ske i praksis.

4.2.1 Samling af udvikling og anvendelse af processer

En væsentlig forskel mellem traditionel SPI og selvforbedrende projektgrupper er, at der ved selvforbedrende projektgrupper ikke er en adskillelse af udvikling af processer og deres anvendelse i praksis. Derfor skal ansvaret for problemerkendelsen og procesejerskabet flyttes fra SPI gruppen til den selvforbedrende projektgruppe, hvilket betyder at der skal foretages en omdefinering af parternes indbyrdes rolle. Indførelse af en ny rollefordeling skal være med til at gøre parterne opmærksom på, hvad de kan forvente af hinanden. På figur 4-1 fremgår et forslag til, hvordan den nye rollefordeling mellem projektgruppen og SPI gruppen kan være. Det væsentligste er, at hvor det traditionelt er SPI gruppen, der er drivkraften bag forbedringsindsatsen og bestemmer hvor, hvornår og hvad der skal forbedres, er det nu den enkelte projektgruppes ansvar.



Figur 4-1: Den nye rollefordeling mellem den selvforbedrende projektgruppe og SPI gruppen når drivkraften flyttes. Den markerede cirkel illustrerer, hvor drivkraften ligger.

Ved den traditionelle SPI stiller SPI gruppen spørgsmål til projektgrupperne for at forstå deres

praksis. På baggrund af eksempler på udviklingsprocesser identificerer SPI gruppen problemerne ved processen, og der udvikles nye procedurer og nye dokumenterede processer, som projektgruppen skal følge. Ved at flytte drivkraften fra SPI gruppen til projektgrupperne får projektgruppen ansvaret for at identificere problemerne og komme med løsninger på problemerne. SPI gruppens rolle ændres til at være sparringspartner for projektgruppen, som kan henvende sig til SPI gruppen med de problemer, de ser i deres udviklingsproces, spørgsmål omkring procesforbedringen og ideer til procesforbedring med henblik på at få en vurdering af ideerne. SPI gruppens rolle ændres hermed til, at gruppen i stedet har en vejledende rolle overfor projektgruppen. Gruppen skal primært opstille rammerne, fx for ressourceforbrug, for projektgruppens forbedringsinitiativer. Gruppen skal også kunne komme med råd og forslag til projektgruppens problemer på baggrund af tidligere erfaringer eller teori indenfor det valgte forbedringsområde. At komme med råd og forslag til projektgrupperne kan være med til at sikre udbredelse af andre projektgruppers tidligere erfaringer i organisationen. Det udspringer af, at SPI gruppens råd og forslag kan stamme fra gruppens erfaringer med succesfulde forbedringsinitiativer andre steder i organisationen. Herved er SPI gruppen med til at formidle andre projektgruppers erfaringer videre. Derfor er det væsentligt, at SPI gruppen er med til at sikre, at projektgrupperne får dokumenteret og nedskrevet deres erfaringer fra forbedringsindsatserne. Dette skal være med til at sikre, at gruppens viden kan benyttes andre steder i organisationen, hvis andre grupper støder på samme problemer.

Desuden kan denne nye rolle for SPI gruppen også være med til at sikre opfyldelse af organisationens overordnede målsætning, som fx modning i forhold til CMM, da gruppen kan give råd, der retter sig imod, hvordan projektgrupperne kan opfylde CMMs mål. SPI gruppen skal også hjælpe grupperne med at få opstillet mål for forbedringsindsatsen. Dette skal igen være med til at sikre organisationens overordnede målsætning. Det kan også være med til at sikre, at grupperne opfylder nogle minimumskrav, der er gældende i organisationen. Eksempelvis har de fleste organisationer krav til, hvad der skal dokumenteres, og hvordan dokumentationen skal udformes.

Ved at samle udvikling og anvendelse af processer, gøres der et forsøg på at få double-loop læringen til at foregå i den selvforbedrende projektgruppe i stedet for i SPI gruppen. Dette sker ved, at de enkelte medlemmer af projektgruppen tvinges til at reflektere over deres egen proces for at forsøge at identificere problemer, forklare hvorfor problemerne er der, udarbejde en løsning på problemerne og til sidst føre løsningen ud i praksis. For at motivere denne refleksion mener vi, det er væsentligt, at der sættes spørgsmålstejn ved om den måde, hvorpå processen udføres, er optimal. Hvis en udefrakommende person, fx et medlem af SPI gruppen, begynder at sætte spørgsmålstejn ved projektgruppens arbejdsproces, kan det være med til at fremme projektgruppens egen refleksion. Ved refleksionen skal projektgruppen kunne erkende et eller flere problemer ved deres nuværende praksis. Disse problemer skal de være istand til at formidle til tredjepart, fx SPI gruppen, for eventuelt at kunne søge råd og vejledning til at finde løsninger på problemerne. Desuden skal gruppen kunne udarbejde et løsningsforslag for, hvordan problemerne løses i praksis. Det kan ske ud fra deres erfaring med software udvikling, fra teori indenfor problemets område eller på baggrund af råd og forslag fra kollegaer. Gruppen skal kunne overføre strategien til praksis og være i stand til at kunne reflektere over om processen blev forbedret ved at ændre praksis.

Selvom ovenstående medfører, at det bliver svært at planlægge organisationens SPI indsats, da det er de enkelte projektgrupper, som bestemmer, hvor der skal procesforbedres, er der helt klart den fordel, at fokus flyttes fra processerne til udviklerne. Udviklerne bliver en aktiv del af SPI

indsatsen, hvilket efter vores mening vil øge sandsynligheden for, at forbedringerne følges i projektgrupperne, da de selv har været med til at igangsætte dem.

4.2.2 Fokusering på etablering af praksis

Inspireret af letvægtsmetoder er kravet til dokumentation i løbet af forbedringsprocessen minimalt. Hvor der ved den traditionelle tilgang er fokus på at få defineret, hvordan organisationen ønsker dens udviklingsproces skal se ud, er det ved selvforbedrene projektgrupper mere væsentligt at få etableret en praksis først. Det vi siger er, at der først skal laves en fungerende proces, og først herefter skal den dokumenteres, hvis det er nødvendigt. Dette er udfra ideen om, at det ikke er muligt at designe sig til bedre processer. Hvis der ønskes bedre processer, er det nødvendigt at ændre den faktiske udførelsen og ikke blot beskrivelsen af processen.

For eksempel er det ikke meningen, at de problemer, der skal beskrives, skal beskrives med flere siders tekst. Et par punkter er rigeligt, så længe der er en fælles forståelse for, hvad de går ud på. På samme måde skal strategien heller ikke være en detaljeret plan for, hvordan målene skal opnås, det skal nærmere være en liste med nogle punkter, eller et diagram, der viser, hvordan målet tænkes opfyldt. Der er på den måde ingen direkte krav til udformningen, det eneste umiddelbare krav er, at gruppen er enig om, hvad det er, der skal foregå, og kan forklare det til SPI gruppen. Det er også med til at flytte opmærksomheden væk fra beskrivelsen af strategien til selve udførelsen af strategien. Alt efter hvad en organisation er vant til, kan dette varieres efter behov. Hvis udformningen er nogenlunde ens på tværs af projekterne, er det lettere for SPI gruppen at samle op og sammenligne projekter og identificere styrker og svagheder ved de enkelte projekters strategier. Hvis der i organisationen er tradition for at udforme planer på en konsistent måde, er der ikke noget i vejen for at køre videre med det.

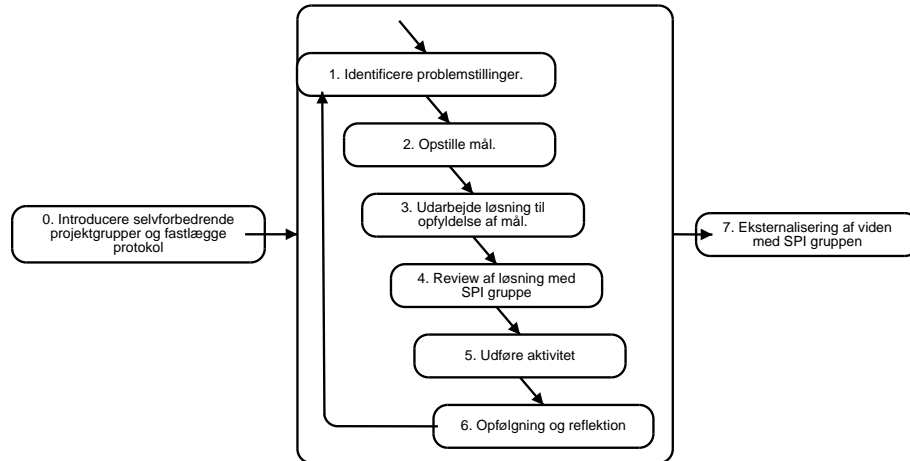
4.3 En alternativ forbedringsmodel

Med udgangspunkt i vores ideer til, hvad der kendetegner en selvforbedrende projektgruppe, samt vores overvejelser om, hvordan selvforbedring kan indføres i en organisation, vil der i det følgende blive opstillet et konkret forslag til, hvordan selvforbedring kan igangsættes i praksis. Dette forslag er illustreret på figur 4-2 og tager form af en forbedringsmodel, der er et alternativ til allerede etablerede modeller som fx IDEAL modellen.

Vores forslag til en alternativ forbedringsmodel baserer sig i høj grad på tanken om, at projektgruppen selv skal have indflydelse på, hvad der anses som problemer, og på at det er projektgruppen selv, der skal løse disse problemer. Selvom modellen på overfladen kan minde meget om IDEAL modellen, er de grundlæggende antagelser fundamentalt anderledes. Modellen lægger sig op af det syn James Bach præsenterer i [Bac98] med, at forbedringer sker bedst i små iterationer, hvor der tages fat på et problem ad gangen, uden at det er alle problemer i organisationen, der forsøges løst på én gang. Derudover ligger ansvaret for forbedringerne hos den enkelte projektgruppe, hvilket som sagt skulle være medvirkende til, at forbedringsinitiativerne bliver nemmere at forholde sig til set i forhold til den traditionelle tilgang.

Da forbedringsmodellen skal afprøves i praksis i Logimatic, har vi også inddraget vores erfaringer herfra i opbygningen af modellen. Blandt andet har medarbejderne vist, at de har interesse for at

forbedre sig, både med hensyn til personlige interesser og ambitioner. Desuden vurderer vi, at det faktum at forbedringsmodellen er mindre formel end fx IDEAL modellen vil gøre, at modellen passer godt ind i Logimatics uformelle miljø. Det bør også være muligt, at få Logimatics medarbejdere til at foretage selvforbedring, da medarbejderstaben er veluddannet og vant til at træffe beslutninger i kraft af, at beslutningskompetencen i Logimatic er både horisontalt og vertikalt decentraliseret. Derfor mener vi, at denne forbedringsmodel passer godt til Logimatic.



Figur 4-2: Forslag til en model til indførelse af selvforbedring.

En væsentlig forskel ved denne tilgang, som det er vigtigt at være opmærksom på, er, at det ikke er et forsøg på at designe forbedringerne, lige som fx IDEAL modellen er et forsøg på at designe forbedringer. I stedet er det et forsøg på, ved at tage vigtige elementer fra letvægtsmetoderne ind, at sætte nogle rammer op, og skabe et miljø, hvori forbedringer kan finde sted, og hvor initiativet til forbedringerne kommer fra projektgrupperne selv. På samme tid sikres det, at SPI gruppen bliver aflastet, så den i stedet for at være den drivende kraft i SPI projekter går over til at have en mere vejledende rolle, hvilket vi ser som en fordel for alle i organisationen. Derfor skal forbedringsmodellen ikke ses som en slavisk fremgangsmåde, men nærmere som vores syn på, hvordan det kan foregå.

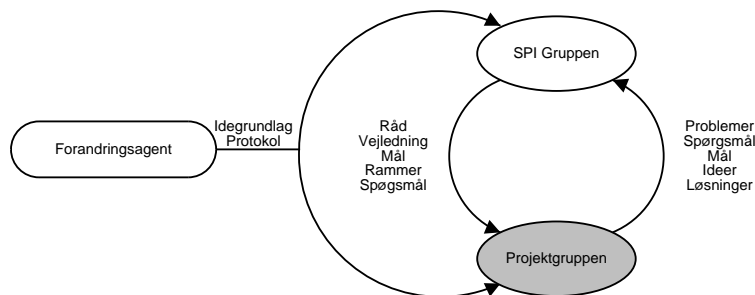
Ved anvendelse af forbedringsmodellen er ideen, at de otte aktiviteter som modellen består af bliver fordelt på separate møder. Dette giver projektgruppens medlemmer og SPI gruppen tid til at tænke over tingene mellem aktiviteterne. I det følgende vil hver af aktiviteterne blive gennemgået med henblik på at forklare, hvad den enkelte aktivitet dækker over.

4.3.1 Forbedringsmodellens aktiviteter

0. Introducere selvforbedrende projektgrupper og fastlægge protokol

Processen starter med, at projektgruppen og SPI gruppen får en introduktion til begrebet selvforbedrende projektgrupper, og den nye rollefordeling mellem projektgruppen og SPI gruppen fastlægges. Dette skal være med til at få fastlagt protokollen for deltagernes interaktion, som er afbilledet på figur 4-3. Vores rolle er at gå ind som forandringsagenter og ændre idegrundlaget hos projektgruppen og SPI gruppen, samt fastlægge rollefordelingen. Det gør, at alle har en fælles

forståelse for, hvad der skal ske, og at der ikke opstår misforståelser med hensyn til hvem, der står for hvilke aktiviteter. Det første trin sker kun en gang pr. projektgruppe, og er i princippet SPI gruppens opgave at klare, men i dette tilfælde er det os, der står for introduktionen, da konceptet er nyt for alle parter.



Figur 4-3: Protokol for projekt- og SPI-gruppen.

1. Identificere problemstillinger

Herefter kan den egentlige forbedringsindsats påbegyndes. Det sker ved at projektgruppen først identificerer en eller flere problemstillinger, som den synes er relevante at arbejde videre med. Her er det vigtigt, at problemerne ikke er alt for store, da en iteration ikke bør tage meget mere end 1-2 måneder og gerne mindre. Viser det sig, at problemerne vil tage meget længere tid brydes de ned i mindre delproblemer, der kan varetages enkeltvis. For at motivere refleksionen kan SPI gruppen evt. hjælpe til ved at stille kritiske spørgsmål til gruppens udviklingsproces.

2. Opstille mål

Ud fra det valgte problem opstilles mål for forbedringsinitiativet. Dette sker i samarbejde med SPI gruppen, som her har mulighed for at rette projektgruppens indsats ind mod virksomhedens overordnede mål, fx i forhold til CMM. Det er dog stadig vigtigt, at projektgruppen selv kan identificere sig med de opstillede mål for at sikre motivationen, men SPI gruppen kan stadig komme med input til målene.

3. Udarbejde løsning til opfyldelse af mål

Endeligt udarbejdes et løsningsforslag for, hvordan målene skal opfyldes. Eftersom dette stort set kun er til internt brug i gruppen, og kan nå at blive ændret mange gange endnu, er det ikke nødvendigt at dokumentere denne løsning i større omfang. Faktisk er det bedst at holde det på en enkelt side med noget tekst eller nogle tegninger. Formålet med at nedskrive det er at sikre, at alle medlemmer af gruppen er enige om, hvad der skal ske, og eftersom grupperne er forholdsvis små, må det forventes, at denne enighed kan opnås gennem dialog og gensidig tilpasning.

4. Review af løsning med SPI gruppen

Efterfølgende skal løsningsforslaget til review hos SPI gruppen for at sikre, at målene kan opfyldes på en realistisk måde, og uden at det går ud over produktiviteten, så der er brug for at kunne forklare SPI gruppen hvad forslaget går ud på. Dette kan igen være mundtligt, og omfattende dokumentation er derfor heller ikke nødvendig her.

5. Udføre aktivitet

Her skal det udarbejdede løsningsforslag for forbedring af aktiviteten anvendes i praksis. Derved

får projektgruppen erfaring med at anvende den forbedrede arbejdsproces, og de får et grundlag for efterfølgende af kunne reflektere over løsningens succes og anvendelighed.

6. Opfølgning og refleksion

Når den aktivitet, der skulle forbedres, er blevet gennemført følges der op på forløbet, og projektgruppen reflekterer over, hvad der gik godt, og hvad der gik skidt. Hvis målene er blevet opfyldt kan en ny iteration påbegyndes med et nyt problem, og ellers er det muligt at springe tilbage til et af de andre punkter. Hvis det fx viser sig, at målene er urealistiske eller irrelevante, kan gruppen gå tilbage til at opstille mål, og så fortsætte derfra.

7. Eksternalisering af viden med SPI gruppen

Ved et projekts afslutning kan SPI gruppen lave yderligere erfaringsopsamling med henblik på at eksternalisere noget af den viden, der er blevet opbygget gennem forbedringerne i den enkelte gruppe. Det kan fx ske ved at udforme best practices på baggrund af projektgruppernes arbejde, eller at notere hvilke problemer projektgruppen har løst, og formidle kontakten til andre projektgrupper, der har lignende problemer.

4.4 Opsummering

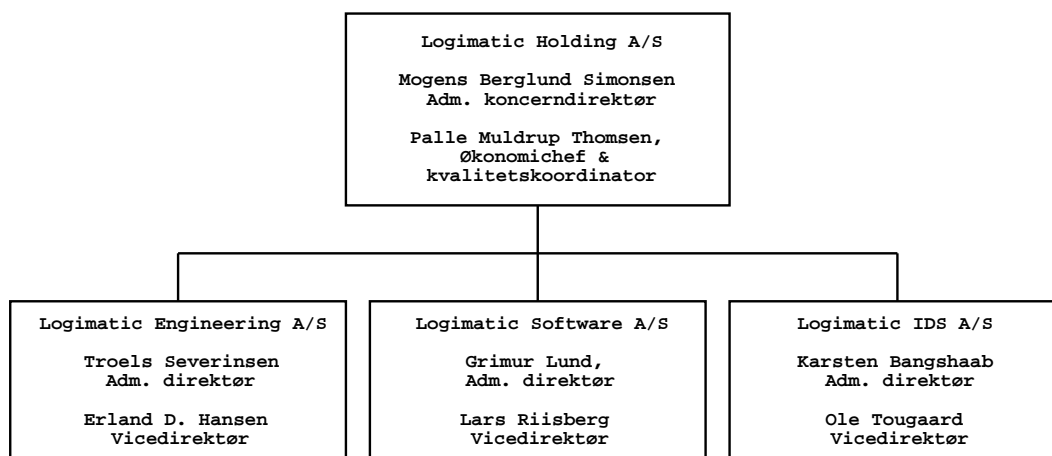
I dette kapitel er ideerne og tankerne bag begrebet selvforbedrende grupper blevet præsenteret. Denne type projektgruppe er kendetegnet ved, at den besidder procesejerskabet i forbindelse med procesforbedring, gruppen har evnen og viljen til at reflektere over dens udviklingsproces og forbedre denne, og gruppen har fokus på at få etableret en praksis, hvor effektive processer er i højsædet. For at kunne indføre selvforbedring i en organisations projektgrupper er det væsentligt, at der sker en omdefinering af rollerne mellem projektgrupperne og SPI gruppen. Desuden skal projektgrupperne motiveres til at foretage double-loop læring og at fokusere på at få etableret en praksis. Til igangsættelse af selvforbedring i en organisations projektgrupper er der blevet præsenteret en alternativ forbedringsmodel. Denne model er grundlaget for vores samarbejde med Logimatic. Gennem en intervention forsøger vi at gennemføre nogle af de principper, som vi har stillet op i dette kapitel.

Inden vi gennemgår interventionen i Logimatic vil Logimatics nuværende SPI tilgang blive præsenteret i det følgende kapitel. Herefter vil selve interventionen med selvforbedrende projektgrupper blive præsenteret.

I dette kapitel vil vi foretage en analyse af Logimatic, hvor vi vil anvende begreber fra Mintzberg [Min83] til at få en forståelse af organisationen. Finn Hedehus, som er forretningsrådgiver for IT-Solutions, har venligst besvaret en række spørgsmål omkring Logimatic. På baggrund af det vil læseren, efter at have læst kapitlet, have en forståelse af Logimatics organisation.

5.1 Logimatic Gruppen

I det følgende afsnit vil vi foretage en vurdering af Logimatics organisation for at få en forståelse af, hvordan den fungerer og er opbygget. Et organisationsdiagram over hvordan Logimatic Gruppen er opbygget i dag kan ses i figur 5-1. Logimatic Gruppen er et holdingselskab, som ejes af otte aktionærer med hver 12,5%. De otte aktionærer er alle ansat i Logimatic Gruppen, og det er fastsat i aktionæroverenskomsten, at forudsætningen for at være medejer er, at vedkommende også er ansat i et af selskaberne i gruppen.



Figur 5-1: Organisationsdiagram over Logimatic Holding A/S.

I årene 1986-87 blev visionerne omkring etableringen af Logimatic modnet og realiseret, da der var udsigt til, at det daværende Aalborg Værft ville blive lukket. I årene derefter oprettede en håndfuld medarbejdere fra værftet i 1987 interessentselskabet Logimatic Marine Consultants, som i 1990 blev omdannet til aktieselskabet Logimatic A/S. Fokus var på rådgivning vedr. el- og automationdesign til skibe, og virksomheden fungerede i starten alene som et rådgivende ingeniørfirma, der primært henvendte sig til den maritime sektor heriblandt skibsværfter og rederier.

Fra 1988, få år efter at Logimatic A/S var etableret, begyndte en spirende softwarefunktion at vokse frem inden for forretningsområdet IT-Solutions. Det primære formål var at dække et internt

behov for specifikke IT-applikationer, som skulle støtte op omkring hovedaktiviteten el- og automationdesign til skibe. Sekundært var formålet at udvikle kundespecifikke løsninger, hvoraf nogen af disse stadigvæk er i brug. Som eksempel kan nævnes et time/sag system til Semco i Aalborg og et system til administrering af udsætning af fisk i vandløb til Ferskvandslaboratoriet i Silkeborg.

I 1989 fik Logimatic A/S opgaven at udvikle konfigurationsstyringssystemet ÆGIR til Søværnets Materiel Kommando. Formålet var at administrere den sammenhørende konfiguration for software, hardware og dokumentation for maskinkontrolsystemet på alle skibe af Stanflex klassen. Derudover indeholdt ÆGIR et vedligeholdsmodule, som var inspirationskilde til udvikling af et regulært vedligeholdelsesstyringssystem, og navnet blev ændret til LOGIHOLD. LOGIHOLD har haft stor betydning for Logimatic A/S både som et selvstændigt og som et rentabelt forretningsområde, men det har også banet vejen for flere større udviklingsprojekter, som organisatorisk løses i IT-Solutions.

I 1990 blev grundlaget for MARS skabt i samarbejde med Aarhus Flydedok A/S. I forbindelse med løsningen af en rådgivningsopgave blev der udtrykt ønske om at udvikle et materiale rekvireringssystem, som skulle styre værftets indkøbsfunktioner og lager. Udviklingen af systemet forløb planmæssigt, og det blev sat i drift i foråret 1991.

Efteråret 1991 blev det store gennembrud, da Logimatic tog et kvantespring med underskriften på en kontrakt med Kværner Masa-Yard i Finland, som er et af verdens største værfter. Referencen fra Kværner Masa-Yard var en strategisk vigtig parameter, og indenfor et par år fulgte flere nye kontrakter.

I 1992 blev softwarefunktionen i Logimatic formelt defineret som et selvstændigt forretningsområde oven på succesen med MARS og LOGIHOLD. Begge projekter startede i princippet som udvikling af kundespecifikke løsninger og har igennem Logimatics historie haft en afgørende betydning for udviklingen af softwarefunktionen. I de følgende år var der dog ikke meget tid til at udvikle kundespecifikke løsninger, og indsatsen blev koncentreret omkring MARS og LOGIHOLD.

Efteråret 1997 blev forretningsområdet Industrisystemer oprettet, og ideen var at skabe en forretning, som var baseret på udvikling, salg og markedsføring af et standardsystem, som kunne styre de mange forskelligartede processer, manuelle såvel som automatiske, der foregår på et lager. Ved opstarten af Industrisystemer var der allerede nogle servicekontrakter samt et enkelt projekt, men forretningsområdet fik først for alvor fart på i 1998 da det, fra EU's mål II midler, blev bevilget udviklingsstøtte til LOGIA, et standardsystem til styring af Intern Logistik og Lager.

I 1998 blev forberedelserne til at opdele Logimatic i forskellige forretningsområder indledt, og holdingselskabet Logimatic Holding A/S så dagens lys med Logimatic A/S som eneste driftsselskab, og koncernstrukturen fik navnet Logimatic Gruppen. Målsætningen med koncernstrukturen var at skabe grobund for en forsat forretningsudvikling og vækst ved at bane vejen for fremtidige selskabsopdelinger med nye forretningsområder. Derudover var det meningen, at der skulle være mulighed for at udvide aktionærkredsen i selskaberne med medarbejdere for dermed at lette et løbende generationsskifte og åbne op for en forsat organisationsudvikling med avancementmuligheder, indflydelse og udfordringer. Langsomt, men sikkert udviklede Logimatic Gruppen sig fra at være et traditionelt rådgivende ingeniørfirma til at være mere et egentlig softwarehus. Samme år

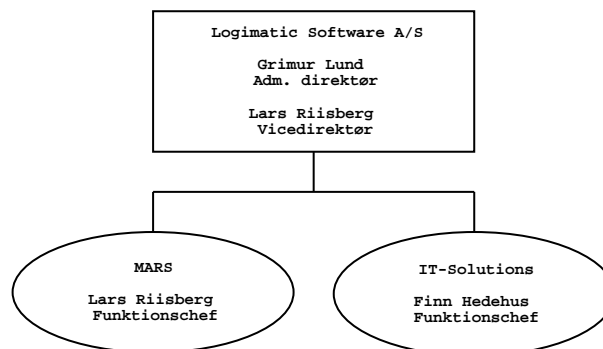
genopstår IT-Solutions som et selvstændigt forretningsområde, da der nu er tilstrækkeligt overskud i organisationen til at støtte op omkring IT udvikling.

I 1999 kom over 70% af gruppens samlede omsætning fra software relaterede aktiviteter. Derfor blev forretningsområdet Industrisystemer samme år udskilt i et særskilt driftsselskab, Logimatic Industrial Data System A/S, hvoraf Logimatic Gruppen ejer 50%, og hvor to medarbejdere hver ejer 25% af aktierne. Forretningsgrundlaget er at skabe en progressiv og lønsom softwarevirksomhed i markedet for industrielle datasystemer indenfor forretningsområderne LOGIA Logistik- og Lagerstyring, Lager Automation og Kommunikation System Integration.

I år 2000 kom den sidste opspaltning af Logimatic A/S i to nye driftsselskaber, som blev til Logimatic Engineering A/S og Logimatic Software A/S. I Logimatic Engineering A/S blev de oprindelige aktiviteter, som Logimatic startede med i 1987 videreført, men på grund af afmatningen indenfor den danske værftsindustri er opgaverne blevet af mere international karakter. De fleste marineopgaver er for udenlandske kunder, dog er Søværnets Materiel Kommando stadig en betydelig kunde. Der er også kommet andre opgaver til Logimatic Engineering A/S, som derfor også beskæftiger sig med elinstallationer og automation indenfor højteknologiske tog og nichebetonet industri. I Logimatic Software A/S blev aktiviteterne IT-Solutions, MARS og LOGIHOLD videreført. Det betyder at Logimatic Gruppen i dag består af tre forretningsområder, hvilket fremgår af figur 5-1.

5.2 Logimatic Software A/S

I forbindelse med dette projekt har vi arbejdet sammen med Logimatic Software A/S. Derfor vil vi i det følgende udelukkende beskæftige os med denne del af organisationen. Logimatic Software er 100% ejet af Logimatic Holding, og i selskabet sidder to fra aktionærkredsen omkring holdingselskabet som henholdsvis direktør og vicedirektør.



Figur 5-2: Organisationsdiagram over Logimatic Software A/S.

I 1992 blev softwarefunktionen for første gang, som nævnt tidligere, defineret som et selvstændigt forretningsområde med udviklingen af den første version af MARS som den væsentligste aktivitet, hvilket skete i samarbejde med Aarhus Flydedok A/S. I midten af 90erne blev softwarefunktionen den mest betydende aktivitet både i forhold til antal ansatte og omsætning, og det er den stadigvæk

i dag. Logimatic Software er inddelt i to forretningsområder, som hver især ledes af en forretningsområdeansvarlig også kaldt funktionschef, hvilket figur 5-2 illustrerer. De to forretningsområder er IT-Solutions og MARS.

Forretningsgrundlaget for Logimatic Software er at skabe en progressiv og lønsom softwarevirksomhed, som gennem faglig kompetence, kvalitet og professionalisme kombineret med moderne teknologi kan opnå stor anerkendelse. Fokus skal være rettet mod at kunne tilbyde software løsninger og ydelser inden for:

- Udvikling, afsætning og implementering af egne IT løsninger.
- Udvikling af kundespecifikke IT løsninger.
- At skabe et kuvøsemiljø for IT iværksættere, som i veletablerede rammer kan udvikle nye forretningsideer og/eller produkter i samarbejde med Logimatic Software A/S.

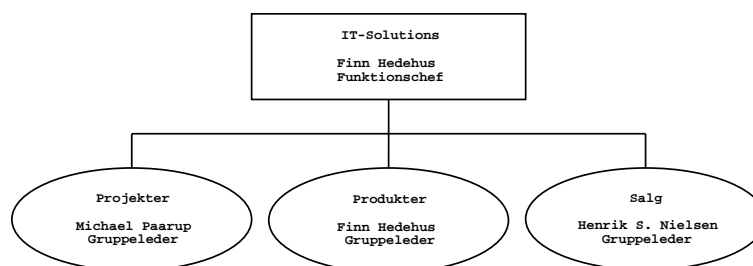
I Logimatic Software A/S er der ca. 42 ansatte, som fordeler sig som vist i tabel 5-1. Det er højt uddannede medarbejdere, der arbejder i organisationen, og som fortrinsvis består af ingeniører, dataloger og datamatikere. I MARS er der dog en overvægt af medarbejdere med en lang universitetsuddannelse i forhold til IT-Solutions, hvor det er medarbejdere med en mellemlang uddannelse, som dominerer billedet.

Medarbejdere	MARS	IT-Solutions
<i>Antal ansatte</i>	23	19
<i>Heraf ingeniører og dataloger</i>	65%	35%
<i>Heraf datamatikere</i>	30%	60%
<i>Heraf andre uddannelser</i>	5%	5%

Tabel 5-1: Ansatte i Logimatic Software A/S.

5.2.1 IT-Solutions

IT-Solutions udvikler kundespecifikke IT-løsninger og var ved opstarten af Logimatic virksomhedens oprindelige softwareaktivitet og dannede i sin tid grundlaget for udvikling af både MARS og LOGIHOLD.



Figur 5-3: Organisationsdiagram over IT-Solutions.

Forretningsområdet led en stille død, da fokus blev rettet mod udvikling af Logimatics kerneområde MARS og LOGIHOLD. I 1998 genopstod IT-Solutions som selvstændigt forretningsområde

og er i dag af strategisk betydning for Logimatic Software A/S. Inden for forretningsområdet er der mulighed for at skabe et kreativt softwaremiljø, afprøve nye teknologier samt udvikle helt nye produkter. Indtil for et år siden var LOGIHOLD et selvstændigt forretningsområde, men er i dag lagt ind under forretningsområdet IT-Solutions. Medarbejderne i LOGIHOLD varetager kundetilpasning og service/support af Logimatics vedligeholdssystem. Projekterne i dette område er typisk relativt små med deltagelse af en til to medarbejdere inklusiv projektlederen. Forretningsgrundlaget for LOGIHOLD er at bevare systemets position som værende blandt de mest solgte systemer i Danmark samt at etablere aftaler med internationale partnere, som vil afsætte og implementere systemet i udlandet. Derudover er formålet med LOGIHOLD produktet, at det skal fungere som døråbner til industrielle virksomheder, som kan være med til at initiere projekter til IT-Solutions.

IT-Solutions medarbejdere er beskæftiget med specifikation, udvikling og implementering af kundespecifikke løsninger. Arbejdet er organiseret i projekter med deltagelse af en til fem medarbejdere pr. projekt. Filosofien i IT-Solutions er, at IT standardsystemer ikke altid er en optimal løsning på en kundes problem. Derfor ønsker IT-Solutions at blive velkendt i Danmark som et professionelt softwarehus og skabe et forretningsområde med den nødvendige kompetence, størrelse og organisation til at løfte store og komplekse systemudviklingsopgaver med dertil hørende ydelser. Forretningsområdet IT-Solutions består af tre grupper, hvilket er illustreret i figur 5-3:

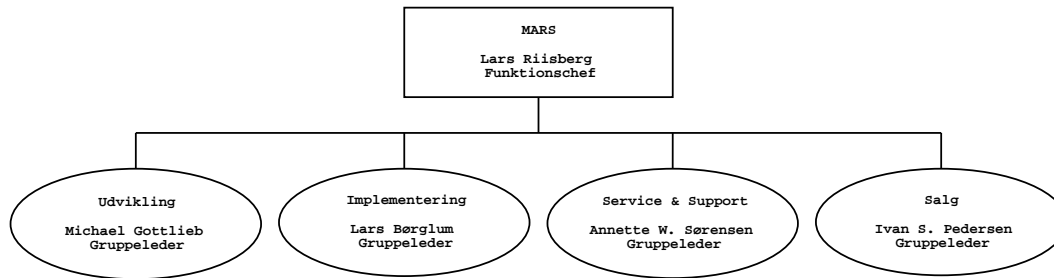
- **Produkter** er gruppen, der har ansvaret for standard produkterne i IT-Solutions. Det er her, LOGIHOLD er blevet lagt ind efter at det blev nedlagt som et selvstændigt forretningsområde.
- **Projekter** er gruppen, som har ansvaret for de forskellige udviklingsprojekter, der kører under forretningsområdet. Opdelingen mellem Projekter og Produkter er dog flydende, hvor personer deltager begge steder.
- **Salg** er gruppen, som er ansvarlig for salg og markedsføring af IT-Solutions produkter, og de er placeret fysisk sammen. Det samme gælder nogle af produktgrupperne, og generelt flyttes der relativt lidt rundt på de forskellige grupperes placering.

5.2.2 MARS

MARS er gennem årene blevet til Logimatics største forretningsområde og har i dag en helt afgørende betydning for firmaets udvikling. Systemet er et af de førende i verden blandt materiale- og produktionsstyringssystemer til skibsværfter. Forretningsgrundlaget for MARS er at udvikle, sælge og implementere administrative softwareapplikationer til materiale- og produktionsstyring samt nært beslægtede IT løsninger for projektorienterede virksomheder med specielt fokus på skibsværfter. Derudover er der et mål om at gøre MARS til det globalt mest anerkendte produkt til styring af materiale- og produktionsprocesser på industrielle ETO virksomheder.

Forretningsområdet MARS forestår udvikling, salg, kundetilpasninger og service & support af Logimatics materiale og produktionsstyringssystem MARS og er inddelt i fire selvstændige enheder, hvilket figur 5-4 illustrerer. Medlemmerne af de enkelte grupper er fysisk placeret sammen.

- **Implementeringsgruppen** forestår implementering af MARS systemet hos kunderne. Medarbejderne er beskæftiget med projektledelse, kravspecifikation, undersøgelse af gennemførlighed, programmering, datakonvertering, etablering af interface til andre systemer samt undervisning af brugere.



Figur 5-4: Organisationsdiagram over forretningsområdet MARS.

- **Udviklingsgruppen** beskæftiger sig primært med udvikling af MARS standardversionen, der danner grundlaget for implementeringen af systemet hos slutbrugerkunden. Endvidere bidrager udviklingsgruppen med programmeringsassistance i forbindelse med implementeringsprojekterne.
- **Service- & Supportgruppen** varetager primært hotline og servicering af systemet efter, at det er taget i drift hos kunden. Endvidere forestår gruppen mindre funktionsændringer i kundens version, og de bidrager også med programmeringsassistance i implementeringsprojekterne.
- **Salg** er ansvarlig for salget af MARS, og gruppen består af en enkelte person, der trækker på funktionschefen for MARS, direktøren samt projektlederne i implementeringsgruppen i forbindelse med salgsindsatsen.

5.2.3 Vurdering i forhold til designparametrene

I følgende afsnit vil vi foretage en analyse af Logimatic Software A/S med vægt på MARS forretningsområdet i forhold til Mintzbergs designparametre. Mintzberg opstiller 9 designparametre, som sammen med situationsfaktorerne medfører en organisationskonfiguration. De fire primære designparametre er *design af stillinger*, *design af organisationsstruktur*, *design af strukturkoblinger* og *design af beslutningssystemer*. Ud over at designparametrene skal passe til situationsfaktorerne, er det ifølge konfigurationshypotesen også vigtigt, at der er intern konsistens mellem designparametrene.

Design af stillinger

Design af stillinger kan opdeles i tre faktorer jobspecialisering, adfærdsformalisering samt træning og indoktrinering, der hver især har betydning for, hvordan stillinger designes.

Jobspecialisering handler om den basale arbejdsdeling i organisationen og kan opdeles i en horisontal og vertikal retning. Den horisontale retning drejer sig om bredden i jobbet, dvs. hvor mange opgaver den enkelte har, og den vertikale retning handler om i hvilken udstrækning, den enkelte har indflydelse på jobfunktionen og egenkontrol over opgavernes udførsel. Jobspecialiseringen i Logimatic er karakteriseret ved at være *horisontal specialiseret og vertikal udvidet*, da det i høj grad er højtuddannede medarbejdere, som er ansat i stillingerne. Den horisontale specialisering betyder, at den enkelte i Logimatic har nogle få specialiserede opgaver at løse. Når jobspecialiseringen er vertikal udvidet, har det samtidigt den betydning, at den enkelte i Logimatic har en

høj grad af indflydelse på sin egen jobfunktion og kontrol over, hvordan opgaverne skal udføres. En høj grad af egenkontrol over opgaverne har også den virkning, at der ikke er behov for en masse mellemledere til at koordinere arbejdet i Logimatic, lige som der ikke er brug for en masse analytikere til at udforme standarder for, hvordan opgaverne skal løses.

Adfærdsformalisering handler om i hvilken grad, arbejdsindholdet i hver stilling skal standardiseres. Det kan ske igennem en regulering af stillingen, arbejdsflowet eller ved udformning af generelle regler. Det er vores indtryk, at der i Logimatic er en *lav grad af formalisering* af de enkelte stillinger. Den måde Logimatic regulerer de enkelte stillinger på er gennem deres kvalitetssystem. I kvalitetssystemet er der procedurer og standarder for, hvordan medarbejderne skal løse nogle af deres opgaver. Fx er der en standard for navngivning af filer eller en standard for SPI gruppens ansvar og rolle. Men ellers er det i vid udstrækning op til den enkelte i Logimatic at beslutte, hvordan arbejdet skal udføres, blot det holder sig inden for rammerne af kvalitetssystemet. En lav grad af formalisering har den betydning, at Logimatic mister noget af kontrollen over medarbejderne i virksomheden. Til gengæld betyder den lave grad af formalisering, at Logimatic har lettere ved at omstille sig, når den omgivne verden forandrer sig. Fordi der eksisterer en vis grad af formalisering i Logimatic i kraft af kvalitetssystemet, og på grund af den standardisering der er sket af medarbejdernes færdigheder igennem deres uddannelse, kan vi karakterisere Logimatic for at være en *bureaukratisk* organisation.

Oplæring og socialisering drejer sig om, hvilke færdigheder og kundskaber en stilling kræver. Oplæring er hvilke kvalifikationer, den enkelte skal have erhvervet sig igennem uddannelse for at besidde stillingen. Socialisering er derimod at lære normer, værdier og holdninger, hvilket er op til organisationen, hvordan den vil præge den enkelte igennem en kulturel og ideologisk påvirkning. I Logimatic har alle medarbejderne en eller anden form for specialistuddannelse bag sig enten som ingeniør, datalog eller datamatiker, der er derfor krav om *meget uddannelse* for at kunne bestride stillingerne i Logimatic. Efteruddannelse sker på baggrund af den enkeltes egne ønsker kombineret med de behov, som jobbet og arbejdsopgaverne stiller. Til de årlige medarbejdersamtaler er efteruddannelse et fast punkt, hvor ønsker og muligheder diskuteres mellem medarbejderen og dennes leder. Der afholdes, specielt inden for MARS området, en del kurser i MARS systemet og tilhørende værktøjer. Herudover anvendes eksterne kurser hos forskellige kursusleverandører. Netop på grund af den meget specialiserede uddannelse, de professionelle i Logimatic har erhvervet, betyder det, at det ikke blot er vanskeligt at standardisere deres arbejdsprocesser, det er også svært at standardisere outputtet fra deres arbejde. Det har den effekt, at meget af magten i Logimatic ligger hos de professionelle på grund af deres viden og færdigheder.

Socialiseringen foregår i Logimatic ved, at det er funktionschefens ansvar at planlægge modtagelsen af den nye medarbejder og herunder fastlægge arbejdsopgaver og projekttilknytning. Gennem introduktionsforløbet introduceres medarbejderne til kvalitetssystemet, det interne administrationssystem, regler for tidsregistrering mv. Endvidere introduceres vedkommende til projektet eller arbejdsopgaven, hvilket typisk foretages af vedkommendes projektleder. Nye medarbejdere kobles typisk til en erfaren medarbejder, det kan være en projektleder, en gruppeleder eller en udvikler i det område, hvor vedkommende skal arbejde. Kulturen og værdierne i Logimatic overføres af kollegaerne gennem deltagelse i projektarbejdet. Det er vores indtryk, at der sker *lidt socialisering* i Logimatic.

Design af organisationsstrukturen

Design af organisationsstrukturen er kendetegnet ved enhedsgruppering, hvor der tages stilling til på hvilket grundlag, stillingerne skal grupperes og enhedsstørrelsen, som drejer sig om størrelsen på gruppenhederne.

Enhedsgruppering er med til at afgøre, hvordan organisationen inddeler og koordinerer sit arbejde. Enhederne kan enten markedsgrupperes efter kunde, produkt eller geografi, eller funktionsgrupperes på baggrund af faglighed eller arbejdsprocesser. Enhedsgrupperingen i Logimatic er kendetegnet ved, at de både er *funktionelt og markeds-mæssigt grupperet*. De enkelte grupper i MARS og IT-Solutions sidder fordelt efter, hvilken arbejdsproces de har, dvs. funktionsgruppering. I figur 5-3 kan vi se, at IT-Solutions sidder grupperet i Produkter, Projekter og Salg. Forretningsområdet MARS er på samme måde kendetegnet ved at være grupperet på baggrund af arbejdsprocesser i grupperne Implementering, Udvikling, Service & Support og Salg, hvilket figur 5-4 illustrerer, dvs. funktionsgrupperet. Samtidig er grupperne i Logimatic også grupperet på baggrund af produkt i forretningsområderne MARS og IT-Solutions, det betyder, at grupperne i Logimatic også er markedsgrupperet.

Enhedsstørrelse drejer sig om, hvor stor hver enkelt enhed skal være, og hvor mange mennesker, som skal rapportere til en bestemt overordnet. I Logimatic består grupperne af 1 til 5 medlemmer, og enhedsstørrelsen er derfor karakteriseret ved *små grupper*. Små enheder vil medføre en smal kontrolspændvidde, hvilket er med til at gøre, at der ikke er et stort behov for standardisering. Tværtimod vil de små enheder betyde, at der er behov for en øget grad af gensidig tilpasning for at koordinere arbejdet imellem enhederne, hvilket vi også har observeret, sker i Logimatic.

Design af strukturkoblingerne

Design af strukturkoblingerne bygger på planlægnings- og kontrolsystemerne i organisationen samt de forbindelses- og kommunikationsordninger, der eksisterer for at fremme kommunikationen mellem de enkelte grupper af i organisationen.

Planlægnings- og kontrolsystemer handler om i hvilket omfang, output fra hver enkelt stilling eller gruppenhed skal standardiseres. De to forskellige måder at udføre planlægning og kontrol på er præstationskontrol og handlingsplanlægning. I Logimatic er planlægning og kontrol ikke en mekanisme, som der benyttes så ofte. Hvert tredje eller femte år udarbejdes der en perspektivplan for hvert forretningsområde, som sætter målene for den næste tre til femårige periode. Planerne revideres ca. hvert andet år af ledelsen i selskabet i samarbejde med medarbejderne og bestyrelsen. Perspektivplanerne beskriver overordnede mål inklusive økonomiske mål, som der efterfølgende følges op og måles på. Det, som Logimatic kalder for perspektivplaner, er det, som Mintzberg karakteriserer som *præstationskontrol*. Det centrale i præstationskontrol er netop at regulere det overordnede resultat for en given enhed, hvilket bl.a. foregår igennem strategiplaner og budgetter. Formålet er dels at måle, men også at motivere. Problemet ved præstationskontrol er, at den skal tilpasses hver enkelt enheds særkende, hvorfor denne form for kontrol vil foregå, hvor enhederne er markedsgrupperet, hvilket MARS og IT-Solutions netop er.

Målene i perspektivplanen realiseres i Logimatic via *handlingsplaner*, der beskriver de enkelte aktiviteter, der er knyttet til at nå målene. Ansvar for en handlingsplan uddeles typisk til en

med-ar-bej-der eller leder, og der følges løbende op på status for handlingsplanerne. På det operative niveau findes yderligere ressourceplaner for hvert forretningsområde, som det er den forretningsansvarliges opgave at udarbejde og afstemme eller kommunikere med medarbejderne. Projektplaner udarbejdes af projektlederen evt. i samarbejde med funktionschefen, og projektplanen godkendes af kunden. I modsætning til præstationskontrol benyttes handlingsplaner, når planlægning foregår på tværs af enhederne og benyttes, hvor grupperne er funktionsgrupperet, hvilket grupperne inden for hvert forretningsområde i Logimatic netop er.

Handlingsplanlægning bruges ofte overfor ikke rutineprægede opgaver og foregår som regel ved en top-down beslutning i organisationen. I relation til hinanden medfører præstationskontrol præstationsstandarder over en tidsperiode, men uden at der er reference til bestemte handlinger, fx forøg salget med 10 procent i år på vores softwareprodukt. Handlingsplanlægning derimod betyder, at beslutninger og handlinger vil blive udført på et bestemt tidspunkt i fremtiden, fx forøg salget med 10 procent ved at lave en forbedret udgave af vores softwareprodukt i indeværende år.

Forbindelsesanordninger handler om hvilke mekanismer, der skal etableres for at den gensidige tilpasning mellem stillinger og enheder fremmes. Derfor vælger organisationen at etablere samarbejdsfremmende foranstaltninger, såsom forbindelsesstillinger, projektgrupper, stående udvalg, integrationsledere og matrixstrukturer.

Forbindelsesstillinger oprettes ofte, hvis der er brug for kommunikation mellem to gruppeenheder til at organisere arbejdet. Formålet er at skabe en direkte kommunikation for derved at omgå de vertikale kanaler, men alligevel har ingen formel magt tilknyttet. I Logimatic er der oprettet forbindelsesstillinger for hver enkelte gruppe, hvor der er udnævnt en gruppeleder. Gruppelederen har ikke nogen formel magt i Logimatic, men det er gruppelederens opgave at være med til at koordinere arbejdet på tværs af grupperne, hvis der er behov for det. Fx er der på MARS området brug for i nogle sammenhænge at koordinere arbejdet mellem grupperne Implementering og Service- & Supportgruppen, når et projekt overdrages fra Implementering til Service & Support.

Projektgrupper og stående udvalg er en anden måde at skabe kontakt på, på tværs af de etablerede grupper. Projektgrupper dannes med et formål for øje som regel med henblik på at løse en stille opgave. Projektgruppen kan bestå af forskellige medarbejdere fra de enkelte enheder, og når opgaven er løst, opløses gruppen igen. I Logimatic dannes forskellige projektgrupper, når en opgave skal løses. På MARS området sker det fx ofte, at udviklere fra de forskellige grupper er med på implementeringsprojekterne, hvor de yder programmeringsassistance. I den forbindelse nedsættes der en projektgruppe afhængig af opgaven.

Stående udvalg derimod har en mere permanent karakter med en regelmæssig mødefrekvens, hvor udvalget mødes for at diskutere problemstillinger eller emner af fælles interesse. Af vigtige stående udvalg i Logimatic findes MARS Produktråd, LOGIHOLD Produktråd, Samarbejdsudvalget, Sikkerhedsudvalget og SPI gruppen. I MARS Produktrådet sidder følgende medarbejdere:

- Lars Riisberg
- Dorte Pedersen
- Annette Wulf Sørensen
- Lars R. Børglum
- Michael Gottlieb

Produktrådets opgave er at udarbejde krav til funktionaliteten i MARS versionerne, ligesom de står for godkendelse af ændringer i produktet. Rådet er et forholdsvis centralt organ i Logimatic, set i lyset af, hvilken betydning MARS produktet har for virksomheden. Rådet består udelukkende af professionelle folk med ekspertise inden for software udvikling. SPI gruppen er også en vigtig faktor i Logimatic, da det er denne gruppe, som står for procesforbedringsaktiviteterne i organisationen.

Udover disse udvalg afholdes der også møder i følgende regi:

- Logimatic Gruppen (alle ansatte)
- Logimatic Software (alle ansatte)
- Ledelsesmøder (ledelsen i Logimatic Software)
- Funktionsmøder (alle ansatte i funktionen)
- Projekt møder (projektleder og deltagere)

I forhold til Ledelsesmøderne er det direktøren i Logimatic Software og funktionslederne, som deltager i disse møder. Når der opstår behov for en højere grad af gensidig tilpasning end de tre foregående forbindelsesordninger, er der mulighed for at etablere en *integrationsleder*, som i modsætning til en forbindelsesstilling har formelt magt i organisationen, og som går på tværs af flere grupper. I denne forbindelse har Logimatic valgt at oprette to integrationsledere for hvert af forretningsområderne MARS og IT-Solutions. Begge ledere har formel magt i organisationen, og det er også dem, der sammen med direktøren i Logimatic Software, afholder ledelsesmøder.

Design af beslutningssystemer

Design af beslutningssystemer opdeles i vertikal og horisontal decentralisering. Det centrale er den administrative arbejdsfordeling, hvordan beslutninger træffes i organisationen, og dermed hvor magten er placeret.

Vertikal decentralisering drejer sig om hvor stor beslutningsmyndighed eller beføjelser, som bør delegeres til mellemledere på forskellige niveauer i hierarkiet. Det handler om, hvordan den formelle magt delegeres ned igennem organisationen fra topledelsen til mellemledelsen. Tre designspørgsmål gør sig gældende, hvilke beslutningskompetencer skal delegeres, hvor langt skal de delegeres, og hvordan skal det kontrolleres, at de overholdes.

Horisontal decentralisering går ud på, hvor stor en beslutningsmyndighed, som bør overgå fra mellemlederne til stabsspecialisterne og operatørerne. Der er fokus på, hvordan den uformelle magt forskydes til medarbejderne i organisationen, dvs. væk fra topledelsen og mellemledelsen til analytikerne, servicepersonalet og operatørerne. Beslutninger i Logimatic træffes så tæt på de personer, som beslutningen vedrører, når behovet for overordnet prioritering og koordinering tages i betragtning. Projektlederen har stor beslutningskompetence med hensyn til projektet indenfor de aftalte rammer, som rummer ressourcer, mål og tid. Hvis der skal ske ændringer i prioriteringen af ressourcer, sker det i samarbejde med funktionschefen/gruppelederen. Selve organisationen er relativ flad med en kort vej til ledelsen, og den åbne dør politik praktiseres overalt i firmaet. Beslutningssystemet er derfor både *vertikal decentraliseret* og *horisontal decentraliseret*, og den uformelle magt i Logimatic er forskudt til eksperterne igennem deres viden og færdigheder.

Opsummering

Vi har i det foregående afsnit foretaget en grundig analyse af Logimatic, ud fra Mintzbergs designparametre. I tabel 5-2 har vi opsummeret de vigtigste træk, som vi har analyseret os frem til.

Designparametre		Variabler
<i>Stillinger</i>	<i>Jobspecialisering</i>	Specialiseret horisontal job Udvidet vertikal job
	<i>Adfærdsformalisering</i>	Lidt bureaukratisk
	<i>Oplæring og socialisering</i>	Meget uddannelse, lidt socialisering
<i>Organisationsstruktur</i>	<i>Gruppering</i>	Markeds- og funktionsopdelt
	<i>Gruppestørrelse</i>	Små grupper
<i>Strukturkoblinger</i>	<i>Planlægning og kontrol</i>	Få handlingsplaner og lidt præstationskontrol
	<i>Samarbejdsfremmende foranstaltninger</i>	Forbindelsesstillinger, projektgrupper, stående udvalg og integrationsleder
<i>Beslutningssystemer</i>	<i>Vertikal decentralisering</i>	Vertikal decentraliseret
	<i>Horisontal decentralisering</i>	Horisontal decentraliseret

Tabel 5-2: Oversigt over designparametre i Logimatic.

5.2.4 Vurdering i forhold til situationsfaktorerne

I følgende afsnit vil vi foretage en analyse af Logimatic i forhold til Mintzbergs situationsfaktorer. Fire situationsfaktorer er sammen med designparametrene, ifølge kongruenshypotesen, afgørende for, hvilken organisationsform der bør vælges. De fire situationsparametre er *alder og størrelse* på organisationen, det *tekniske system*, der benyttes i produktionskernen, *omgivelserne*, som består af *stabilitet, kompleksitet, heterogenitet* og *fjendtlighed* og den sidste situationsfaktor er *magtfaktorer* som består af *ejerforhold, magtbehov* og *kulturelle forhold*. Grundlæggende betyder det, at den succesfulde organisation vælger at designe sin struktur, så den passer bedst muligt til den situation, som organisationen befinder sig i.

Alder og størrelse

Alder og størrelse er to parametre, som hænger sammen. Alder har den betydning, at desto ældre organisationen er, jo mere formaliseret vil den være og afhængig af sin historiske baggrund. Størrelse har den indflydelse, at desto større organisationen er, desto mere specialiseret er den, enhederne vil være større, og der vil være en mere formaliseret adfærd. Logimatic blev stiftet i 1986-87 og er ikke en ung organisation, og der er styr på, hvordan arbejdet bliver koordineret. Dog er Logimatic ikke blevet så formaliseret og præget af sin historiske baggrund, at vi vil karakterisere virksomheden som en gammel organisation. I stedet for vælger vi at karakterisere Logimatic som en *moden* virksomhed, der er kommet over den usikre opstartsperiode, og som har fået styr på sine arbejdsprocesser. I Logimatic arbejder der 42 eksperter, og vi mener derfor, at der

er tale om en relativ lille virksomhed, i forhold til organisationer som fx KMD, hvor antallet af medarbejdere tælles i tusinder.

Det tekniske system

Det tekniske system findes ifølge Woodward i tre forskellige former. Enhedsproduktion der er karakteriseret ved en organisk organisation med lidt grad af formalisering og præget af gensidig tilpasning eller direkte overvågning, og hvor der produceres en enkelt enhed, som er kundetilpasset. Masseproduktion hvor der produceres mange standardenheder, og som er karakteriseret ved bureaukrati, øget formalisering og kraftig indflydelse fra teknostrukturen. Procesproduktion hvor der er en meget høj grad af automatisering hvilket betyder, at det tekniske system kan betjenes af et fåtal af medarbejdere, og hvor der er en meget høj grad af standardisering. Det tekniske system i Logimatic er kendetegnet ved *enhedsproduktion*, hvor der bygges produkter, der ikke bliver massefremstillet. Produktionsapparatet der benyttes er *simpelt og ikke reguleret*, hvilket ikke skal forstås som, at vi mener, der er tale om simple produktionsmetoder, men at de remedier som bruges i form af computere og hjælpeværktøjer er et simpelt teknisk system. Teknologien i virksomheden i form af viden og erfaring er derimod sofistikeret. Når det tekniske system er simpelt og ikke reguleret, betyder det mindre bureaukrati i produktionskernen og mindre behov for en støttestab til at vedligeholde det tekniske system.

Omgivelserne

Omgivelserne består af fire faktorer, som er styret af stabilitet, kompleksitet, heterogenitet og fjendtlighed.

Stabilitet handler om, i hvor høj grad organisationens omverden er præget af usikkerhed og hvor ofte, der kan ske pludselige skift i kundepræferencer, politiske indgreb eller andre faktorer, som for organisationen er uventede og uberegnelige begivenheder. Dimensionen går fra stabil til dynamisk, hvor vi karakteriserer en omgivelse som dynamisk, hvis den er præget af høj usikkerhed, hvilket vil bevirke en mere organisk organisation med lidt formalisering.

Kompleksitet er et begreb, som definerer hvilke opgaver, virksomheden arbejder med og de krav, som stilles til organisations ekspertise og teknologi. Dimensionen går fra simpel til kompleks, hvor en øget kompleksitet i opgaveløsningen vil medføre en decentral struktur, hvorimod simple opgaver vil medføre centralisering.

Heterogenitet handler om, hvor forskelligt det marked er, som virksomheden opererer i. Dimensionen går fra homogent til heterogent, og er afhængige af tre variabler som er kunder, produkter og geografiske område. Et heterogent marked vil medføre, at organisationen grupperer sig i små markedsbaserede grupper, og omvendt vil et forholdsvis homogent marked betyde større grupper, der er organiseret efter funktion.

Fjendtlighed drejer sig om i hvilken grad, der er konkurrence og forskellige former for konflikter i den omgivne verden. I en verden som er fjendtlig stemt over for organisationen, vil der være et behov for, at der kan reageres hurtigt på udefra komne trusler. Det vil, alt andet lige, medføre, at organisationen vil have behov for at indrette sig, så den hurtigt kan reagere på trusler og udfordringer hvilket medfører en centraliseret struktur.

I forhold til omverdensfaktorerne må vi sige at Logimatic opererer i en forholdsvis *stabil* verden, hvor der ikke sker så meget forandring. Det skal forstås på den måde, at nok sker der løbende forandringer, men det sker i et stabilt og roligt tempo. Logimatic har været på markedet længe, og samtidig er det et marked, som de er fortrolige med. De medarbejdere, som var med til at starte Logimatic op i 1987, havde en baggrund fra Aalborg Værft, hvorfra de har en solid erfaring i, hvordan et værft er organiseret, og hvordan de interne arbejdsprocesser er. Selvom værftindustrien udvikler sig, og der kommer nye arbejdsprocesser til, så sker der ikke de store grundlæggende forandringer såsom at holde styr på materialerne og bestilling af nye råvarer og reservering til de enkelte produktionsenheder. Derfor er Logimatic bekendt med, hvad kunderne efterspørger. Men det er klart, at selvom værftindustrien er kendt, så ændrer teknologien også mulighederne for, hvordan softwareprodukterne ser ud, og igennem de senere år er der kommet fokus på, at MARS skal integreres med andre applikationer, hvorfor der i perspektivplanen for MARS er fokus på udvikling af certificerede interfaces til CAD systemet Tribon, økonomisystemerne Oracle Financials og SAP.

Selv om MARS produktet er et standardssystem, som bliver tilpasset den enkelte kunde, er der tale om et forholdsvis *kompleks* produkt, hvor der især skal holdes øje med de enkelte kundetilretninger, der bliver foretaget i produktet. Det er en af grundene til, at netop konfigurationsstyring er overordenligt vigtigt for MARS, og derfor har Logimatic udviklet deres eget konfigurationssystem til at holde styr på, hvilke versioner der er installeret hos den enkelte kunde.

Det marked Logimatic opererer i er både *homogent* og *venligt* stemt, da MARS er et nicheprodukt, der primært henvender sig til skibsværfter. Systemet har dog potentiale til at blive anvendt inden for anden ordreproducerende industri, men pt. er fokus stort set udelukkende rettet mod værfter. I dette segment er der andre konkurrenter, heriblandt store ERP leverandører som SAP, IFS, Baan etc. Disse systemer henvender sig også til andre brancher, men det er Logimatics indtryk at de finder skibsværftsbranchen mere interessant end tidligere. Øvrige konkurrenter til MARS kan være små lokale leverandører, der evt. har udviklet en skræddersyet løsning til et enkelt værft. Hvis virksomheder som SAP begynder at fokusere mere på værftindustrien, kan det øge konkurrencen for Logimatic. Det er måske også derfor, at virksomheden er begyndt at fokusere på CMM og ISO.

Magtfaktorer

Magtfaktorerne har også en indflydelse på, hvordan en organisation vil vælge at indrette sig.

Ekstern kontrol er den første parameter, som skal vurderes. Hvis der er en høj grad af ekstern kontrol, vil det medføre, at organisationen vælger en høj grad af formalisering igennem enten ledelse eller regler.

Magtbehov er en anden magtfaktor, som betyder, at der kan være forskellige krav fra personalegrupper i organisationen. Afhængig af hvilke grupper som har magtbehovet, vil det skabe forskellige typer af decentralisering.

kulturelle forhold eller sociale normers udvikling er den sidste magtfaktor, hvor aktuelle eller kulturelle sædvaner kan være med til at fremme visse strukturer. Dvs. hvad der nu er oppe i tiden gør, at en organisation vælger at indrette sig på en bestemt måde.

Der er *ingen ekstern kontrol* i forhold til Logimatic Software, da alle ejerne arbejder i organisationen. Men da både direktøren og vicedirektøren er medejere af Logimatic Gruppen betyder det,

at der kan være en tendens til et magttræk mod centralisering i organisationen. Dog er det ikke vores indtryk, at det er et magttræk, der dominerer i virksomheden. Dette skyldes, at topledelsen udgøres af professionelle folk, her forstået som folk med samme kompetencer som produktionskernen. Dvs. der er tale om en professionel ledet administration, der lytter og træffer beslutninger ud fra teknisk rationelle grunde, og derfor er magttrækket snarere henimod *professionalisme* i organisationen. I forhold til de kulturelle forhold er det en *moderne* virksomhed, som har et moderne syn på tilværelsen, og der anvendes en uformel ledelsesstil, hvor åben dør politikken praktiseres.

I tabel 5-3 har vi opsummeret de vigtigste situationsfaktorer hos Logimatic, som vi har analyseret os frem til i forhold til Mintzberg.

Situationsfaktorer		Variabler
<i>Alder/størrelse</i>		Moden, lille
<i>Teknisk system</i>		Enhedsproduktion, ikke reguleret, simpelt
<i>Omgivelser</i>	<i>Stabilitet</i>	Stabil
	<i>Kompleksitet</i>	Kompleks
	<i>Heterogenitet</i>	Heterogen
	<i>Fjendtlighed</i>	Venlig
<i>Magtfaktorer</i>	<i>Ejerforhold</i>	Ingen ekstern kontrol
	<i>Magtbehov</i>	Professionalisme
	<i>Kulturelle forhold</i>	Moderne

Tabel 5-3: Oversigt over situationsfaktorer for Logimatic.

5.2.5 Samlet vurdering

Vi har foretaget en analyse af virksomheden Logimatic Software A/S og er kommet frem til at der er flere faktorer som peger på, at Logimatic i Mintzberg termer, er et professionelt bureaukrati. Det professionelle bureaukrati er præget af horisontal specialisering med en decentral struktur. Konfigurationen er præget af en bureaukratisk struktur, men med en lille grad af formalisering. Grupperingen er både funktionel og markedsræssig, og de fleste medarbejdere er placeret i produktionskernen. Gruppestørrelserne er relative små med en smal kontrolspændevidde. Derudover er der relativt få planlægnings- og kontrolstrukturer, og koordineringen foregår gennem standardisering af fagkyndighed. Det professionelle bureaukrati opererer i en stabil omverden, men med en relativt kompleks opgaveløsning. I forhold til de opsummeringer vi har lavet i tabel 5-3 over situationsfaktorerne og i tabel 5-2 over designparametrene, mener vi, at Logimatic er et professionelt bureaukrati.

Logimatic har tidligere gennemført forskellig kvalitetsinitiativer, hvor det tiltag, som har haft størst betydning har været processen omkring indførelse af et kvalitetssystem. De indledende skridt blev taget i 1996-97, hvilket resulterede i, at virksomheden blev certificeret i henhold til ISO 9001:94 standarden i september 1998.

Det forløb, som ledte frem til selve certificeringen, har medført en række forbedringer i form af, at forretningsgangene i Logimatic er blevet beskrevet og dokumenteret. Der er blevet opstillet formelle krav til, hvordan udvikling, implementering og servicering af software skal foregå og stillet krav til, hvordan udviklingsprojekterne skal ledes. Derudover er der, som en del af certificeringsprocessen, blevet udviklet et internt system, RLMC, som skal understøtte projektlederne og udviklerne i discipliner som kravstyring, styring af arbejdsopgaver og registrering af ændringer i kode. Da kvalitetssystemet blev oprettet, fokuserede Logimatic på i første omgang, at beskrive de processer, der blev brugt i virksomheden på daværende tidspunkt, i stedet for at designe nye processer.

Processerne blev beskrevet af nogle enkelte medarbejdere, der hver fik deres område at arbejde med. Vores tidligere undersøgelser i Logimatic har dog vist, at kvalitetssystemet på nogle områder fokuserer meget på MARS området, og at der ikke senere er taget hensyn til, at processerne i IT-Solutions er anderledes. Det betyder, at på trods af, at Logimatic er en forholdsvis lille virksomhed, viser det sig, at det er svært at få processerne i kvalitetssystemet til at stemme overens med den måde, der rent faktisk arbejdes på [RJN01].

ISO certificeringen har betydet, at der er kommet øget fokus på kvalitet og refleksion over processerne, og det har betydet et betydeligt løft i kvalitetsstyringen af softwareprocesserne hos Logimatic. Det der desværre er sket for Logimatic siden certificeringen er, at fokus på kvalitetssikring og softwareprocesforbedring har været faldende blandt både ledelsen og medarbejderne [APH01].

For igen at skabe fokus på kvalitetssikring og få fornyet indspark til igen at skabe grobund for en øget fokusering på procesforbedringer i Logimatic, deltog tre medarbejdere i uddannelsen IT-projektledelse i perioden september 2000 til marts 2001. Målet for deltagerne var at gennemføre et konkret forbedringsinitiativ i Logimatic og initiere en fortsat forbedringsproces i virksomheden. Gruppen valgte i deres fremgangsmåde af forbedringsindsatsen at benytte sig af IDEAL modellen [APH01].

Første fase i gruppens arbejde var derfor at foretage en diagnosticering i form af en intern CMM baseret modenhedsmåling, som skulle afsløre svagheder og styrker i udførelsen af projektledelse i virksomheden. På baggrund af diagnosens resultater blev der udvalgt et antal interventionsområder, og der blev iværksat et konkret forslag udmøntet i kvalitetsmøder. Kvalitetsmøderne har været afprøvet i nogle projektgrupper som pilotprojekter, men derefter er SPI indsatsen i Logimatic gået mere eller mindre i stå. Især nedturen inden for softwareindustrien har betydet, at ledelsen

i Logimatic ikke har kunnet afse så meget tid til den fortsatte SPI indsats¹.

Derudover har Logimatic i efteråret 2001 gennemført endnu en CMM måling, der viste, at organisationen ikke havde rykket meget længere op på CMM skalaen, end det den forrige måling viste. Målingen i 2001 er Logimatics grundlinje, som de fremover vil sammenligne deres udvikling, i forhold til CMM skalaen, imod.

6.1 SPI gruppen

Rammerne for SPI gruppen blev etableret under projektlederuddannelsen af medarbejdere fra Logimatic, herunder blev også den egentlige SPI gruppe nedsat som en permanent gruppe i virksomheden. Gruppens overordnede mål er at bringe Logimatic Software fra nuværende CMM niveau 1,5 til et tillempet niveau 3 i løbet af en 3 årig periode startende fra år 2001. På baggrund af årlige CMM målinger vil der blive opstillet delmål, hvilket vil blive søgt opfyldt ved hjælp af udarbejdelse af handlingsplaner.

Gruppen består af 3-4 personer, hvoraf der mindst skal være en ledelsesrepræsentant samt en repræsentant med relationer til kvalitetssystemet. Udpegningen til gruppen foretages af Logimatics ledelse, og SPI gruppen konstituerer sig selv med en formand, som har ansvaret for mødeindkaldelse og udarbejdelse af mødereferater. Mindst en gang om måneden afholder gruppen møder, og det tilstræbes, at varigheden ikke overstiger to timer. Efter hvert møde udarbejder formanden et referat der summarisk oplister alle vedtagne beslutninger på mødet. Desuden indeholder hvert referat en punchliste, som lister alle udestående opgaver, en status på dem samt en ansvarlig og termin for, hvornår de skal være udført. Medlemmer af SPI gruppen er:

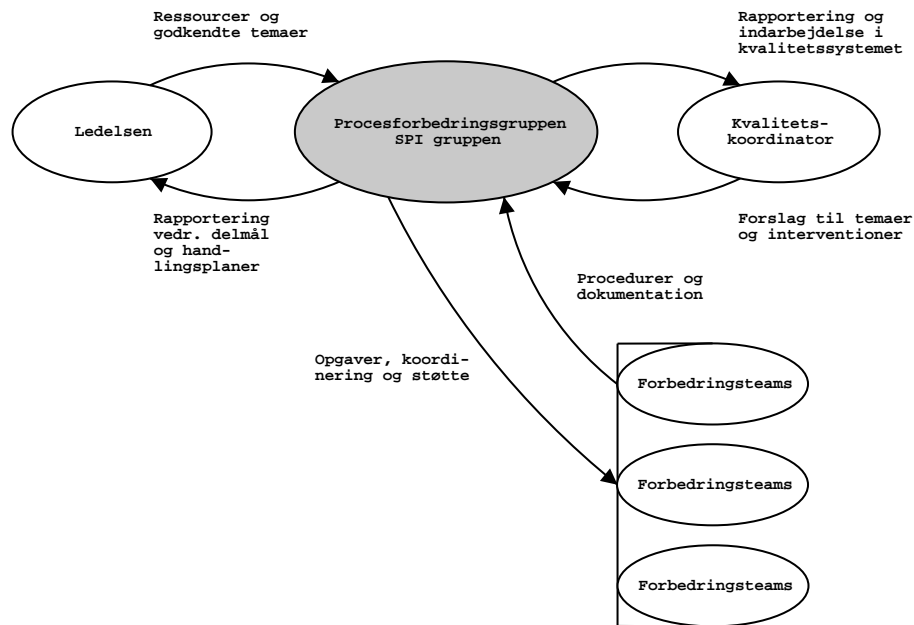
- Dennis Andersen (formand), projektleder i MARS implementering.
- Lars Riisberg (tidligere formand), funktionsansvarlig for MARS og vicedirektør.
- Brian Pedersen, projektleder IT-Solutions.
- Finn Hedehus, funktionsansvarlig IT-Solutions.

Organiseringen af SPI gruppen, hvordan den er forankret i organisationen og dermed også hvordan Logimatic forestiller sig, at interaktionen mellem resten af organisationen og SPI gruppen skal foregå, ses på figur 6-1.

SPI gruppen er en permanent gruppe, hvis arbejdsopgaver består i at forberede og udføre CMM målingerne og efterfølgende konkludere på resultaterne. Ud fra resultaterne udarbejdes der anbefalinger til konkrete procesforbedringer, som udmøntes i handlingsplaner. Derudover er det gruppens opgave at tage initiativ til forbedringsgrupper, koordinere og støtte op samt vejlede dem i deres arbejde.

Forbedringsgrupper nedsættes af SPI gruppen efter behov og består af folk uden for SPI gruppen, men hvor det dog skal tilstræbes, at der i gruppen er en repræsentant fra SPI gruppen til stede. Forbedringsgrupperne får tildelt en opgave pr. gruppe, hvis opgave det er at planlægge og udføre forbedringsinitiativer som fastlagt i en handlingsplan udarbejdet af SPI gruppen. Gruppen udarbejder derefter procedurer og dokumentation, som afleveres til SPI gruppen, som dernæst assisterer kvalitetskoordinatoren med at indarbejde resultaterne i kvalitetssystemet.

¹Oplyst af Logimatic under indledende møder i Logimatic 2002.



Figur 6-1: Organisationsdiagram over SPI gruppen.

Ledelsen er sponsor for SPI initiativet og stiller ressourcer til rådighed for SPI gruppen. Samtidig forventes det, at SPI gruppen fremkommer med forslag til procesforbedringstemaer, som koordineres og afstemmes med ledelsen. Desuden skal SPI gruppen rapportere status for realiseringen af delmål og handlingsplaner til ledelsen.

Kvalitetskoordinatoren er bindeledet mellem SPI gruppen og kvalitetssystemet, hvis resultaterne af procesforbedringsinitiativerne har relation til virksomhedens kvalitetssystem. Derfor rapporterer SPI gruppen løbende til koordinatoren, og assisterer denne i forbindelse med indarbejdelse af resultaterne i kvalitetssystemet. Kvalitetskoordinatoren har også mulighed for at få indflydelse på, hvilke temaer og interventionsområder, der skal udvælges, da vedkommende har kompetence til at foreslå temaer til SPI gruppen. Det kan være i forhold til områder i kvalitetssystemet, som koordinatoren igennem sit arbejde har fundet specielt problematiske eller interessante.

6.2 Vurdering af SPI indsatsen

SPI indsatsen i Logimatic foregår hovedsageligt i SPI gruppen, hvilket harmonerer meget godt med den traditionelle SPI tilgang. Fremgangsmåden følger IDEAL modellen hvor situationen først analyseres. Derefter designes og dokumenteres processerne i forbedringsgrupper, de afprøves i pilotprojekter for til sidst at blive udrullet i resten organisationen. Forslag til temaer og interventioner kommer enten fra SPI gruppen eller kvalitetskoordinatoren, og double-loop læringen foregår udelukkende i SPI-gruppen eller i forbedringsgrupperne. Det betyder, at det er i SPI gruppen, der reflekteres over, hvordan processerne skal være. Medarbejderne, derimod, korrigerer deres handlinger i forhold til kvalitetssystemet, hvorved der sker single-loop læring. Ligesom traditionelt SPI er Logimatics SPI tilgang præget af, at der tages udgangspunkt i CMM normen og i ISO 9001:94

standarden. Dog er det især ISO standarden, der præger organisationen.

Derudover er SPI indsatsen i Logimatic mere eller mindre gået i stå på grund af manglende ressourcer, og målet CMM niveau 3 lader ikke til at være opnåeligt inde for den tidsramme, som virksomheden har opstillet. Det er vores vurdering, at SPI indsatsen i øjeblikket ikke er så højt prioriteret, men hvis andre konkurrenter begynder at interessere sig for deres nichenmarked, kan det i sidste ende blive en konkurrenceparameter. Det skal ses i lyset af, at en væsentlig del af deres indtjening alene kommer på baggrund af deres MARS produkt.

I tabel 6-1 har vi identificeret de egenskaber, som kendetegner Logimatics SPI tilgang. Det er tydeligt, at Logimatics tilgang i høj grad ligner den traditionelle tilgang med de samme kendetegn.

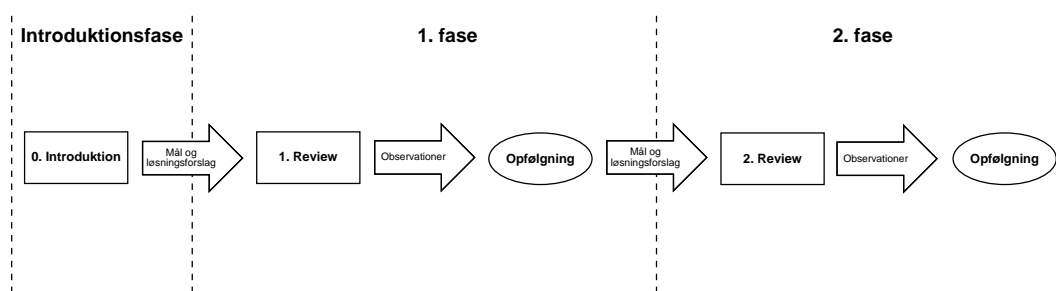
Egenskaber	Kendetegn
<i>Problemerkendelse</i>	SPI gruppen kommer med problemer.
<i>Procesejerskab</i>	SPI gruppen kommer med løsninger.
<i>Planlægning</i>	Planlagt SPI forløb.
<i>Design</i>	Design af processer.
<i>Dokumentation</i>	Konstrueres på et bestemt tidspunkt.
<i>Normer</i>	Normer er udgangspunktet for SPI.
<i>Fokus</i>	Fokus er på processer.
<i>Læring</i>	Double-loop læring hos SPI gruppen.

Tabel 6-1: Logimatics SPI tilgang.

Dette kapitel præsenterer vores indledende forberedelse i forbindelse med interventionen i Logimatic. I kapitlet vil gruppen, som blev udvalgt til interventionen, blive præsenteret. Desuden vil det planlagte forløb af interventionen kort blive skitseret og vores forbedringsmodel fra figur 4-2 vil blive revideret til at afspejle de faktiske aktiviteter, som de blev gennemført. Til sidst vil vi kort gennemgå teorien for reviews, da den danner grundlag for vores observationer. Efter at have læst kapitlet vil læseren have en forståelse af den planlagte intervention.

7.1 Forløbet hos Logimatic

I forbindelse med interventionen i Logimatic udførte vi fem aktiviteter, og forløbet af de fem aktiviteter er illustreret på figur 7-1 og vil blive forklaret i det følgende afsnit.



Figur 7-1: Forløbet af interventionen hos Logimatic.

7.1.1 Introduktionsfasen

Introduktionsfasen var det første led i vores model for selvforbedrende projektgrupper, som vi introducerede i figur 4-2 på side 46. I denne fase ønskede vi at præsentere tankerne bag selvforbedrende projektgrupper over for Logimatic samt at få fastlagt en protokol for det videre forløb. På mødet deltog foruden os selv to medlemmer fra SPI gruppen og to medlemmer fra den projektgruppe, som vi skulle samarbejde med. Vi havde på forhånd fremsendt artiklen [Aae02], som er udgangspunktet for vores tanker omkring en alternativ SPI tilgang.

Vi startede mødet med at blive introduceret for projektgruppen, som bestod af fire personer, der skulle foretage review over to omgange på to forskellige projekter. Gruppen til første review bestod af:

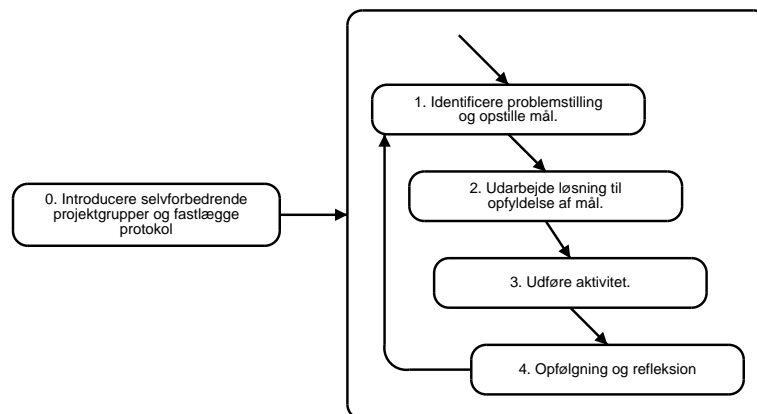
- Dorte Pedersen - Projektleder i Udvikling (Standard MARS).
- Annette W. Sørensen - Gruppeleder i Service & Support.

- Michael Gottlieb - Gruppelider i Udvikling.
- Klaus S. Nielsen - Projektleder i Udvikling.

Derudover skulle Lars Børglum, gruppeleder i Implementering, deltage i det andet review, da han skulle være projektleder på et lignende projekt senere hen. Repræsentanterne fra projektgruppen gav udtryk for, at de så nogle problemer i forbindelse med deres reviewaktiviteter, og dem ønskede de at få løst. Det overordnede problem var, at det var problematisk blot at få aktiviteten afholdt. Det blev derfor aftalt med projektgruppen, at vi skulle deltage i begge reviews, hvor det første var på et foreløbigt designdokument til et internt udviklingsprojekt, og det andet et review på et færdigt designdokument til et kundeprojekt, begge dele inden for standardproduktet MARS. Det skal i denne sammenhæng bemærkes, at der indenfor MARS området ikke skelnes kraftigt mellem en kravspecifikation og et designdokument. De indledende krav er specificeret i en CP¹, og er derefter beskrevet mere detaljeret i et designdokument, som implementationen tager udgangspunkt i.

Interventionsmodellen

Efter den indledende introduktion præsenterede vi vores forbedringsmodel over for mødedeltagerne. Dog skal det bemærkes, at vi havde udeladt fase syv i modellen på figur 4-2 på side 46, da vi på forhånd havde valgt at se bort fra eksternaliseringsfasen. Det afledte nogle spørgsmål til modellen, men ellers lod det til, at der var opbakning til at prøve en alternativ SPI tilgang. Dog mente mødedeltagerne, at der var lige rigeligt med møder, og derfor blev den oprindelige model skåret ned fra de planlagte syv aktiviteter til fem aktiviteter, og den nye model fremgår af figur 7-2. I forhold til den gamle model blev aktivitet 1 og 2 lagt sammen til aktivitet 1 i den nye model. Aktivitet 3 blev bibeholdt, men fik nummer 2 i stedet for. Aktivitet 4 udgik da vi blev enige om, at aktiviteten godt kunne undværes, og aktivitet 5 og seks blev til henholdsvis aktivitet 3 og 4.



Figur 7-2: Den faktiske model for interventionen, som vi kom frem til på det indledende møde med projektgruppen og SPI gruppen.

Forholdet til Kolbs læringscyklus

Hvis vi skal forholde modellen i figur 7-2 til læringsteorien, kan vi se, at der er en meget god overensstemmelse mellem vores model og Kolbs læringscyklus på figur 2-3 side 16. Første element i

¹Change Proposal, det der svarer til kundens krav.

Kolbs læringscyklus starter med et spørgsmål eller problem, hvilket er ensbetydende med, at der allerede er sket en *refleksion*. Næste element handler om at *tænke*, hvor der gennem analyse skabes overblik over problemet, hvilket harmonerer meget godt med aktivitet 1 i vores model. *Beslutte* er næste element i Kolbs læringscyklus, hvor der træffes beslutning om hvordan problemet skal løses, hvilket er aktivitet 2 i vores model. Derefter skal løsningen føres ud i livet, hvilket i vores model sker i aktivitet tre, og som Kolb har som elementet *handle*. Det sidste element i Kolbs læringscyklus drejer sig om at *reflektere*, hvilket er den systematiske evaluering af handlingens konsekvens, og som skal kaste værdi over værdien og effekten af indholdet i de foregående elementer. I forhold til vores model foretager vi denne aktivitet i aktivitet 4.

I forhold til vores interventionsforløb, havde vi allerede overstået aktivitet 0 i vores interventionsmodel, og nedenunder har vi kort skitseret det planlagte forløb i de to næste faser.

7.1.2 Første fase

Første fase af intervention indeholdt to aktiviteter, hvor den første aktivitet var reviewet af design-dokumentet på det interne udviklingsprojekt og var fastlagt til d. 8. april. Inden reviewet havde vi fremsendt CMM mål for reviews, som projektgruppen havde taget udgangspunkt i ved reviewet. Under denne aktivitet observerede vi, hvordan gruppen foretog reviews, og observationerne anvendte vi på opfølgningen på første review, som fandt sted d. 9. april. Denne opfølgning havde til formål at gennemføre aktivitet 1 og 2 i interventionsmodellen, dvs. at få gruppen til at *tænke* over problemerne ved deres reviews og få opstillet mål for reviews. Efterfølgende skulle de selv *beslutte*, hvordan de ønskede at opfylde målene.

7.1.3 Anden fase

Anden fase indeholdt også to aktiviteter, hvor den første aktivitet bestod af andet review af design-dokumentet på kundeprojektet og en efterfølgende opfølgning. Deltagelse på reviewet havde til formål, at vi ville observere hvilke forbedringer, der skete i forhold til første review, og det var i denne fase reviewgruppen skulle *handle*. Reviewet fandt sted d. 19. april. På den efterfølgende opsamling, som fandt sted d. 29. april skulle reviewgruppen vurdere hvilke forbedringer, der var sket, dvs. *reflektere*, hvilket svarer til aktivitet 4 i interventionsmodellen. Desuden skulle deltagerne komme med kritik på vores foreslåede tilgang til procesforbedring gennem selvforbedrende projektgrupper.

7.2 Reviews

I dette afsnit vil vi præsentere reviews ud fra en teoretisk synsvinkel, så vi senere kan styre vores intervention i Logimatic ud fra den gængse opfattelse af, hvad reviews er, og ikke blot vores egne ideer til det. Dette vil gøre det lettere at identificere problemer ved den måde, hvorpå Logimatic udfører reviews, og det vil også gøre det lettere at komme med vejledning til projektgruppen. Den teoretiske fremstilling er bygget på Roger S. Pressmans bog om Software Engineering [Pre00].

7.2.1 Formål med reviews

Et formelt review er en kvalitetssikringsaktivitet, der sigter mod:

- At finde fejl i funktionalitet, logik eller implementation,
- at verificere at produktet overholder de opstillede krav,
- at sikre at produktet er udarbejdet i henhold til prædefinerede standarder,
- at opnå en ensartet udvikling af produktet og
- at gøre projektet lettere at styre.

Derudover kan reviews være med til at sikre kontinuitet i projektet, da flere personer får kendskab til samme område. En anden fordel ved reviews er, at udviklerne lærer af hinanden, så dem uden så meget erfaring kan få indblik i andre metoder og lære ud fra det.

7.2.2 Reviewforløbet

Fokus på et review skal være på et konkret arbejdsprodukt, det kan være et designdokument, noget kode eller noget helt tredje. Producenten, som er den, der har stået for udarbejdelsen af arbejdsproduktet, efterspørger et review, hvorefter projektlederen udpeger en reviewleder, som finder de personer, der skal være med til at reviewe, og distribuerer de nødvendige materialer til dem. Herefter forventes det, at hver reviewer bruger tid på at reviewe arbejdsproduktet og på at lave noter. I mellemtiden forbereder reviewlederen en agenda for reviewet.

På selve reviewet deltager reviewlederen, producenten og de personer, der er udpeget til at deltage som reviewere. En af reviewerne tager rollen som referent, og har ansvaret for at nedskrive alle vigtige problemer og spørgsmål, der dukker op i løbet af reviewet. Reviewlederen præsenterer agendaen, og producenten giver en kort introduktion til arbejdsproduktet, hvorefter vedkommende går gennem arbejdsproduktet og forklarer det undervejs. Reviewerne påpeger problemer og kommer med spørgsmål når det er relevant, og når disse problemer og spørgsmål er reelle bliver de noteret af referenten.

Inden reviewet er slut skal deltagerne beslutte om arbejdsproduktet skal godkendes uden yderligere rettelser, om det skal afvises på grund af væsentlige fejl eller mangler, eller om det skal godkendes med forbehold for, at de mindre fejl, der er opdaget, bliver rettet. Endeligt skal deltagerne skrive under på, at de har været til stede, og at de er enige i de resultater, der blev noteret.

Referentens noter sættes sammen til en liste over problemer og spørgsmål, som indgår i en sammenfatning af reviewet, der beskriver hvad der blev reviewet, hvem der gjorde det og hvad resultatet af reviewet blev. Det er vigtigt at have en opfølgningprocedure, så det sikres, at alle punkter på listen bliver rettet. Dette kan fx ske ved at reviewlederen også sørger for at følge op på reviewet.

7.2.3 Retningslinjer

For selve reviewet er der nogle retningslinjer, som skal være på plads inden reviewet. Alle deltagere skal være enige om, at disse retningslinjer er gældende, og at de vil følge dem bedst muligt. Følgende er minimale retningslinjer for et review:

- Reviewet drejer sig om **produktet**, ikke producenten. Tonen bør altid være konstruktiv, og meningen skal aldrig være at ydmyge. Det er reviewlederens ansvar at sikre, at tonen er passende, og afbryde reviewet hvis det løber løbsk.

- Opstil en **dagsorden** og overhold den. Som med alle andre typer af møder, så er en af de store farer, at diskussionen kommer ud på et sidespor, og det er reviewlederens ansvar at sørge for, at det ikke sker, også selv om det betyder, at diskussioner bliver stoppet med hård hånd.
- **Begræns diskussionerne.** Et problem, der påpeges af en reviewer, har muligvis ikke en løsning, der tilfredsstillende alle, men fremfor at spille reviewtid med at diskutere det, så bør problemet noteres og diskuteres andetsteds.
- **Påpeg problemer**, men kom ikke med løsninger på dem alle. Formålet med et review er ikke at løse problemer, men at identificere dem. Ofte kan de løses af producenten alene, eller med hjælp fra en eller to andre.
- **Tag noter.** Ofte kan det være en god idé hvis referenten bruger en tavle til noter, så alle er enige i formuleringer og prioriteringer.
- **Begræns antallet af deltagere** og sørg for, at alle er forberedte. For mange deltagere forsinker processen, uden at det giver nogen positiv indvirkning.
- **Lav checklister** for de produkter, der skal reviews. Det hjælper til at styre reviewet, og hjælper også reviewerne til at fokusere på de vigtige elementer.
- **Afsæt tid og ressourcer** til reviews, det gælder både tid til forberedelse, tid til selve reviewet og opfølgingsaktiviteter. Den afsatte tid til henholdsvis forberedelse og udførelse af reviewet bør ikke overstige 2 timer for hver af aktiviteterne.
- **Gennemfør uddannelse** af reviewere. For at lave effektive reviews er det nødvendigt, at reviewerne kender til både de tekniske aspekter og de menneskelige aspekter ved et review.
- **Review reviewprocessen.** Det kan være gavnligt at reviewe selve processen, og det første der skal reviews kan meget vel være retningslinjerne for et review.

Som allerede nævnt, så er dette nogle minimale retningslinjer, som bør tilpasses den enkelte virksomhed. I det følgende vil vi bruge disse retningslinjer til at identificere problemområder i Logimatics reviewproces.

I de to følgende kapitler vil vi kort beskrive aktiviteterne objektivt og derefter komme med vores vurdering af hver af aktiviteterne. For at sikre os, at gennemgangen er konsistent og dækkende, vil den tage udgangspunkt i den teoretiske beskrivelse af et review. De objektive beskrivelser af de to reviews og de opfølgende møder har været fremsendt til de involverede parter med henblik på gennemlæsning og kommentering. Herved har deltagerne haft mulighed for skriftligt at komme med kommentarer til beskrivelserne af forløbet. Disse kommentarer er efterfølgende blevet indført i de enkelte beskrivelser, hvorved vi har sikret triangulering af empirien. Dvs. at beskrivelserne af forløbet afspejler både vores og deltagernes opfattelse af forløbet.

Dette samt det efterfølgende kapitel beskriver interventionen i Logimatic, og fungerer som det empiriske grundlag, som vi senere vil analysere og vurdere selvforbedrende projektgrupper ud fra. Kapitlerne beskriver de problemer, der er ved Logimatics reviewproces, hvordan nogle af problemerne er løst, og hvordan projektgruppen fungerer i forhold til at forbedre sig selv. De vigtigste iagttagelser for første del af interventionen, som vil danne det primære grundlag for den efterfølgende analyse, og er opsummeret i tabel 8-9.

8.1 Første review

Formålet med det første review var at give os som observatører en mulighed for at overvære, hvordan reviews normalt gennemføres i Logimatic, så vi lettere kunne stille spørgsmål og komme med råd og vejledning senere. Selv om vi kender til de mere teoretiske aspekter af reviews, er det ofte tilfældet, at reviews i praksis gennemføres på en anden måde og med andre formål end dem, der er nævnt i teorien.

Som forberedelse til reviewet havde vi studeret CMM normen, og fundet frem til de mål, som CMM opstiller for et review af en kravspecifikation, se tabel 8-1. Målene er opstillet på baggrund af de aktiviteter, CMM beskriver for kravstyring, nemlig at manglende krav skal identificeres, og det skal afgøres, om kravene er implementerbare, klart formulerede, konsistente og verificerbare. Disse mål havde vi sendt videre til projektgruppen, der kunne bruge dem som inspiration, dog uden at der blev stillet noget krav om det. Tidligere havde projektgruppen også fået en beskrivelse af den metode, vi havde valgt at anvende, og som er vist på figur 7-2, så de var forberedte på, hvad hele forløbet indebar. Indkaldelsen til reviewet kom fra projektlederen fra Standard MARS og indeholdt designdokumentet, der var udgangspunktet for reviewet. Der var ikke nogen dagsorden for reviewet ud over, at designdokumentet skulle gennemgås.

CMM mål
- Identificere manglende krav.
- Afgøre om kravene er: <ul style="list-style-type: none">- Implementerbare.- Klart formulerede.- Konsistente med hinanden.- Verificerbare.

Tabel 8-1: CMM mål for review.

Reviewet blev afholdt d. 8. april, og der var afsat to timer til det. Deltagerne på reviewet var de førnævnte medlemmer af gruppen, som skulle forstå reviewet. Projektlederen fra Standard MARS havde rollen som mødeleder, og projektlederen fra Udvikling var producent på designdokumentet

og havde ansvaret for at indføre rettelserne til dokumentet. Det er væsentligt at bemærke, at denne rollefordeling ikke var beskrevet eksplicit, men den var underforstået blandt deltagerne. Ud over de nævnte personer deltog vi sammen med Finn Hedehus, der er medlem af SPI gruppen, som observatører. Det vil under normale omstændigheder ikke være nødvendigt for SPI gruppen altid at være til stede under de normale projektaktiviteter, da det vil tage for meget tid, og da SPI gruppen i princippet allerede er bekendt med de gængse arbejdsmetoder i virksomheden, eller i det mindste med den kultur der hersker.

Den foreløbige udgave af designdokumentet, der var blevet fremsendt inden reviewet, var udgangspunktet. Det fremgik dog ikke, hverken inden eller under reviewet, hvad den overordnede målsætning dækkede over, altså hvad der konkret skulle reviewes i dokumentet, fx om reviewet havde til hensigt at klarlægge, hvorvidt der manglede beskrivelser af funktionalitet.

Selve reviewet blev igangsat af mødelederen. Da der ikke forelå en dagsorden, blev der i starten diskuteret, hvordan reviewet skulle foregå. Selv om det ikke fremgik klart blev fremgangsmåden, at først skulle der være generelle kommentarer til dokumentet, og derefter skulle hvert afsnit gennemgås i detaljer.

På trods af, at de overordnede kommentarer skulle gennemgås først, endte diskussionen hurtigt inden for et specifikt område af ændringerne. Diskussionen gik på, hvordan en ændring, som var beskrevet i designdokumentet, ville have indflydelse på den øvrige funktionalitet. Denne diskussion tog ca. halvdelen af den afsatte tid, da der var stor uenighed blandt deltagerne om, hvordan funktionaliteten ville blive påvirket.

I den sidste halvdel af reviewet blev de øvrige ændringer gennemgået. Her blev der sprunget mellem de enkelte afsnit i dokumentet, og der blev ofte gået tilbage til tidligere diskuterede afsnit. På trods af, at mødet overskred den afsatte tid med 15 minutter, nåede deltagerne kun at gennemgå halvdel af designdokumentet på reviewet.

I løbet af reviewet tog specielt mødelederen og producenten notater af de beslutninger, der blev truffet. Ved reviewets afslutning ridsede mødelederen kort op, hvilke beslutninger hun havde noteret, og hun sikrede sig, at der var overensstemmelse med de øvrige deltagere. Herefter var det producentens ansvar at indføre beslutningerne i designdokumentet.

Umiddelbart efter reviewet havde vi et kort møde med repræsentanten fra SPI gruppen, for at se om vi havde observeret nogenlunde det samme. Det viste sig at være tilfældet, han syntes også at reviewet virkede ustruktureret og havde også troet, at det ville være mere dokumentnært. Tabel 8-2 opsummerer, hvad vi observerede til det første review hos Logimatic.

Observationer under første review
<ul style="list-style-type: none"> - Formålet med reviewet var kendt for deltagerne. - Arbejdsfordelingen var kendt. - Ingen reel dagsorden. - Diskussion af specifikke områder i stedet for generelle emner. - Tog lidt længere tid end hvad der var afsat. - Ikke hele designdokumentet blev gennemgået. - Intet referat.

Tabel 8-2: Resultater af deltagelse i det første review.

8.2 Vurdering af første review

Formålet med det første review var at etablere en forståelse af, hvad et review er for Logimatic, og hvilke problemer det medfører. På baggrund af det tilsendte materiale havde vi forventet et mere traditionelt review, hvor det var designdokumentet, der var i centrum, men det viste sig, at formålet primært var at få diskuteret de overordnede linjer i dokumentet.

Vores vurdering af reviewet er, at det bar præg af at være ustruktureret. Årsagen til dette bliver klar ved at vurdere reviewet i forhold til retningslinjerne fra afsnit 7.2.3, hvilket vil blive gjort i det følgende. Resultatet af dette er vist i tabel 8-3, der viser Logimatics review i forhold til retningslinjerne.

Det var positivt, at reviewet var fokuseret på at *reviewe produktet* og ikke producenten af produktet. Kommentarerne blev aldrig til personlige angreb på trods af, at det godt kan være svært at komme med kritik på et produkt, uden at producenten tager det personligt.

En *dagsorden* for mødet ville kunne have hjulpet med til at sikre en vis struktur. Hermed ville mødelederen have haft noget at støtte sig til, når reviewet skulle styres. Et godt eksempel på dette er, at selvom det var blevet aftalt, at de overordnede kommentarer skulle gennemgås først, endte diskussionen hurtigt med at være meget specifik, og der blev ikke fokuseret på blot at *påpege problemer*. I stedet blev mulige og umulige løsningsforslag også diskuteret, noget der endte med at tage meget af tiden. Her kunne checklister også have hjulpet med til at holde fokus på de rigtige steder og ville have været en hjælp i forberedelserne til reviewet. Hvis deltagerne på forhånd havde vidst hvilke elementer, der var vigtige, kunne de have koncentreret sig mere om dem.

Derudover var det tydeligt, at reviewet ikke var et review af selve designdokumentet, men nærmere et review af de overordnede principper. Det betød, at der var nogen usikkerhed omkring, hvad der skulle implementeres og hvorfor. Eftersom projektet var et internt udviklingsprojekt på standardproduktet MARS, var der ingen direkte kunde, der stillede kravene, derimod kom kravene fra Produktrådet for MARS, som de tre af deltagerne sidder i. De specifikke krav blev ikke holdt op mod designdokumentet, og det gav anledning til *længere diskussioner*, der til sidst endte med, at forslaget blev afvist på baggrund af, at det ikke var et krav fra Produktrådet. Sådanne lange diskussioner kunne muligvis være undgået ved tidligere at tage fat på de krav, som var opstillet fra Produktrådet.

Nogle af de ting, der fra projektgruppen var blevet nævnt som problemer på et af de indledende møder var, at der i det hele taget var problemer med at få afholdt reviews i første omgang, og at deltagerne ikke altid mødte velforberedte frem, og at der foregik meget småsnak ind i mellem. Dette var problemer, som vi ikke oplevede, men grunden kunne være, at det var fordi, vi var til stede, og det derfor fik deltagerne til at fokusere mere. Den manglende forberedelse kan hænge sammen med, at der ikke er *afsat tid* til reviews og forberedelse til reviews i projektplanen, lige som problemet med irrelevant småsnak kan hænge sammen med en manglende agenda, og at der ikke er udarbejdet *checklister*, der kan sikre, at fokus bliver holdt på de relevante steder.

Der blev ikke taget fælles *noter* til reviewet, så der kunne let opstå tvivl om, hvad der var blevet besluttet, og om alle var enige. Det er både et problem for producenten, der senere skal rette designdokumentet til, ligesom det kan være et problem for de andre deltagere, og det er let at komme til at diskutere de samme emner igen.

Retningslinjer for review		
<i>Retningslinje</i>	<i>Overholdt</i>	<i>Observation</i>
<i>Review produktet</i>	+	Der var ingen personlige angreb eller kritik.
<i>Dagsorden</i>	–	Der var ingen dagsorden, der kunne følges.
<i>Begræns diskussioner</i>	–	Mange diskussioner, der ikke var relateret direkte til reviewet.
<i>Påpeg problemer</i>	–	Der blev brugt en del tid på at diskutere alternative løsninger og strategier.
<i>Tag noter</i>	+	Producenten tog noter, men der var ikke udpeget en referent.
<i>Begræns antal</i>	+	Der var et passende antal reviewere til stede.
<i>Lav checklister</i>	–	Ingen checklister.
<i>Afsæt tid og ressourcer</i>	–	Tiden blev taget fra andre aktiviteter.
<i>Gennemfør uddannelse</i>	–	Ingen har fået uddannelse i hvordan reviews afholdes.
<i>Review reviewprocessen</i>	–	Sker ikke.

Tabel 8-3: Retningslinjer for review i forhold til Logimatics review.

Opsummering

Som tidligere nævnt, viser tabel 8-3 det første review i forhold til de teoretiske retningslinjer. Tabellen viser klart, at der er mange punkter, der mangler at blive opfyldt. Det overordnede problem syntes at være manglen på struktur. Det kan ledes tilbage på den manglende agenda, der gjorde det uklart, hvad formålet med reviewet var. At der ikke blev lavet noget referat gjorde, at det blev uklart, hvad der var blevet besluttet. Tiden blev godt nok overskredet, men det betyder ikke, at der ikke var afsat nok tid, det er snarere et resultat af, at diskussionerne gik over til at være reelle løsningsforslag i stedet for at koncentrere sig om at identificere problemer. Det er også sandsynligt, at brug af checklister ville have gjort reviewet mere struktureret, da det på den måde ville være lettere at se, hvad der manglede, og hvad der allerede var blevet gennemgået.

På grund af disse mangler blev reviewet ustruktureret, og reviewlederen havde ikke nogen videre mulighed for at rette op på det, da de nødvendige midler, som fx dagsorden og et klart formuleret formål, ikke var til stede.

8.3 Opfølgning på forløbet

Formålet med opfølgningen på første review var at gennemføre aktivitet et og to fra figur 7-2. Det handlede primært om at få identificeret problemer og få opsat mål for næste review samt at komme med løsningsforslag til, hvordan målene kunne opnås.

Opfølgningen blev holdt tirsdag d. 9. april, dagen efter afholdelsen af reviewet, og der var afsat 1 time til formålet. Til opsamlingen deltog to medlemmer af projektgruppen samt en observatør fra SPI gruppen. SPI gruppen var repræsenteret for at lære, hvordan den kan støtte op omkring

selvforbedrende projektgrupper. Fra projektgruppen var det projektlederen fra Standard MARS og gruppelederen fra Service & Support, der var til stede. Den anden gruppeleder og projektlederen, som havde været producent på designdokumentet, var ikke til stede ved opfølgningen.

Dagsordenen for opfølgningen var blevet udsendt af os og indeholdt de punkter, vi skulle igennem på opfølgningen. Som nævnt, drejede det sig om først at identificere problemer ved reviewet, og derefter at opstille mål for næste review. Disse ting blev samlet op på tavlen, hvor vi havde stillet et skema op med problemer, mål og løsninger, som blev udfyldt løbende. Ud over en dagsorden og de tidligere fremsendte CMM mål for reviews havde deltagerne på opfølgningsmødet ikke fået yderligere materiale.

Mødet startede med, at projektgruppen blev bedt om at komme med deres mening om det netop afholdte review, se tabel 8-4, der kort opsummerer problemerne i den rækkefølge, de blev nævnt til opfølgningen. Det største problem viste sig at være, at reviewet var for ustruktureret, hvilket passer godt overens med vores iagttagelser. Der var tvivl om, hvorvidt hele designdokumentet var blevet gennemgået, og om alle punkter blev gennemgået tilfredsstillende. Som en mulig årsag til den manglende struktur blev manglen på en dagsorden nævnt. På trods af at det for os var lidt uklart, hvad formålet med reviewet var, så mente gruppen ikke, at det var noget problem, da alle i gruppen var klar over, at reviewet ikke var et review af selve dokumentet, men nærmere af de tanker der lå bag.

Noget andet, der havde undret os lidt, var manglen på opfølgning. Der blev ikke udarbejdet noget referat af reviewet, på trods af, at både mødelederen og producenten af designdokumentet tog noter. Da vi spurgte, om det ikke ville være en god idé at sende referat ud til alle deltagerne, så alle havde en fælles opfattelse af, hvad der skulle ændres, var svaret, at det i denne forbindelse ville være for formelt.

En ting, der ikke helt var enighed om, var, om der blev gået for meget i detaljer med nogle af punkterne, så der blev diskuteret irrelevante detaljer. På den ene side er det meget rart at få snakket nogle af detaljerne igennem, men på den anden side er et review måske ikke det helt rigtige sted, da der går meget tid med det.

Derudover blev der nævnt en mere overordnet problemstilling, nemlig at der ikke altid er afsat tilstrækkelig tid og ressourcer til, at det er muligt at sætte sig ind i de dokumenter, der skal reviews. Dette kan gøre udbyttet af reviews væsentligt mindre. Et andet potentielt problem af samme type er, at reviews ikke er udspecificeret som en aktivitet i projektplanerne på trods af, at alle ved, at de skal afholdes. Et større fokus på review som en selvstændig aktivitet ville muligvis give et større udbytte.

På den positive side blev nævnt, at alle var velforberedte, og diskussionerne blev aldrig personlige, men var bygget på tekniske argumenter fremfor holdninger. Det bevirkede, at der ikke var nogen dele af reviewet, der blev opfattet som komplet overflødige, da stort set det hele havde et teknisk indhold. Der blev heller ikke gået unødigt i detaljer med fx kommafejl og formuleringsfejl, hvilket dog til dels skyldes, at formålet med reviewet ikke lagde op til den slags kommentarer.

I forhold til retningslinjerne fra tabel 8-3, er det ikke alle punkterne, der direkte er dækket i tabel 8-4, men struktur blev nævnt som et overordnet problem, og mange af punkterne i tabel 8-3 vil resultere i ustrukturerede reviews, hvis de ikke er opfyldt.

Problemer ved reviews
<ul style="list-style-type: none"> - Struktur - Manglende dagsorden - For lidt tid til at sætte sig ind i dokumenterne - Der er ikke afsat fast tid til reviews i projektplanen

Tabel 8-4: De problemer som gruppen identificerede ved deres reviews.

På baggrund af de problemstillinger der blev identificeret, og som er vist i tabel 8-4, blev der opstillet en række mulige mål for efterfølgende reviews. Det var både mål i forhold til CMM og i forhold til projektgruppens egne ønsker. De mål, der kom frem, er vist i tabel 8-5. Listen over mål var resultatet af en brainstorm-lignende aktivitet, hvor alle mulige mål blev stillet op. Eftersom struktur var et af de problemer, der blev identificeret, blev der opstillet et overordnet mål om, at reviews skulle være bedre strukturerede. Et forslag til at opnå dette var at opstille en dagsorden for reviews, så alle er enige om, hvad der skal ske. Det kan også være med til at sikre, at producenten får konstruktiv feedback, da en dagsorden er med til at sikre fokus, og at der ikke diskuteres unødvendige emner. Endeligt blev der foreslået nogle mål, der tager udgangspunkt i CMM målene, nemlig at det skal sikres, at designdokumentet opfylder de opstillede krav, dvs. at der ikke mangler noget, og at der rent faktisk kan findes realiserbare løsninger på problemerne.

Overordnede mål
<ul style="list-style-type: none"> - Reviews skal være strukturerede. - Der skal være en dagsorden for reviewet. - Producenten skal have konstruktiv feedback. - Reviews skal sikre opfyldelsen af CPer. - Der skal findes realiserbare løsninger. - Overholdelsen af standarden for dokumentation skal sikres.

Tabel 8-5: De overordnede mål for reviews som gruppen opstillede.

Gennem denne aktivitet blev det klart, at der var forskellige mål, alt efter hvilken type review, der var tale om. Det første review vi var med til handlede primært om at hjælpe udvikleren videre med designdokumentet, og ikke så meget om selve dokumentets indhold og udformning. Det gav anledning til nogle andre mål end det næste review, vi skulle deltage i, hvor det var selve dokumentet, der skulle reviews. Tabel 8-6 viser de mål, gruppen kom frem til skulle gælde for reviews, der ligner det første review. Som det kan ses, handler det i højere grad om at sørge for, at den generelle retning for projektet er i orden. Det vil sige, at reviewet skal sikre, at alle krav er med i designdokumentet, at der ikke er inkonsistens mellem dem, og at der ikke bliver foreslået urealistiske løsninger.

Mål for indledende reviews
<ul style="list-style-type: none"> - Manglende krav skal være identificeret. - Der skal være overensstemmelse mellem kravene. - Løsningerne skal være realiserbare.

Tabel 8-6: De mål som gruppen opstillede for indledende reviews.

Tabel 8-7 viser de mål, projektgruppen mente skulle gælde for et review af et færdigt designdokument, altså et review, hvor der også tages hensyn til form og indhold af selve dokumentet. Disse mål var inspireret af CMM målene for et review, som vi havde fremlagt som inspiration, og handler om at sikre, at kravene i dokumentet er realistiske og implementerbare. Derudover skal kravene være anvendelige for udvikleren, når produktet skal udvikles, og at kunden kan forstå kravene, så kunden kan følge med i processen. Endeligt er det vigtigt, at det færdige designdokument opfylder virksomhedens standard for designdokumenter.

Mål for reviews af færdige dokumenter
<ul style="list-style-type: none"> - Kravene skal være implementerbare. - Kravene skal være forståelige, både for udviklere og kunder. - Kravene skal kunne testes og verificeres. - Standarden for designdokumenter skal være opfyldt.

Tabel 8-7: Målene for reviews af færdige dokumenter, som gruppen opstillede.

Endeligt blev mulige løsningsmodeller diskuteret, uden at der kom noget konkret svar på bordet. I stedet for at foreslå en løsning overlod vi det arbejde til projektgruppen selv, som havde ansvaret for at udarbejde et løsningsforslag, som skulle bruges til det næste review. Tabel 8-8 viser nogle områder, der kunne være med til at forbedre reviewaktiviteterne. En bedre struktur kunne opnås ved at have en dagsorden og et formuleret formål. For at sikre, at hele dokumentet blev gennemgået, kunne gruppen vælge at tage udgangspunkt i de specificerede krav, og så holde dem op mod dokumentet, i stedet for at holde dokumentet op mod kravene. På den måde ville det blive sikret, at alle krav var opfyldt i dokumentet. En mere overordnet forbedring af reviewaktiviteten kunne være at sætte mere fokus på reviews i projektplanen, så der direkte var afsat tid til aktiviteten.

Løsningsforslag
<ul style="list-style-type: none"> - Lav bedre dagsorden. - Tag udgangspunkt i CPerne. - Tag udgangspunkt i dokumentet. - Sæt fokus på reviews i projektplanen.

Tabel 8-8: Gruppens løsningsforslag til forbedring af reviews.

8.4 Vurdering af opfølgningen på første review

Opfølgningen på det første review var den første del af forbedringsprocessen, der er vist i figur 7-2, hvor projektgruppen skulle identificere problemer og finde mulige løsninger. Et umiddelbart problem ved opfølgningen var, at det ikke var hele projektgruppen, der var til stede. To af deltagerne fra reviewet var ikke med til opfølgningen, hvilket kan hindre læringsaspektet i selvforbedrende projektgrupper. Som tidligere nævnt kan det være problematisk, hvis det kun er halvdelen af projektgruppen, der reflekterer over egen arbejdsproces, da det vil være svært at få ført de ting over til resten af gruppen. Det må dog forventes, at de personer der deltog, viderebringer noget af den viden og erfaring de har fået gennem reviewet og den efterfølgende opsamling.

Gennem opfølgningen blev det også klart, at det ikke altid er nogen let opgave at få en gruppe til at erkende deres problemer. Dette kan til dels skyldes, at kun halvdelen af gruppen var repræsenteret,

men selv hvis de havde været der alle sammen, ville det nok ikke have været helt problemfrit. Et af de store problemer for gruppen var at blive enige om, hvad der var problemer, og hvad der ikke var. For eksempel mente en, at der blev gået for meget i detaljer, mens en anden følte at det var helt i orden. Dette viser blot, at der ikke er enighed om målene for et review, selvom deltagerne tidligere havde givet udtryk for det modsatte. Der var dog enighed om, at reviewet kunne have været mere struktureret, men hvordan denne struktur kunne opnås, var der ikke udbredt enighed om.

Det blev rimeligt klart, at det kræver forholdsvist meget af en SPI gruppe at få en projektgruppe til at erkende deres problemer og derefter beskrive dem. Teknikker som SSM og fiskebensdiagrammer kan bruges, men især SSM kan være omfattende, og kan ikke gennemføres forsvarligt på en time, som cirka er den tid, en projektgruppe kan afsætte til formålet. I forhold til selvforbedrende projektgrupper kan det være et problem, især når det kommer til læring. Et af de væsentlige karakteristika ved selvforbedrende projektgrupper er, at deres læring er baseret på double-loop læring, men denne læring er ikke nødvendigvis let at opnå. Et af problemerne i Logimatic kan være den manglende erfaring med at reflektere over egen proces, noget som jo er nødvendigt, hvis double-loop læring skal opnås. Derfor er det vigtigt, at der bliver gjort en indsats for at fremme denne reflektering, så selvforbedrende projektgrupper bliver en reel mulighed.

Efter opfølgningen stod det også klart, at det ikke kan forventes, at projektgruppen selv tager initiativet til at finde problemstillingerne i første omgang. Her er det nødvendigt at SPI gruppen, eller vi i det konkrete tilfælde, sørger for at motivere og sætte processen i gang. Især når det er første gang, en gruppe skal forbedre sig selv, kan det være et problem, da der ikke er nogen tidligere erfaringer, der peger på, at det overhovedet er gavnligt, og dermed kan gruppen godt opfatte det hele som tidsspilde.

En af de ting, vi også blev opmærksomme på var, at der er forskel på hvilke forbedringsinitiativer, der kan foreslås, da det kommer meget an på projektgruppens erfaringer og fordomme. For eksempel var projektgruppen ikke modtagelig overfor forslaget om at lave referater af reviews, da det blev opfattet som værende for formelt. Hvorvidt denne opfattelse er baseret på erfaringer eller på fordomme er uklart, men hvis det viser sig, at den er baseret på fordomme, kan SPI gruppen vælge at lægge vægten på undervisning og forklare fordelene ved netop denne del. Modsat, hvis opfattelsen er baseret på erfaringer, er det nok bedre at søge efter alternative strategier eller forsøge at identificere præcist, hvor det gik galt tidligere.

På trods af, at projektgruppen ikke virkede så motiverede for at forbedre sig selv, havde vi dog alligevel en forventning om, at der ville ske forbedringer til næste review, specielt med hensyn til dagsorden, som blev udnævnt til en af hovedårsagerne til, at reviewet var så ustruktureret som det var. Som udenforstående ville det også være rart, hvis formålet med reviewet var dokumenteret, i det mindste i dagsordenen, men eftersom det ikke blev anset som et problem af projektgruppen, havde vi ikke nogen forventning om, at det var noget, der ville blive gjort noget ved, især ikke når reviews er en intern aktivitet i gruppen.

8.5 Opsummering på interventionen fase 1

Tabel 8-9 opsummerer nogle af de vigtigste iagttagelser, vi gjorde i forbindelse med første del af interventionen. Disse iagttagelser er relateret til selve ideen om selvforbedrende projektgrupper og

Vigtige iagttagelser efter første fase	
<i>Positive</i>	<ul style="list-style-type: none"> - Gruppen kunne identificere problemer. - Procesejerskabet kunne umiddelbart godt flyttes til projektgruppen. - Den uformelle tilgang passede godt til gruppen.
<i>Negative</i>	<ul style="list-style-type: none"> - Der kan være problemer med at reflektere over egen proces. - Hænger i gamle processer og vaner. - Der kan ikke afsættes meget tid til forbedringsaktiviteterne.

Tabel 8-9: Vigtigste iagttagelser i forbindelse med opfølgningen.

På den positive side blev det klart, at på trods af nogle indledende problemer med at få projektgruppen til at identificere svagheder i den nuværende proces, så lykkedes det alligevel for deltagerne at komme frem til stort set de samme punkter, som vi havde observeret. Dermed er et af kravene til en selvforbedrende projektgruppe opfyldt, nemlig at den selv kan identificere problemer. Derudover blev flytningen af procesejerskabet også godt modtaget, projektgruppen kunne se fordelene i selv at have kontrol over processerne og i selv at have indflydelse på hvilke problemer, der arbejdes på at løse. Endeligt virkede den uformelle tilgang, som selvforbedrende projektgrupper lægger op til, meget passende for gruppen, som i forvejen arbejdede uformelt.

Af mere negative aspekter var, at på trods af at gruppen fik identificeret problemer, så tog det alligevel noget tid at få den i gang, og det skete først efter, at der var blevet stillet spørgsmål til processen, fx. hvad årsagen til den manglende struktur kunne være. Nogle forbedringsforslag blev afvist helt, hvilket kan hænge sammen med, at gruppen hænger i gamle processer og vaner og har meget svært ved at frigøre sig. Et eksempel på dette er det manglende referat fra reviewet, som kun manglede, fordi det normalt ikke blev lavet. Endeligt blev det også klart, at det ikke kan forventes, at projektgruppen bruger særlig meget tid på at forbedre sig selv, i hvert fald ikke i første omgang. Det hænger både sammen med at den tid, der skal bruges, skal tages fra det normale projektarbejde, og at fordelene ved forbedringerne endnu ikke har vist sig.

Første del af interventionen handlede om at identificere problemer ved Logimatics nuværende reviewproces og opstille mål for hvilke forbedringer, der kan løse nogle af problemerne. Anden fase havde til formål at undersøge, om nogle af målene blev fulgt og opfyldt og at undersøge, hvordan projektgruppen havde det med at forbedre sig selv. De vigtigste iagttagelser for anden fase er opsummeret i tabel 9-5 side 90.

9.1 Andet review

Formålet med deltagelse i andet review var at se, om projektgruppen havde indført nogle af de forbedringer, der var blevet diskuteret under opfølgningen. Dette skulle være med til at vise, om selvforbedrende projektgrupper er en reel mulighed i Logimatic, og om selvforbedringsprocessen falder nogenlunde naturligt ind i gruppens arbejde.

Indkaldelsen til reviewet kom fra projektlederen fra Standard MARS og indeholdt en dagsorden, der specificerede, hvad der skulle ske. Dette var en af forbedringerne i forhold til det første review, som projektgruppen også selv havde udpeget som en svaghed. Tabel 9-1 viser den udsendte dagsorden, der indeholdt en gennemgang af formålet med reviewet, diskussion af den CP designdokumentet byggede på og til sidst selve reviewet af dokumentet, især med fokus på at checke, at løsninger var realiserbare, at løsninger kan programmeres, at løsninger kan testes og endeligt om designdokumentet lever op til Logimatics standard for designdokumenter.

Dagsorden
- Formålet med reviewet.
- Diskussion af inkluderet CP.
- Review af designdokument.
- Løsningen er realiserbar.
- Beskrivelsen af løsningen er programmerbar.
- Beskrivelsen er testbar.
- Dokumentet følger den interne standard.

Tabel 9-1: Gruppens dagsorden for andet review.

Andet review blev afholdt fredag d. 19 april, hvor der var afsat halvanden time til aktiviteten. Deltagerne på reviewet var stort set de samme som på første review bortset fra at Lars Børghlum, som er gruppeleder på Implementering, også deltog. Det viste sig, at den afsatte tid ikke var tilstrækkelig, hvilket resulterede i, at gruppelederen fra Service & Support og SPI gruppens repræsentant måtte forlade reviewet før det sluttede. Under reviewet havde projektlederen fra Standard MARS rollen som reviewleder og projektlederen fra Udvikling rollen som producent.

En af de ting, vi så som problemer ved første review var, at formålet med reviewet var uklart. Selv om der ikke var fuld enighed om problemets omfang i projektgruppen, kom det alligevel med som et punkt på dagsordenen. Reviewet blev på den måde indledt med, at reviewlederen præsenterede formålet og dermed fastlagde spillereglerne for reviewet. Det blev også lagt fast, at først skulle overordnede kommentarer diskuteres, hvorefter selve dokumentet ville blive gennemgået. Rettelser til selve teksten blev ikke gennemgået, men i stedet blev der opfordret til, at rettelserne blev afleveret til producenten. Det blev også lagt fast, at reviewet ikke var et review af det endelige designdokument som først antaget, men at det var et review af det midlertidige dokument, og at det derfor ikke var nødvendigt at gå i detaljer med dokumentet. Det gjorde også, at de to sidste punkter på dagsordenen ikke blev diskuteret, dvs. at reviewet primært var fokuseret på, om kravene var realiserbare, og om løsningen var programmerbar.

For at sikre en fælles forståelse for hvad det var, der skulle implementeres, blev den bagvedliggende CP kort gennemgået. Her viste det sig, at der var nogle ændringer til den, der ikke var blevet registreret tidligere.

Gennemgangen af designdokumentet holdt sig til dokumentets opbygning uden de store afvigelser, dog var der nogen irrelevant diskussion omkring hvad, der skulle inkluderes i dokumentet, og hvad der hørte til andre steder. En ting vi dog lagde mærke til var, at reviewet i starten blev styret godt af mødelederen, men senere i processen fik deltagerne lov til at køre ud på nogle sidespor, der ikke var så relevante for reviewet.

Tabel 9-2 opsummerer de umiddelbare ændringer, vi observerede i forhold til det første review. Den største ændring var med hensyn til dagsordenen, der denne gang var meget mere detaljeret, og indeholdt adskillige nye punkter. De nye punkter var at fastlægge formålet med reviewet for at styre det bedre og at gennemgå den CP, der lå til grund for designdokumentet. Det hjalp også til at holde reviewet på det rette spor, og gjorde det muligt at undgå nogle diskussioner helt.

Observationer
- Dagsorden med flere detaljer.
- Fastlæggelse af formål.
- Gennemgang af CP.
- Bedre styring.

Tabel 9-2: Observerde ændringer i forhold til første review.

9.2 Vurdering af andet review

Vores forventning til det andet review var, at der ville komme nogle forbedringer, især med hensyn til dagsorden og strukturering af reviewaktiviteten. Det er vores vurdering, at denne forbedring også fandt sted, og at reviewet var mere struktureret end det første. Der var i starten af reviewet et klart fokus på at undgå nogle af de typiske faldgruber som at gå i unødvendige detaljer, og at diskussionerne løber ud på et sidespor. Det betyder, at selv om der ikke var opstillet mål inden for alle retningslinjerne fra afsnit 7.2.3, skete der alligevel forbedringer inden for flere af dem. Dette skyldes primært det overordnede mål om at sikre en bedre struktur og ved at udarbejde en mere detaljeret dagsorden, blev der indirekte taget fat på nogle af de andre punkter.

Ifølge gruppens egne mål skulle reviewet tage udgangspunkt i CPen. Trods dette mål, havde reviewlederen ikke distribueret selve beskrivelsen af CPen sammen med dagsordenen for reviewet. Dette kan have været med til, at reviewdeltagerne ikke har reviewet designdokumentet med henblik på at identificere, om designdokumentet stemte overens med de indledende krav. Vi vurderer, at det ville have været givtigt at fremsende CPen sammen med dagsordenen og designdokumentet, så alle ville have været klar over, hvad der stod i CPen og derfor kunne sikre, at CPens krav var opfyldt.

En ting vi dog stadig kan være nervøse for er, at de nye punkter på dagsordenen måske kun har relevans for de to personer, der har været med til selvforbedringsaktiviteterne tidligere, og som derfor ved, hvorfor punkterne er der. De øvrige deltagere på reviewet kender ikke ræsonnementet bag og kan derfor have svært ved at gennemskue, hvorfor de er med på dagsordenen. Dette viste sig ikke at være noget større problem, men til gengæld kan det være, at de øvrige deltagere gik glip af nogle fordele. Dem der var med til opfølgningen på det første review ved, at det har haft en indvirkning på det andet review at have punkterne med på dagsordenen, men denne viden har resten af gruppen ikke, og kan derfor ikke på egen hånd udbrede erfaringen til andre dele af organisationen.

Opsummering

Tabel 9-3 viser det andet review i forhold til de retningslinjer, der blev præsenteret i afsnit 7.2.3, og i forhold til tabel 8-3, der viser det første review, er den største ændring sket ved agendaen, hvor der i modsætning til første review blev opstillet en mere formel dagsorden. Ligesom til første review var der ikke problemer med, at reviewet blev personligt, der var hele tiden fokus på produktet og ikke producenten. På trods af, at dagsordenen og reviewlederen hjalp til at begrænse nogle af diskussionerne, fik reviewet alligevel lov til at køre ud på nogle sidespor en gang imellem, især hen imod slutningen af reviewet, ligesom der også blev diskuteret en del løsningsforslag i stedet for blot at påpege problemerne. Til gengæld har projektgruppen taget de første skridt med at reviewe selve reviewprocessen, da det jo er en indlejret del af selvforbedringsprocessen. Det betyder, at der stadig er mulighed for, at der vil blive rettet op på de punkter, der stadig mangler, og som gør reviewet mindre optimalt. I forhold til den opstillede dagsorden fra tabel 9-1, henvender punkterne derfra sig primært til opstilling af agenda, men har også nogen indflydelse på at begrænse diskussioner. Ved at opstille et klart formål med reviewet bliver det klart for alle, hvad der skal diskuteres, og derved kan unødvendige diskussioner undgås, eller i det mindste formindskes.

9.3 Opfølgning på forløbet

Opfølgningen på andet review var den sidste større aktivitet i Logimatic. Formålet med opfølgningen var at foretage en refleksion over den udførte reviewaktivitet. Denne opfølgning svarer til punkt 4 i vores metode, som er illustreret på figur 7-2. Desuden havde opfølgningen til formål at evaluere forløbet af vores intervention som helhed og finde ud af, om projektgruppen havde fået noget ud af det.

Det opfølgende møde blev afholdt mandag d. 29. april, og der var afsat en time til aktiviteten. Deltagerne var projektlederen fra Standard MARS og gruppelederen fra Service & Support, som

Retningslinjer for review		
<i>Retningslinje</i>	<i>Overholdt</i>	<i>Observation</i>
Review produktet	+	Der var hele tiden fokus på produktet.
Agenda	+	Dagsorden var sendt rundt, og formålet blev præsenteret ved reviewets start.
Begræns diskussioner	–	Der var en del diskussioner.
Påpeg problemer	–	Der blev brugt meget tid på at diskutere løsninger.
Tag noter	+	Ingen referent, men alle blev opfordret til at aflevere deres noter til producenten.
Begræns antal	+	I alt 5 deltagere.
Lav checklister	–	Ingen checklister.
Afsæt tid og ressourcer	–	Der blev brugt mere tid end afsat, og reviewet varede over to timer.
Gennemfør uddannelse	–	Ingen har fået uddannelse i hvordan reviews afholdes.
Review reviewprocessen	+	Skete gennem selvforbedringsprocessen.

Tabel 9-3: Retningslinjer for review i forhold til Logimatics andet review.

også deltog i de tidligere aktiviteter, to medlemmer af SPI gruppen og os. Dennis Andersen, som er medlem af SPI gruppen, havde ikke deltaget i de tidligere aktiviteter, men dette blev ikke anset som et problem, da diskussionen også skulle dreje sig om, hvordan SPI gruppens rolle burde være i fremtiden. Dagsordenen for mødet tog udgangspunkt i formålet for opfølgningen, som omhandler opfølgning på andet review og en generel opfølgning på forløbet med selvforbedrende projektgrupper.

Opfølgning på andet review

Deltagerne vurderede overordnet, at interventionen havde betydet, at der var blevet en bedre strukturering på reviewaktiviteten. De var enige om, at det andet review blev styret mere, og der blev gennemgået mere end ved det første review. Ifølge den ene deltager var producenten af det reviewede dokument tilfreds med reviewet, da han fik en del konstruktiv kritik. Desuden havde interventionen været med til at sikre, at de to reviews overhovedet blev afholdt, hvilket var et af de største problemer for reviews i Logimatic.

Det blev dog påpeget, at det var et problem, at der var for lidt tid til reviews. Dette kunne ifølge deltagerne skyldes, at det var et stort emne, der skulle gennemgås. Desuden var det et generelt problem med manglende tid under reviews i Logimatic, da der ifølge standarderne for reviews maksimalt må afsættes 2 timer til aktiviteten. Dette blev dog både anset for godt og skidt, da det satte en fast tidsgrænse, der selvfølgelig kunne være lidt i underkanten for nogle reviews. Desuden var den ene af deltagerne i tvivl om, om de var nået rundt om hele designdokumentet under reviewet.

For at sikre en fremtidig forbedring i forbindelse med reviews, blev de to medlemmer fra projektgruppen enige om, at reviewaktiviteterne i fremtiden skal fremgå som selvstændige aktiviteter i projektplanen. Dette var et andet af de løsningsforslag, der blev opstillet i forbindelse med opfølgningen på første review og skal være med til, at reviews bliver mere synliggjorte i Logimatic. Ud

over denne fremtidige forbedring nævnte projektlederen, at estimering kunne være et interessant emne for en fremtidig forbedring. Som opsummering af opfølgningen på andet review er positive og negative aspekter ved reviewet opstillet i tabel 9-4.

Positivt	Negativt
<ul style="list-style-type: none">- Bedre strukturering.- Bedre styring.- Fik gennemgået mere.- Givet konstruktiv kritik.	<ul style="list-style-type: none">- For lidt tid.- Tvivl om alt var blevet gennemgået.

Tabel 9-4: Positive og negative aspekter ved andet review ifølge projektgruppen.

Opfølgning på forløbet som helhed

Deltagerne var positive overfor erfaringerne med selvforbedrende projektgrupper som tilgang til procesforbedring. Specielt blev inddragelse af medarbejderne og det, at gruppen selv sidder med procesejerskabet, nævnt som et stort plus ved denne metode. Deltagerne mente dog, at det ville have været godt med nogle konkrete eksempler at arbejde ud fra, da der skulle findes løsninger på gruppens problemer. Medarbejderne i Logimatic har ikke tid eller overskud til at læse op på teorien indenfor forbedringsaktivitetens område, i dette tilfælde hvad et review er. Her ville det have været godt, hvis der fx var blevet afholdt en forelæsning om emnet, inden selve procesforbedringsaktiviteten blev påbegyndt.

Der var ikke enighed om, hvem der skal stå for sådanne forelæsningsaktiviteter. De to medlemmer af projektgruppen mente, at det var SPI gruppen, der bør stå for undervisningen og dermed inspirere og påvirke gruppernes forbedringsindsats. Derimod mente SPI gruppen ikke, at det nødvendigvis skulle være dem, men at det også kunne være medarbejdere andre steder i Logimatic, som har interesse indenfor det område, der skal procesforbedres. Der var dog enighed om, at det er SPI gruppens opgave og ansvar at igangsætte aktiviteterne omkring selvforbedrende projektgrupper, ellers vil der ikke blive igangsat nogen forbedringsinitiativer.

Med hensyn til hvordan Logimatic i fremtiden vil anvende erfaringerne fra selvforbedrende projektgrupper, havde SPI gruppen internt talt om at indføre selvforbedring som et generelt krav til nye projekter. Dvs. at alle projekter skal tage et område op, som de ønsker at procesforbedre i løbet af projektet, ligesom der er krav om, at projektet udarbejder en projektplan. Der havde bl.a. været forslag om, at forbedringsinitiativer bliver igangsat og taget op under de såkaldte kvalitetsmøder, som snart skal til at køre i organisationen igen. Disse møder skal afholdes med 1-2 måneders mellemrum, og her kan et forbedringsinitiativ passe godt ind. For at få selvforbedrende projektgrupper op og stå og få samspillet mellem projektgrupperne og SPI gruppen til at fungere, er det, ifølge et af medlemmerne fra SPI gruppen, væsentligt, at rollefordelingen bliver synliggjort i organisationen, så det er klart for alle parter, hvilke krav der stilles til hvem. Derfor bør der i fremtiden arbejdes på at få denne rollefordeling op og stå.

En stor del af diskussionen under opsamlingen omhandlede generelt reviews i Logimatic. Denne diskussion omhandlede hvornår de skal afholdes, hvorfor de ikke bliver afholdt, og hvordan det kan sikres, at reviews bliver afholdt. Det blev diskuteret, om det er muligt at opstille regler for,

hvornår det er nødvendigt at afholde reviews. Det er ifølge standarderne i kvalitetssystemet projektlederens ansvar at afgøre, om der skal afholdes review. Eksempelvis er nogle opgaver så små, at det ifølge det ene medlem af SPI gruppen, ikke giver mening at udføre reviews. Der er kun behov for reviews, hvis kompleksiteten er høj. Det blev konkluderet, at det altid vil være en vurderingssag, om der skal afholdes reviews, og at det ikke er muligt at opstille generelle regler.

Som baggrund for at reviews ikke bliver afholdt i Logimatic, blev det nævnt, at der var en opfattelse blandt medarbejderne af, at de bruger en del tid på en aktivitet, som de ikke føler, de får udbytte af. Ifølge et af medlemmerne af SPI gruppen var denne opfattelse den største barriere for en generel indførelse af reviews i Logimatic. Her nævnte det andet medlem, at det i hans afdeling er meget anvendt at udføre uformelle reviews, set i forhold til de standarder der ligger for reviews i kvalitetssystemet. Dette sker ved, at dokumenter bliver distribueret til en til tre andre medarbejdere, der læser dem igennem og kommer med kommentarer.

For at sikre at reviews bliver afholdt, var forslaget om en checkliste oppe og vende et par gange. Det virkede som om, at en checkliste kunne være en understøttende hjælp til reviewaktiviteten. Fx var der flere, der mente, at det kunne være med til at gøre, at reviews hyppigere ville blive afholdt. En af deltagerne mente, at hvis nogle projektgrupper fik gode oplevelser med reviews, ville disse gruppemedlemmer påvirke andre grupper, hvor de også er medlem, til at igangsætte afholdelsen af reviews. Vedkommende anså det dog ikke som en mulighed udelukkende at anvende denne form for udbredelse i organisationen, da det ville tage for lang tid. Ifølge en anden af deltagerne var det også væsentligt, at der blev lagt nogle standarder i kvalitetssystemet for afholdelse af reviews. Hvis der ikke er nogle standarder i kvalitetssystemet, vil nye projektledere ikke følge processen. Det er dog væsentligt, at kvalitetssystemet ikke er så stramt, at det enkelte projekt ikke kan tilpasse sin egen proces. Et af medlemmerne fra SPI gruppen var ikke enig i, at det ville hjælpe at opdatere kvalitetssystemet, og kom i stedet med et forslag om, at forcere reviews i en gruppe for at få gruppen til at indse, at det er konstruktivt at afholde reviews og herved overbevise gruppen om, at det er en god ide at afholde reviews.

9.4 Vurdering af opfølgningen på andet review

Vores opfattelse af, at der var sket forbedringer indenfor nogle af de områder, som projektgruppen havde identificeret, blev bekræftet under opfølgningen på andet review. Denne opfattelse var delt med projektgruppens medlemmer, hvilket var positivt, da et væsentligt aspekt ved selvforbedrende projektgrupper er, at de hurtigt selv ser udbyttet af deres indsats. Desuden var vi meget glade for deltagernes positive tilkendegivelser vedrørende selvforbedrende projektgrupper og denne tilgang til procesforbedring.

Under opfølgningen blev det nævnt, at der var for lidt tid til reviewet, selvom der var afsat halvdelen af timen. Desuden blev det påpeget, at den øvre grænse for reviews på to timer, der var sat i kvalitetssystemet, ikke altid var nok. Hvis denne kritik holdes op imod retningslinjerne for reviews, siger retningslinjerne netop, at varigheden af et review ikke bør overstige to timer. Vi mener, at hvis det ikke er muligt at reviewe det planlagte materiale indenfor den afsatte tid, så bør projektgruppen udvælge dele af dokumentet, og fokusere på dem. Herved kan reviewprocessen af fx et designdokument deles op over flere reviews, i stedet for at forsøge at gennemgå det hele under et review.

Det væsentligste udbytte af opfølgningen i forhold til selvforbedrende projektgrupper var kritikken af, at gruppen godt kunne have tænkt sig nogle eksempler at arbejde ud fra, lige som de godt kunne have tænkt sig en introduktion til teorien indenfor reviews. Vi havde tidligere diskuteret, om vi skulle holde en kort forelæsning for gruppen om den generelle teori indenfor reviews og komme med eksempler på værktøjer, der kan anvendes til at støtte op om reviews. Det blev dog fravalgt, da vi var bange for, at det ville præge de problemer, som projektgruppen identificerede for meget, ligesom det ville have indflydelse på de opstillede mål og løsningsforslag. Under aktiviteterne viste det sig også, at projektgruppen selv var i stand til at identificere mange af de problemer, der bliver nævnt som faldgruber i litteraturen. Vi har dog erfaret, at praktikere gerne vil have noget at arbejde ud fra i stedet for at starte på bar bund hver gang. Dette kan hænge sammen med den tænkemåde, som mange praktikere har, nemlig at alt kan og skal designes, før selve arbejdet kan gå i gang. Der er en frygt for det usikre, og ved at gennemgå en omfattende designfase kan denne frygt overkommes, dog uden at det har nogen videre indvirkning på slutproduktet, da det næsten altid er umuligt at forudse alt. Derfor kan det være lige så godt bare at gå i gang og så ellers være indstillet på, at der kan ske ændringer senere i processen.

Det ene af medlemmerne fra SPI gruppen nævnte, at det var væsentligt, at rollefordelingen for samspillet mellem projektgrupperne og SPI gruppen kom op og stå. Dette harmonerer godt med vores hovedprincipper for selvforbedrende projektgrupper, der er illustreret i protokollen på figur 4-3 side 47, og som blev introduceret for deltagerne på det indledende møde om interventionen. Ifølge protokollen er SPI gruppens rolle at være med til at opstille rammerne og målene for projektgruppens forbedringsindsats samt at komme med råd og vejledning til gruppen. Projektgruppens rolle er derimod at komme med problemerne, spørgsmålene og ideerne for hvilket område, der skal forbedres, og forslagene til hvordan de vil foretage denne forbedring. I forbindelse med vores intervention burde vi nok havde lagt mere vægt på at gøre deltagerne opmærksomme på denne rollefordeling, så der ikke var tvivl om, hvad parterne kunne forvente af hinanden.

Forslaget fra SPI gruppen om, at de i fremtiden har overvejet at indføre bottom-up procesforbedring gennem kvalitetsmøderne i Logimatics projektgrupper, var for os noget overraskende. Vi havde tidligere diskuteret tilgangen med at stille krav om, at alle projektgrupper mindst tog et område op, som de ønskede at procesforbedre i løbet af projektet, som en mulig tilgang for procesforbedring i hele organisationen. At dette forslag kom fra Logimatic selv, synes vi er meget positivt, når den fremtidige anvendelsesmulighed af vores tilgang skal vurderes.

9.5 Opsumming på interventionen fase 2

Tabel 9-5 viser de vigtigste iagttagelser, vi gjorde i forbindelse med opfølgningen på andet review og forløbet som helhed. Disse iagttagelser forholder sig til ideen om selvforbedrende projektgrupper, og ikke så meget til selve reviewprocessen.

Den primære iagttagelse var, at selvforbedringsprocessen havde givet anledning til mærkbare forbedringer i projektgruppens reviewproces. Dette blev både påpeget af projektgruppen, og det var noget vi selv kunne observere i forbindelse med det andet review. Projektgruppen følte, at vores rolle, som er den rolle, SPI gruppen normalt skal påtage sig, var god, da den gjorde det muligt for projektgruppen at bevare procesejerskabet, mens den på samme tid ikke efterlod projektgruppen

helt for sig selv. Ligesom vi observerede efter den første opfølgning, blev flytningen af procesejerskabet og det tilhørende ansvar opfattet som en god ting både af projektgruppen og af SPI gruppen. SPI gruppen mente, at den øgede refleksion, som dette gav anledning til, var en klar fordel for hele organisationen.

Vigtige iagttagelser efter anden fase	
<i>Positive</i>	<ul style="list-style-type: none"> - Processen havde givet anledning til forbedringer. - SPI gruppen som vejledende enhed gav gode resultater. - Flytningen af ansvaret for procesforbedringen fra SPI gruppen til projektgrupperne blev opfattet som en god ting af både SPI gruppen og projektgruppen. - Refleksion er godt.
<i>Negative</i>	<ul style="list-style-type: none"> - Det kan være svært at vænne projektgruppen til selv at sidde med ansvaret hele tiden. - Projektgrupperne kan have svært ved at finde tid til at lave forbedringer. - Projektgruppen kan føle, at den ikke har kompetencerne til at lave forbedringer uden at have et ordentligt vidensgrundlag.

Tabel 9-5: Vigtigste iagttagelser i forbindelse med opfølgningen på anden fase.

På den negative side viste det sig, at det alligevel godt kan være problematisk at placere hele ansvaret i projektgruppen, da den godt kan føle, at der ikke altid er tid til at lave forbedringer, ligesom den kan have problemer med at vænne sig af med gamle vaner, der siger, at ansvaret for procesforbedringer ligger hos SPI gruppen. Dette kom til udtryk, da projektgruppen efterspurgte mere pres fra SPI gruppen. Endeligt kan projektgruppen føle, at den ikke besidder evnerne til at gennemføre brugbare forbedringer. Det blev nævnt, at det ville have været rart med mere viden indenfor området, så forbedringerne kunne være blevet bedre.

9.6 Opsamling på hele interventionen

I dette afsnit vil vi kort opsummere de vigtigste iagttagelser, vi gjorde i forbindelse med interventionen i Logimatic. I det efterfølgende kapitel vil disse iagttagelser blive analyseret yderligere med henblik på endeligt at besvare de spørgsmål, som blev stillet op i problemdefinitionen.

Disse iagttagelser er opsummeret i tabel 9-6. Overordnet set lykkedes det projektgruppen at gennemføre nogle forbedringer af reviewprocessen. Disse forbedringer skete på baggrund af problemer, gruppens medlemmer selv havde identificeret, og som også passede meget godt overens med de observationer, vi gjorde i forbindelse med det første review. Både SPI gruppen og projektgruppen selv syntes, at det var en fordel at flytte procesejerskabet fra SPI gruppen og ned i organisationen til projektgruppen, da det aflastede SPI gruppen, mens det på samme tid gjorde projektgruppen mere opmærksom på at forbedre processerne. Den nye forbedringstilgang, hvor SPI gruppen fungerer mere som en vejledende enhed fungerede godt, da det støttede godt op omkring projektgruppens forbedringsarbejde, uden at det gik ud over projektgruppens evne og lyst til at forbedre sig selv.

Vigtige iagttagelser ved interventionen	
<i>Positive</i>	<ul style="list-style-type: none"> - Gruppen kunne identificere problemer, opstille mål og opnå resultater. - Flytningen af procesejerskabet og -ansvaret blev godt modtaget både af projektgruppen og SPI gruppen. - Den dialektiske tilgang og SPI gruppen som vejledende enhed gav gode resultater. - Refleksion over egen proces kan give gode resultater hurtigt.
<i>Negative</i>	<ul style="list-style-type: none"> - Det kan være svært at vænne projektgruppen til selv at sidde med ansvaret hele tiden. - Der kan være problemer med at reflektere over egen proces. - Det kan være svært at frigøre SPI gruppen og projektgruppen fra gamle vaner og processer. - Projektgrupperne kan have svært ved at finde tid til at lave forbedringer. - Projektgruppen kan føle, at den ikke har kompetencerne til at lave forbedringer uden at have et ordentligt vidensgrundlag.

Table 9-6: Opsummering af de vigtigste iagttagelser i forbindelse med interventionen i Logimatic.

Det viste sig også, at gode resultater kan opnås forholdsvis hurtigt. Vores intervention i Logimatic løb over cirka en måned, og i løbet af den måned skete der væsentlige forbedringer i projektgruppens reviewproces, og projektgruppen havde allerede ved anden opfølgning forslag til nye problemområder. Der kan dog stilles spørgsmålstegn ved, hvorvidt det er lykkedes at få forankret forbedringerne i gruppen. Fx erklærede projektgruppen før interventionen, at det største problem i første omgang var overhovedet at få afholdt reviews. Derfor er det ikke sikkert, at forbedringerne vil blive anvendt i praksis, hvis der ikke ændres ved den grundlæggende antagelse om, hvorvidt der skal afholdes reviews eller ej. Interventionen har ikke direkte været inde og ændre ved denne antagelse, hvilket kan være en hæmmende faktor for forankringen af forbedringerne. Desuden vil der altid være en fare for, at projektgruppen falder tilbage på gamle vaner.

Omvendt var det heller ikke helt uproblematisk at flytte hele ansvaret for forbedringerne ned i projektgruppen. Det blev fx efterspurgt, at SPI gruppen har en mere aktiv rolle, når forbedringer skal sættes i gang. Vi erfarede også, at det kan være svært at få projektgruppen i gang med at reflektere over processerne, hvilket først kom rigtigt i gang, da vi havde hjulpet lidt til.

Et af de problemer, der blev påpeget gentagne gange er, at det er svært for projektgrupperne at afsætte tid til at lave forbedringer, da tiden skal tages fra det normale udviklingsarbejde, og der er tidsplanerne i forvejen presset til det yderste.

Endeligt kan det være et problem, at projektgruppen kan føle, at den ikke besidder de kompetencer, der skal til for at gennemføre fornuftige forbedringer. Dette skyldes højst sandsynligt en forventning om, at processer kan designes, og at det bedste resultat kommer gennem grundig analyse. Tilgangen med selvforbedrende projektgrupper bygger derimod på den tankegang, at processer

ikke kan designes, men at projektgrupperne i stedet skal gøre, hvad de føler er rigtigt. Det kræver ikke en komplet forståelse af det område, der skal forbedres, for eksempel viste det sig, at på trods af at projektgruppen ikke havde fået en teoretisk gennemgang af, hvordan et review bør afholdes, fik den alligevel identificeret en del problemer, som også påpeges som faldgruber i teorien.

I følgende kapitel vil vi på baggrund af de observationer, som vi har gjort os under interventionen i Logimatic beskrive reviewgruppen og foretage en vurdering af, hvordan de samarbejder i gruppen.

10.1 Reviewgruppen

Den gruppe, i det følgende betegnet reviewgruppen, som vi blev introduceret til under interventionen bestod af fire personer, som skulle foretage review over to omgange. På trods af, at de to reviews foregik på to forskellige projekter, bestod reviewgruppen af de samme personer, som var:

Reviewgruppen, som vi blev introduceret til under interventionen, bestod som før nævnt af fire personer, som skulle foretage review over to omgang. Ud over disse fire personer deltog der en person til under det sidste review. Det, som vi ønskede at observere, var, hvordan reviewgruppen udførte og koordinerede dens arbejde, og hvordan den generelt kom frem til en beslutning, når den skulle have løst en opgave. Derudover ønskede vi at observere, hvordan et review foregik i Logimatic.

Målet for det første review var at skabe enighed om, i hvilken retning ændringer og simplificering af et stålmodul i standard MARS skulle foregå. Andet review's mål var at skabe en fælles database for Kvaerner Masa-yards i forbindelse med opgradering til MARS 5.2. Begge opgaver er ikke simple, men forholdsvis komplekse, da ændringerne vil påvirke MARS produktet mange steder. Det var måske også baggrunden for, at det hovedsageligt var nøglemedarbejdere fra de enkelte grupper i MARS og fra Produktrådet, som deltog. Vi observerede at de argumenterede sagligt for deres sag, og hvis en anden mødedeltager kom med et teknisk eller saglig argument for ændringer, var deltageren villig til at ændre sit standpunkt. Yderligere så vi, at mange af de beslutninger der i løbet af reviewprocessen blevet taget skete i konsensus.

Vores observationer igennem begge reviews harmonerer meget godt med Constantines åbne paradigme [Con93]. Det åbne paradigme er baseret på tilpassende samarbejde, integreret innovation med stabilitet, individualitet og med fælles interesser igennem forhandling og diskussion. Her er roller og ansvar fleksibelt fordelt. Grupper baseret på det åbne paradigme deler informationer frit og evner at kombinere forskellige tilgangsvinkler hvilket gør, at gruppen er god til at løse komplekse opgaver. Faren ved det åbne paradigme er, at deltagerne i gruppen kan bruge en frygtelig masse energi på endeløse diskussioner uden at opnå enighed. En af de observationer, som vi gjorde, var, at gruppen en gang imellem rent faktisk brugte en masse tid på at diskutere små detaljer for til sidst at konkludere, at det alligevel ikke lå inden for de opstillede CPer.

10.2 Vurdering af reviewgruppen

Vi så, at reviewgruppen var gode til at løse den komplekse opgave, den var sat til. Den koordinerede indsatsen igennem en gensidig tilpasning af samarbejdet, dog så det ikke ud som om, at roller og ansvarsområder var helt klart defineret på forhånd. Vi kunne se at medlemmerne tilpassede sig til hinanden på en meget fleksibel måde. Reviewlederen sørgede for at lede mødet, mens producenten automatisk begyndte at tage notater til reviewdokumentet. Derudover observerede vi, at deltagerne med det samme indtog rollen som kritikere af reviewdokumentet. Vi så, at der ikke var nogen klar autoritetsstruktur i reviewgruppen, og at de igennem forhandlinger og diskussioner opnåede konsensus. Der var ikke nogen i reviewgruppen, der ensidigt dikterede en beslutning.

Vi lagde også mærke til, at da reviewgruppen traf beslutninger, foregik det igennem en åben dialog, hvor deltagerne lyttede opmærksomt til hinanden og fremførte deres argumenter uden at blive personlige i deres adfærd. Det foregik i en meget fordragelig atmosfære. Det, der var kendetegnet ved beslutningerne, var, at når gruppen nåede frem til den endelige løsning på problemet, var der ingen af deltagerne, som sad tilbage med en fornemmelse af, at de ikke selv havde haft en eller anden form for indflydelse på beslutningen. Der er selvfølgelig altid en risiko for, at deltagerne bliver enige uden reelt at være enige, men den faldgrube vil der altid være. Det vigtige er, at reviewgruppens medlemmer har indflydelse og dermed også føler et vist ejerskab over de løsninger, som gruppen kommer frem til.

En anden vigtig observation i denne sammenhæng er, at netop diskussionen omkring hvad den bedste løsning er, får reviewgruppen til at foretage en refleksiv proces og dermed double-loop læring. Diskussionen betyder også, at gruppen ikke blot fremkommer med en standard-løsning, men at der ud fra forskellige faktorerers indvirkning og vægtning findes frem til en unik løsning på problemet. Det er netop det, der ifølge Constantines åbne paradigme, er kendetegnet for gruppetyper, at den er god til at løse komplekse problemstillinger.

Det er vores vurdering, at reviewgruppen højst sandsynligt vil reagere negativt eller direkte tage afstand fra processer, som de skal bruge, hvis de ikke selv har haft indflydelse på, hvordan de er blevet til. Det kan forklares ud fra, at reviewgruppen, som vi observerede, er vant til at træffe beslutninger i fælleskab og igennem diskussion, dvs. igennem en gruppeproces. Det betyder, at hvis gruppens medlemmer ser processerne, som de ikke mener er teknisk holdbare eller ud fra deres synsvinkel ikke er gangbare, vil de alligevel ikke følge processen som foreskrevet. En observation vi igennem vores interview i efteråret også gjorde os, var at en af de udviklere, som vi havde interviewet, udtalte, at vedkommende fulgte processerne i Logimatics kvalitetssystem såfremt det var relevant eller gav mening, men hvis standarden ikke gav mening ud fra tekniske argumenter, mente vedkommende, at det var velbegrundet at fravige dem.

Det er derfor, efter vores opfattelse, vigtigt at reviewgruppen får indflydelse på, hvordan standarder og processer, som de i deres daglige arbejde skal følge, bliver udformet. Reviewgruppen er vant til at reflektere over deres arbejde, dvs. double-loop læring, og det harmonerer ikke med regler, der prøver at regulere deres adfærd, hvor de skal foretage en tilpasning i forhold til reglerne, dvs. single-loop læring. At prøve at formalisere deres adfærd, i alt for høj grad, uden at de selv har haft indflydelse på resultatet vil ikke være vellykket. De er vant til indflydelse på beslutningerne, og har de ikke det, mister de følelsen af procesejerskab og indflydelse, hvilket kan have uheldige konsekvenser. Det er derfor vores vurdering, at det er vigtig, at reviewgruppen igennem hele interventionen har haft indflydelse på den proces, som de gennemgik.

I følge Constantine er de måder, en åben gruppe kan påvirkes på, ved at opstille overordnede planer, strategier og mål, og derefter lade det være op til reviewgruppen selv at træffe beslutning om, hvordan problemet konkret skal løses. En anden måde reviewgruppen kan påvirkes på er igennem en forbedring af deres færdigheder, hvor de tilføres ny viden, som de kan bruge i deres problemløsning.

I tabel 10-1 har vi opsummeret, hvad der karakteriserer reviewgruppen, som vi har arbejdet sammen med i Logimatic igennem vores intervention.

Egenskaber	Karakteristika
<i>Prioriterer</i>	Stabilitet og forandring, gruppe og individ, tilpassende effektivitet
<i>Koordination</i>	Tilpassende samarbejdende proces
<i>Beslutningstagning</i>	Forhandlet, enighed, igennem gruppeproces
<i>Gruppens målfastsættelse</i>	Planlægning, strategier, fastsættelse af dagsorden, forbedring af færdigheder, afklaring af roller
<i>Gruppens form og stil</i>	Samarbejdende, udforskende, strategisk, fleksibel, læring og øvelser
<i>Gruppens styrker</i>	Praktisk tilpasning, informationsdeling, kompleks problemløsning
<i>Gruppens svagheder</i>	Kaotiske indviklinger, uendelige processer

Tabel 10-1: Gruppens karakteristika.

Procesforbedring gennem selvforbedrende grupper kan forklares ved hjælp af to ting, nemlig den forbedringsmetode der er skitseret på figur 7-2 side 68 samt de egenskaber, der karakteriserer selve grupperne. Gennem forbedringsmetoden understøttes og realiseres de egenskaber, som selvforbedrende projektgrupper har, og i dette kapitel vil vi undersøge, hvilken sammenhæng der er mellem forbedringsmetoden og de resultater, der blev opnået gennem interventionen, og hvorvidt der er specielle omstændigheder i Logimatic, der har gjort selvforbedrende projektgrupper til en mulighed. Det vil vi især prøve at forklare gennem de iagttagelser, vi gjorde os gennem interventionen, og som er opsummeret i tabel 9-6. Derudover prøver vi at identificere hvilke egenskaber ved selvforbedrende projektgrupper, der er nyttige i en konkret forbedringssituation, og hvilke egenskaber, der ikke er så hensigtsmæssige, og som der derfor bør arbejdes på at minimere effekten af.

11.1 Procesejerskab og problemerkendelse

Hos selvforbedrende projektgrupper ligger procesejerskabet i de enkelte grupper i stedet for i SPI gruppen. Det betyder, at projektgruppen føler et større ansvar for at forbedre processerne, og det gør det muligt for projektgrupperne at tage fat på områder, der er relevante for dem, fremfor områder der er relevante for SPI gruppen.

Flytningen af procesejerskabet sker i vores metode ved, at det er projektgruppen selv, der står for at identificere problemer og derefter udarbejde en løsning på problemerne. SPI gruppens rolle bliver i den forbindelse ændret fra at være den enhed, der udarbejder og udruller processer til at være en understøttende enhed, som kan vejlede og rådgive projektgrupperne i deres forbedringsarbejde. I Logimatic fungerede vi nærmest som SPI gruppen, da det var os, der hjalp projektgruppen med at identificere problemer, opstille mål og komme med løsningsforslag. En af de iagttagelser vi gjorde i forbindelse med interventionen var, at projektgruppen godt kunne identificere problemer ved deres proces, i det specifikke tilfælde i forbindelse med deres reviewproces, og at de også kunne komme med forslag til løsninger. I mange tilfælde viste det sig, at de problemer, som projektgruppen fik identificeret, passede meget godt overens med de problemer, vi selv observerede, og som også bliver nævnt som problemer i teorien om reviews, som er gennemgået i afsnit 7.2.

Både projektgruppen og SPI gruppen var godt tilfredse med den nye ansvarsfordeling, noget der også blev givet udtryk for under opfølgningen på forløbet. På en måde har det hele tiden været sådan i Logimatic, men det er først nu, at det er blevet klart for alle, at projektgruppen selv kan styre udviklingen af processerne uden de store problemer. Tidligere tilpassede projektgrupperne kvalitetssystemet efter den situation projektgruppen befandt sig i, men uden så meget at overveje hvorfor, og uden rent faktisk at etablere nye processer i gruppen. Nu er det i stedet blevet klart, at projektgruppen selv har ansvaret for processerne, og dermed kan gruppen med god samvittighed gå ud og sige, at de selv har lavet en brugbar proces.

11.2 Læring

Et ændret procesejerskab kan være med til at lægge grunden for en ændret læringsproces i projektgrupperne. Ved selv at sidde med procesejerskabet har gruppen mulighed for at ændre processen efter behov og vilje, hvilket gør det muligt at skifte fra single-loop læring til double-loop læring. Double-loop læring er et vigtigt kendetegn ved selvforbedrende projektgrupper og danner grundlaget for, at projektgruppen kan foretage den nødvendige refleksion over dens proces. Gennem double-loop læring reflekteres der over den nuværende proces, og problemer identificeres på baggrund af denne refleksion. Denne double-loop læring fører frem til en ændret adfærd i gruppen, hvilket resulterer i en forbedret udviklingsproces. Vi så et eksempel på en sådan læringsproces i Logimatic, hvor det var projektgruppen selv, der kom frem til problemerne ved reviewprocessen og på baggrund af det forbedrede deres reviewproces. Spørgsmålet er så, om resultatet ville have været det samme hvis vi eller SPI gruppen var kommet med det samme forslag til en forbedret proces, blot uden at projektgruppen havde været igennem den samme læringsproces.

Svaret på dette er ikke klart, men det vi kan sige er, at både en del af projektgruppen og en del af SPI gruppen tidligere har været på projektlederuddannelse, hvor de har fået indblik i de mere traditionelle SPI tilgange. På trods af det er SPI indsatsen i Logimatic gået lidt i stå, mens der blev udtrykt tilfredshed med vores metode. Det kan betyde, at den kan få større succes end fx en traditionel IDEAL tilgang. Selvfølgelig er det stadig et åbent spørgsmål, hvor meget vores tilstedeværelse har spillet ind. Det er klart, at en eller anden indflydelse har det haft, for det tvang projektgruppen til i det mindste at holde reviews, noget der har været et problem tidligere.

Vores forventning om, at double-loop læring er fordelagtigt i forhold til procesforbedring viste sig at blive opfyldt. Projektgruppen havde ikke nogen videre problemer med at ændre deres reviewproces, da de først havde fået identificeret de problemer, der var ved den, og refleksionen over gruppens egen proces ledte dermed til bedre resultater hurtigt. Helt smertefrit var det dog ikke, for double-loop læring i forbindelse med refleksion over egne arbejdsprocesser var nyt for projektgruppen, og det kan være svært at reflektere over egen proces for første gang. Med de rette værktøjer kan disse startvanskeligheder dog minimeres, for eksempel ser vi ideerne i SSM som brugbare i den sammenhæng. Det er dog ikke nok at ændre processerne, de ændrede processer skal også fastholdes. Vi har ikke været i Logimatic lang nok tid til at se, hvor god projektgruppen er til at fastholde processer, der er opstået på baggrund af double-loop læring, men det er nok ikke helt usandsynligt, at der vil komme de sædvanlige problemer med at fastholde dem. Især kan det være et problem, når der er tale om så små iterationer, som vi lægger op til, for her kan det være svært umiddelbart at se de store forbedringer, og fokus skifter hele tiden fra et problem til et andet.

Problemerne skyldtes til gengæld ikke modstand fra hverken projektgruppen eller selve organisationen. SPI gruppen hilste den noget anderledes tilgang velkommen, og lod projektgruppen få mere eller mindre frit spil. Det har været med til at fremme tilgangen, og det hjalp også med at lægge procesejerskabet ned til projektgruppen.

I relation til læring, så bemærkede projektgruppen under interventionen, at det ville have været rart med lidt mere viden omkring reviews inden forbedringerne gik i gang. Baggrunden var, at gruppen ikke følte, at den havde den rette kompetence til at sætte forbedringer i gang, og i nogen udstrækning kan det også være sandt. Vi hjalp trods alt ved at levere CMM mål og komme med forskellige forslag, men det var heller ikke på baggrund af teorien på området, det var nærmere ud

fra, hvad vi syntes var fornuftigt i det pågældende tilfælde. Dermed er noget af svaret på problemstillingen også givet, for det er klart, at jo mere viden gruppen har på det pågældende område, i dette tilfælde reviews, jo bedre forbedringer kan de lave. Faren er dog, at forbedringerne prøver at favne for vidt, og at de derfor mislykkes. Derfor er det efter vores opfattelse mindst lige så godt at lave forbedringerne ud fra, hvad gruppen selv synes er godt og skidt, og så eventuelt senere sammenligne med teorien på det område, forbedringerne koncentrerer sig om. Som allerede nævnt, viste det sig jo også, at de områder, der var blevet forbedret, stort set var de samme områder, som nævnes som vigtige i teorien omkring reviews, så frygten for at manglende kompetence vil lede til dårligere forbedringer er ikke helt gyldig.

11.3 Dokumentation og design

En konsekvens af det ændrede procesejerskab og den ændrede læringsproces er også, at produktionen af ledelsesrelaterede produkter som dokumentation og design ændres fra at være nødvendigt til at være noget, der laves, når det gavner projektgruppen. Som nævnt i afsnit 4.1, så bygger vores tilgang på, at formålet med processerne er at tilfredsstille udviklerne og opfylde deres behov. Det betyder, at dokumentationsaspektet er nedtonet kraftigt, og at processerne har en mere praktisk tilgang fremfor at være baseret på krav fra andre dele af organisationen. Det kan ses som et resultat af det ændrede procesejerskab, for når procesejerskabet ligger i projektgruppen, er der ingen grund til at projektgruppen skal gøre andet, end hvad den føler er nødvendigt. Det kan, som tidligere nævnt, være i konflikt med organisationens overordnede målsætninger, men her vægter vores tilgang projektgruppens behov over andres behov. Faren ved dette har også vist sig at være mindre end først antaget. Gennem interventionen viste det sig, at de problemer, som projektgruppen identificerede ved reviewprocessen, lå meget godt op af de problemer, som kunne identificeres ved hjælp af teorien på området. Det betyder, at forskellen mellem de overordnede mål og de mål, som projektgruppen sætter sig, ikke nødvendigvis er særlig stor. Den eneste forskel er, at det kun er de relevante problemstillinger, der bliver behandlet i projektgruppen, hvor organisationens overordnede mål nødvendigvis må dække meget mere. For eksempel har Logimatic CMM niveau 3 som overordnet mål, men nogle af de indsatsområder, der er i CMM, er ikke relevante for projektgruppen i det projekt, der er i gang i øjeblikket. Der er selvfølgelig ingen garanti for, at de selvforbedrende projektgrupper altid vil ligge helt på linje med organisationens overordnede mål, men i det tilfælde er det SPI gruppens ansvar at sørge for, at der ikke sker alt for store afvigelser, fx ved at hjælpe projektgruppen med at opstille nogle relevante mål.

Ved at lægge vægt på, at processerne skal være henvendt til udviklerne og skal have fokus på praksis, sikres engagementet og motivationen blandt gruppens medlemmer også. Det kan være svært at bruge designede processer i praksis, også selv om de umiddelbart virker meget fornuftige. Helt let er det dog ikke, da det kan være svært for gruppen at slippe de tidligere processer. Vi så det i Logimatic ved, at kvalitetssystemet ofte blev nævnt som en faktor, der skulle tages hensyn til på trods af, at de standarder, der ligger i systemet, ofte er enten mangelfulde eller irrelevante.

Logimatic er, som mange andre softwarevirksomheder formentligt også er, en organisation, der i forvejen har en forholdsvis flad struktur, og medarbejderne sidder med en del af ansvaret. Dermed er grundlaget for en forbedringstilgang, der bygger på medarbejdernes behov allerede lagt, da der er tradition for at lytte til medarbejderne og lade dem komme med forslag. I en organisation,

der ikke er vant til at uddelegere ansvaret, vil det være langt sværere at indføre selvforbedrende projektgrupper, da den nødvendige tillid mellem projektgruppen og ledelsen ikke er til stede, og derfor først skal opbygges.

11.4 Planlægning

Modsat traditionelle SPI tilgange bygger vores tilgang på, at den verden vi lever i er under konstant udvikling, og at de mennesker, der lever i den, dermed også er under konstant udvikling. Derfor er det meget svært, hvis ikke helt umuligt, at planlægge alt i detaljer. Det er selvfølgelig nødvendigt med en overordnet plan, men i stedet for at have form som en stram tidsplan kan det være mere hensigtsmæssigt fx at udforme planen som et mål og en række prioriterede delmål, som til sammen leder frem til det overordnede mål. Tilgangen med selvforbedrende projektgrupper prøver at tilgodese disse forhold gennem to elementer. For det første foregår der et minimum af planlægning, for det andet sker forbedringerne i små iterationer, hvor hver iteration tager fat på et enkelt lille problem. Det medfører, at projektgruppen ikke prøver at følge urealistiske planer, og at forbedringsaktiviteterne hele tiden kan tilpasses omstændighederne. Dvs. at skulle der opstå uventede komplikationer, eller skulle projektet tage en uventet drejning, så er processerne ikke så faste, at de ikke kan ændres til at fungere under de nye forhold.

Det projekt vi fulgte i Logimatic tog ikke nogen uventet drejning, men forbedringerne blev alligevel ikke planlagt i detaljer. I stedet koncentrerede projektgruppen sin indsats omkring det, der var brug for på det pågældende tidspunkt, altså reviews. Der blev ikke planlagt længere frem i tiden, for ingen vidste præcist hvad der ville ske og hvilke opgaver, der præcist ventede forude. Ved at tage fat på aktuelle problemstillinger kan engagementet igen sikres, og projektgruppen risikerer ikke at lave procesforbedringer på områder, de måske aldrig kommer til at arbejde med.

SPI gruppen støttede op omkring denne tilgang og udtalte, at det virkede meget fornuftigt at tage fat på aktuelle problemstillinger i stedet for at prøve at se ud i fremtiden. Der kan dog stadig være en fare for, at det længere oppe i organisationen ses som et problem at der ikke er langsigtede planer for forbedringerne. Det hænger igen sammen med hvor meget tiltro, der er til medarbejderne, og hvor meget ansvar organisationen er villig til at skubbe ned til projektgrupperne. Umiddelbart virkede det ikke som et problem i Logimatic, ikke mindst fordi der rent faktisk blev fremvist positive forbedringer, og det dermed blev sandsynliggjort, at selvforbedrende projektgrupper kan være fordelagtige.

11.5 Normer

Procesforbedring gennem selvforbedrende projektgrupper har stort set kun fokus på, at udviklerne skal have et godt miljø at arbejde i, og ikke så meget på at opfylde organisationens forretningsmål. Normer som CMM og ISO er lavet bl.a. for at kunne sammenligne både projektgrupper internt i organisationer og på tværs af organisationer. Normerne gør det muligt at måle ydelsen, og på baggrund af dette målrette og prioritere forbedringsindsatsen [AAMN00].

Som vi også vil komme ind på senere, er det med selvforbedrende projektgrupper SPI gruppens opgave og ansvar at styre procesforbedringerne mod det overordnede mål. Dette mål kan være sat

i forhold til en norm eller i forhold til et mål opsat selvstændigt af organisationen. Hvis det er et mål, der er opsat på baggrund af en norm, er spørgsmålet, om det med selvforbedrende projektgrupper overhovedet er muligt at opfylde normen. Problemet er, at projektgrupperne kan komme til at arbejde i vidt forskellige retninger, og mange projektgrupper vil formentlig ikke komme til at arbejde med de samme problemer, hvilket betyder, at processerne på tværs af organisationen muligvis aldrig kan blive ensrettet, som det jo er formålet med en normbaseret tilgang. Hvorvidt det er muligt at bruge selvforbedrende projektgrupper og samtidig følge en norm er et uafklaret spørgsmål, og er et område, der stadig mangler at blive udforsket grundigt.

Vi er enige i, at normer kan være både fornuftige og nødvendige, hvis en organisation ønsker at ensarte udviklingsprocesserne over hele organisationen. På den anden side er vi af den opfattelse, at normerne ofte overvurderes og ophøjes til mere end de rent faktisk er, hvilket kan resultere i, at normen kommer i fokus fremfor de udviklere, der skal følge normen. På den baggrund er selvforbedrende projektgrupper ikke en total afvisning af normbaseret procesforbedring, der er blot et uafklaret spørgsmål om, hvordan forholdet mellem de to skal være.

11.6 Processen

Den nedtonede planlægning bidrager også til, at processerne får en anden karakter. Uden planlægning er det svært at prøve at designe processer, hvilket vi ser som en god effekt af den nedtonede planlægning. Vi har set det modsatte forsøgt med traditionel SPI, og vi har set det konkret i Logimatic med deres kvalitetssystem. Processerne designes og forsøges derefter udrullet, hvorefter der ikke gøres meget ved dem på trods af, at de både kan være mangelfulde pga. forældelse eller ikke dækker alle områder. Processerne for at få ændret i kvalitetssystemet er beskrevet, og der er også udpeget ansvarlige, men alligevel sker det kun sjældent, at systemet rent faktisk ændres. Grunden til dette kan være, at der ikke lægges vægt på, at det er vigtigt at holde systemet opdateret, og at ændringsprocessen er for bureaukratisk.

Ved i stedet at lægge vægt på, at processer skal være praktiske, og at forbedringer ikke kan planlægges i detaljer, sikres det også, at processerne aldrig ligger stille. Den samlede udviklingsproces er hele tiden under udvikling, og fastfryses stort set aldrig for at lave ændringer. I stedet sker der hele tiden små ændringer rundt omkring, og det gør processen brugbar for udviklerne. I det øjeblik noget ikke længere passer, tager projektgruppen fat på det pågældende område og forbedrer processen. Det så vi tydeligt gennem interventionen i Logimatic, hvor den standard for reviews, der lå i kvalitetssystemet, ikke længere var brugbar, og projektgruppen i stedet besluttede selv at forbedre den del af processen. Dette er igen i konflikt med et ønske om at forbedre processer i forhold til normer, da normerne ofte siger, at processerne skal være ensartede på tværs af organisationen. Desværre er det temmelig svært at bruge den samme udviklingsproces overalt, når processen aldrig fastfryses, men i stedet er under konstant udvikling.

Tilgangen, som selvforbedrende projektgrupper bygger på, lægger også op til denne løbende forbedring af udviklingsprocessen. Dog er der ikke tale om en fælles udviklingsproces for hele organisationen, men en udviklingsproces for de enkelte projektgrupper. Det har allerede vist sig at være meget svært at lave en fælles proces for hele organisationen, i Logimatic er der fx to holdsvis forskellige forretningsområder, der beskæftiger sig med forskellige typer af udvikling,

og at lave en fælles udviklingsproces for de to områder er nærmest umuligt, noget Logimatics kvalitetssystem også lider under.

At processen hele tiden er under udvikling gør, at dokumentation af processen bliver svært. Hvis projektgruppen sætter sig ned for at dokumentere processen, er der stor sandsynlighed for, at denne dokumentation vil være forældet inden ret længe. Et alternativ til dokumentationen er best practices, som beskriver hvad projektgruppen gjorde i et konkret tilfælde med hensyn til en enkelt forbedring. Det kan bruges som inspiration i andre projektgrupper, og kan bruges af SPI gruppen til at hjælpe andre projektgrupper på vej. Der er dog stadig et stykke vej til egentlige procesbeskrivelser, men som allerede nævnt, lægger vores tilgang op til minimal dokumentation, da det både fjerner fokus fra den egentlige udvikling, og da det også kan være med til at sætte udviklingen i stå, og dokumentationen vil i øvrigt sjældent give et retvisende billede af, hvad det er, der foregår i projektgruppen og i organisationen som helhed.

11.7 Fokus

Vi har tidligere kritiseret traditionel SPI for at have fokus på processerne fremfor på de udviklere, der skal bruge processerne, og selvforbedrende projektgrupper er vores forslag til, hvordan dette syn kan ændres, så det bliver udviklerne, der er i fokus. Med selvforbedrende projektgrupper ændres fokus gennem de egenskaber, der er gennemgået ovenfor. De bidrager alle til at sætte projektgruppens behov i fokus fremfor andres behov, og dette er med til at gøre tilgangen mere menneskelig. Formålet med dette er at sikre projektgruppens engagement i procesforbedringerne, og at sikre at der ikke bliver udviklet unødvendige processer. Ved at flytte fokus bliver flere medarbejdere i organisationen dermed involveret i forbedringsarbejdet, og der er på den baggrund større chance for, at forbedringer rent faktisk vil blive gennemført og fastholdt.

I Logimatic viste det sig at være godt på flere måder. Først og fremmest var der ingen af deltagerne, der udtalte sig direkte negativt om tilgangen, i stedet udtalte SPI gruppen, at den godt kunne se perspektiverne i en sådan tilgang netop på grund af, at den tager udgangspunkt i projektgruppens behov. På den baggrund kan det antages, at det med selvforbedrende projektgrupper også i praksis kan lykkes at ændre fokus, også selv om ændringen ikke sker i hele organisationen på én gang.

Igen afhænger denne succes af hvilke traditioner, der eksisterer i organisationen i forvejen, og hvilken type organisation, der er tale om. Logimatic er tidligere blevet karakteriseret som et professionelt bureaukrati, og projektgrupperne som åbne projektgrupper, der er baseret på konsensus og gensidig tilpasning. Dette er en karakteristik, der formentligt passer forholdsvis godt på de fleste softwarevirksomheder, og derfor forventer vi, at selvforbedrende projektgrupper også vil kunne passe ind i andre virksomheder end Logimatic. Det professionelle bureaukrati er netop bygget på, at en del af ansvaret er skubbet ned i organisationen, og åbne grupper bygger på denne antagelse, da de åbne grupper typisk selv sidder med meget ansvar. Uden denne ansvarsfordeling kan selvforbedrende projektgrupper ikke blive en succes, da det så heller ikke vil være muligt at placere procesejerskabet i projektgruppen.

Der er nogle situationer, hvor det ikke er muligt eller fordelagtigt at ændre fokus så kraftigt, som selvforbedrende projektgrupper lægger op til. Som allerede nævnt, så er der i det professionelle bureaukrati en tradition for at lægge ansvar ned i organisationen, noget der er nødvendigt hvis

selvforbedrende projektgrupper stiller som et krav. En organisationstype, der ikke tillader denne ansvarsfordeling har derfor ikke mulighed for umiddelbart at drage nytte af selvforbedrende projektgrupper. Et andet tilfælde, hvor fokusskiftet kan være u hensigtsmæssigt er hvis der virkelig er brug for en ensartet proces i organisationen. Det kan fx dreje sig om udviklingsprojekter til forsvarret, hvor der er strenge krav til produktets kvalitet, og her kan det være hæmmende for udviklingen, hvis processen hele tiden ændrer sig.

11.8 SPI gruppens rolle

De egenskaber, som selvforbedrende projektgrupper har, lægger også op til en ændret SPI gruppe. Under interventionen i Logimatic påtog vi os rollen som SPI gruppe. Denne rolle var væsentlig anderledes end den, SPI gruppen normalt havde i Logimatic. Den normale rolle er at opstille mål for organisationen, igangsætte forbedringsinitiativer rundt omkring, samle op på disse initiativer og eventuelt udbrede erfaringerne til resten af organisationen. Det betyder, at SPI gruppen er vant til at have procesejerskabet, og det er et af de væsentlige elementer, der er ændret med selvforbedrende projektgrupper. Det vil sige, at vores rolle ikke var at påpege problemer, men at vejlede og rådgive projektgruppen når den skulle identificere problemer. Dette samspil mellem projektgruppen og SPI gruppen er skitseret på figur 4-3 på side 47, som viser forholdet mellem de to grupper. For at sikre procesejerskabets placering og at fokus ikke rykkes over på design af processer, er det ikke tilladt for SPI gruppen at stille krav til projektgruppens processers udformning eller indhold. Det eneste SPI gruppen må gøre er at styre organisationens overordnede mål.

Vi har tidligere nævnt, at SPI gruppen udtrykte sig positivt omkring den ændrede ansvarsfordeling. Vi er af den opfattelse, at SPI gruppen i de fleste softwareorganisationer ligner hinanden, og har nogenlunde samme opgaver. Der er sandsynligvis forholdsvis stor forskel på, hvor mange ressourcer, der er afsat til SPI grupperne, men ofte vil situationen formentligt se ud som i Logimatic, hvor SPI gruppen ikke har tilstrækkelig tid til at igangsætte de nødvendige forbedringer. Derfor kan Logimatics SPI gruppes positive holdning begrundes med, at gruppen ved hjælp af selvforbedrende projektgrupper kan spare tid, og dermed koncentrere sig om de mere overordnede strategier.

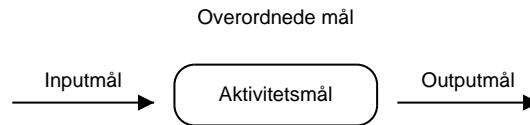
11.8.1 Målstyring

Spørgsmålet er hvordan SPI gruppen lægger langsigtede strategier for organisationens SPI indsats, og her er målstyring vores bud på et svar. Det eneste SPI gruppen kan gøre overfor projektgrupperne er at opstille mål. Hvorvidt disse mål skal være udformet i henhold til eksisterende standarder, fx CMM eller ISO, vil vise sig gennem praktiske erfaringer, men det ændrer ikke på, at opstilling af mål er den eneste måde SPI gruppen kan få indflydelse på.

Et meget godt spørgsmål i forhold til opstilling af mål er, hvordan målene er formuleret, og hvad de er formuleret på baggrund af. Hvordan de er formuleret har indflydelse på, hvordan de opfattes og hvor meget frihed, der overlades til projektgruppen selv.

Mål kan opdeles i forskellige kategorier, alt efter hvor specifikke de er, og hvilke områder de henvender sig mod. På figur 11-1, som er delvist inspireret af Humphrey [Hum90], er vores forslag til en opdeling vist. Her skelner vi mellem overordnede mål, inputmål, aktivitetsmål og outputmål. Overordnede mål er de mål, der overordnet gælder for aktiviteten. I vores tilfælde med Logimatic

og reviewaktiviteter kan det være det overordnede mål fra CMM, nemlig at defekter i udviklingsprodukter elimineres. Inputmål er mål, der stilles for at aktiviteten kan starte. Det kan fx være at der skal være afsat tilstrækkelig tid til aktiviteten, og at aktiviteten er planlagt. Aktivitetsmål er mål for den specifikke aktivitet. For et designreview i Logimatic kan et mål være, at manglende krav i designdokumentet skal identificeres, og at den interne standard for designdokumenter skal være opfyldt. Endeligt er outputmål de mål, der skal opfyldes for at aktiviteten kan afsluttes. Det kan være mål som, at der skal være et referat fra reviewet, der opsummerer, hvad der blev gennemgået og besluttet på reviewet.



Figur 11-1: De fire kategorier af mål i forhold til en aktivitet.

Tabel 11-1 opsummerer CMM målene inden for hver kategori.

Kategori	Mål
<i>Overordnede</i>	Defekter i udviklingsprodukter identificeres og fjernes.
<i>Input</i>	Reviewaktiviteter er planlagt.
<i>Aktivitet</i>	Identificer manglende krav. Afgør om kravene er implementerbare, klart formuleret, konsistente med hinanden og verificerbare.
<i>Output</i>	Resultater er dokumenterede.

Tabel 11-1: Mål for reviews.

Eftersom der er tale om selvforbedrende grupper, er det vigtigt, at SPI gruppen ikke opstiller målene som om, at de er absolutte krav, men at de i stedet stilles op som forslag. I mange tilfælde gælder det om at argumentere for, hvorfor målene er relevante for den konkrete projektgruppe, hvorefter projektgruppen ofte vil kunne se, at det måske er en god idé at forfølge målene. Det er dog stadig vigtigt ikke helt at se bort fra projektgruppens egne forslag til mål, da de vil være baseret direkte på de problemstillinger, projektgruppen oplever i hverdagen. På den baggrund vil de mål, der kommer fra SPI gruppen ofte være mere langsigtede, der bygger på CMM, ISO eller lignende standarder, hvor målene fra projektgruppen er mere kortsigtede og sigter mod at løse konkrete problemstillinger. Det gør også, at de to grupper ikke nødvendigvis er i konflikt, men at mål af begge typer kan opstilles uden videre problemer.

11.8.2 Ressourcekrav

Som det blev nævnt af projektgruppen under opfølgningen på interventionen, er en minimal SPI gruppe ikke helt uproblematisk. Problemet er ikke, at projektgruppen ikke er interesseret i at lave forbedringerne selv, problemet er nærmere, at der, som situationen ser ud i Logimatic i øjeblikket,

ikke er afsat den nødvendige tid til at foretage forbedringerne. Det er et typisk problem, som også eksisterer i andre softwarevirksomheder, hvor interessen er til stede, men det bliver straks værre når der skal forbedres i praksis, for så koster det tid, ligegyldigt hvilken tilgang, der benyttes. Med selvforbedrende projektgrupper er det efter vores opfattelse ikke så meget tid, der skal bruges af de enkelte projektgrupper. De forbedringer, der blev gennemført i Logimatic, blev gennemført på baggrund af omkring 3 timers ekstraarbejde, hvilket var den tid, det tog at identificere problemer, opstille mål og reflektere. Det inkluderede også en introduktion til selvforbedrende projektgrupper, noget som kun skal gøres en gang for hver projektgruppe, en opfølgning på første review, hvor problemerne blev identificeret og en opfølgning på det andet review, hvor forbedringerne blev evalueret. Selve reviewene er ikke talt med, da det ikke var ekstraomkostninger. Det vil sige, at selve forberedelserne til forbedringerne til en iteration blev gennemført på 2 timer, hvilket efter vores mening er meget fornuftigt, da resultaterne var umiddelbart mærkbare, og projektgruppen var tilfreds med resultatet, og havde straks forslag til andre områder, der kunne forbedres. Det betyder dog ikke nødvendigvis, at en projektgruppe kan nå at gennemføre mange iterationer i løbet af kort tid, da de udviklingsaktiviteter, som skal forbedres, også skal gennemføres for at komme igennem en forbedringsiteration.

Selv om der reelt ikke skal bruges megen ekstra tid, kan det umiddelbart godt virke som en stor og uoverkommelig opgave at skulle forbedre processer selv. Det hænger i Logimatics tilfælde sammen med, at tilgangen er ny, og det derfor kan være svært at vurdere, hvor mange kræfter den vil tage, men vi mener stadig, at det er mere givtigt end at have en SPI gruppe til at gøre det. Derfor burde det også være muligt at afsætte den minimale tid, der skal bruges på det, da der er en umiddelbar gevinst ved at lave forbedringerne. Det betyder også, at SPI gruppen får mere tid til at koordinere den overordnede indsats og udbrede erfaringer fra de enkelte projektgruppers forbedringsinitiativer, og mange af de traditionelle problemer med SPI grupper forsvinder helt automatisk. For eksempel behøver SPI gruppen ikke længere så mange ressourcer, og det er ikke nødvendigt at have de dygtigste medarbejdere siddende der, da forbedringerne sker i de enkelte projektgrupper, hvor de dygtige medarbejdere jo normalt arbejder.

11.8.3 Erfaringsopsamling

Når en projektgruppe har gennemført en forbedring, er det vigtigt, at den erfaring, projektgruppen har gjort sig, også bliver udbredt til resten af organisationen. Med selvforbedrende projektgrupper er det SPI gruppens opgave at hjælpe med at evaluere forbedringen og uddrage nyttige erfaringer. Først og fremmest drejer det sig om, at SPI gruppen skal bruge de erfaringer, der er gjort, når de skal hjælpe andre grupper med at forbedre sig selv. Hvis en gruppe står i en situation, en anden gruppe allerede har forbedret sig i, vil det være meget nærliggende at inddrage de erfaringer, der tidligere er gjort. Igen skal det kun være i form af råd og vejledning, for selv om en gruppe havde succes med en metode, er det ikke sikkert, at det samme vil gøre sig gældende i en anden gruppe, da andre faktorer kan spille ind.

For at gøre arbejdsbyrden mindre for SPI gruppen kunne en mulig fremgangsmåde være, at projektgruppen selv udarbejder et dokument, der beskriver hvad de gjorde og hvilke faktorer, der spillede ind, og selvfølgelig hvad resultatet blev. Dette dokument kan så indgå i en samling af forbedringserfaringer, hvor de mest succesfulde vælges ud og gøres til best practices, som så bliver tilgængelige for resten af organisationen, og som kan bruges til inspiration. Dette kan også være

med til at eksternalisere noget af den viden, der automatisk kommer til at ligge i SPI gruppen, så det bliver muligt at skifte medlemmer ud, uden at miste al for meget viden. Selvfølgelig vil det aldrig være det samme at have erfaringer nedskrevet i dokumenter, som det vil være at have de personer, der var til stede til rådighed, men det er trods alt bedre end ingenting.

En anden mulighed med hensyn til erfaringsopsamling kunne være, at en projektgruppe, der med succes har indført forbedringer, kommer og fortæller andre projektgrupper om, hvad der skete, og hvorfor projektet forløb som det gjorde. En ting, der dog er vigtig at bemærke i forbindelse med erfaringsopsamling er, at forbedringserfaringerne ikke må tolkes som andet end erfaringer. Som allerede nævnt, er der mange faktorer, der spiller ind, og hvad der er en umiddelbar succes i en gruppe kan blive en fiasko i en anden. Derfor skal erfaringer tages for hvad de er, og ikke uden omtanke ophøjes til standarder, der skal gælde for hele organisationen. Først hvis det viser sig, at en forbedring har succes i flere dele af organisationen, kan SPI gruppen begynde at overveje den slags, men i første omgang er erfaringerne kun brugbare som inspiration, og til at gøre rådgivningen bedre.

SPI gruppens opgaver
<ul style="list-style-type: none"> - Vejledere og rådgivere for projektgrupperne. - Hjælpe med problemidentifikation. - Målstyring af forbedringsinitiativer. - Erfaringsopsamling og udbredning.

Tabel 11-2: SPI gruppens opgaver i forhold til selvforbedrende projektgrupper.

Tabel 11-2 opsummerer SPI gruppens vigtigste opgaver i forhold til selvforbedrende projektgrupper.

11.9 Faldgruber for selvforbedrende projektgrupper

Selvforbedrende projektgrupper har mange gode egenskaber, og under de rette omstændigheder kan det lede til forbedrede processer. Interventionen i Logimatic viste, at det godt kan lade sig gøre at lave bottom-up procesforbedring på kort tid, men den gjorde det også klart, at der er nogle faldgruber, som der skal være opmærksomhed omkring, når selvforbedrende projektgrupper anvendes. Tabel 11-3 opsummerer de væsentligste faldgruber. Alle punkterne i tabellen kan undgås, det gælder bare om at være opmærksom på, at det er noget, der kan gå galt.

Et problem ved en metode, der udfører små iterationer, som resulterer i små ændringer hele tiden, kan føre til, at forbedringerne overses eller glemmes. Det kan betyde, at forbedringsindsatsen opfattes som tidsspild, der alligevel ikke fører til noget brugbart. Dette vil dog oftest ske, hvis forbedringerne ikke er direkte relateret til projektgruppens nuværende problemer, for som vi så i Logimatic, var forbedringerne i reviewprocessen forholdsvis små, men de var alligevel synlige og gav anledning til en forbedret proces. Som tidligere nævnt kan der stadig være problemer med at fastholde de forbedrede processer næste gang den samme udviklingsaktivitet skal gennemføres. Hvordan disse problemer kan overvindes, og hvorvidt det i det hele taget er noget, der hæmmer selvforbedrende projektgrupper i særlig høj grad, er et åbent spørgsmål. Vores forventning er dog, at eftersom forbedringerne sker på relevante områder og er udarbejdet af gruppen selv, er der en

større chance for, at de vil blive fastholdt, end der ville være, hvis forbedringerne var udarbejdet af andre.

En af de største farer er, hvis projektgrupperne ikke har nok tid til at forbedre. Der er ikke nogen let løsning på det problem, men som allerede påpeget, var det samlede tidsforbrug i Logimatic på 2 timer, og der er en god sandsynlighed for, at de 2 timer kan spares i det efterfølgende arbejde som resultat af den forbedrede reviewproces. Derfor må det være et spørgsmål om, at organisationen indser, at det er en fordel at ofre lidt tid på løbende forbedringer, da det vil øge kvaliteten af slutprodukterne. Denne fare udspringer primært af en kortsigtet udviklingsstrategi, der først og fremmest fokuserer på at udvikle nye funktioner i stedet for også at forebygge fejl. Dette er dog ikke noget, der udspringer direkte af selvforbedrende projektgrupper, for stort set alle tilgange til procesforbedring, har dette problem.

Vi har også været inde på, at det kan ses som et problem, hvis der ikke er nogen samlet udviklingsproces for hele organisationen. Det er oftest i forhold til CMM eller ISO, at dette problem trækkes frem. Her kan problemet være, at de to normer begge lægger en del vægt på, at processerne skal være dokumenterede, og det har vi jo argumenteret imod. Med selvforbedrende projektgrupper ligger hovedvægten på at praktisere processerne og ikke at dokumentere dem. I stedet udarbejdes best practices, som dokumenterer, hvad en projektgruppe har gjort i en situation. Der er dog stadig uoverensstemmelse mellem hvad, der er beskrevet, og hvad der rent faktisk foregår i organisationen, så hvorvidt selvforbedrende projektgrupper kan eksistere i en organisation, der også sigter mod CMM eller ISO er endnu et åbent spørgsmål. Umiddelbart mener vi, at organisationen må vægte de to tilgange overfor hinanden og foretage et valg, alt efter hvad der er vigtigst for organisationen.

Endeligt er der en fare for, at selvforbedrende projektgrupper ikke bakkes tilstrækkeligt op i organisationen. Det kan ske som følge af, at standardisering vægtes højere, eller af at resten af organisationen ikke er villig til at ændre synet på procesforbedring. Selvforbedrende projektgrupper kræver, at synet på processer ændres, så de ikke opfattes som noget, der kan designes, men som noget der løbende udvikles af de personer, der skal bruge dem. Bliver ledelsen og SPI gruppen ved med at holde fast i et traditionelt syn på processer, bliver det meget svært for projektgrupperne at opnå den nødvendige opbakning, og forbedringerne vil højst sandsynligt aldrig blive til noget.

Faldgruber
<ul style="list-style-type: none">- Små forbedringer kan blive overset og glemt.- Forbedringerne bliver ikke fastholdt.- Manglende tid til forbedringer.- Ingen samlet udviklingsproces for hele organisationen.- Projektgrupperne arbejder i forskellige retninger uden styring.- SPI gruppens eller ledelsens syn på procesforbedring ændres ikke.

Tabel 11-3: Faldgruber ved selvforbedrende projektgrupper.

Vi har gennem dette projekt introduceret selvforbedrende projektgrupper og afprøvet ideen i praksis. Det viste sig, at det er muligt at lave metodisk bottom-up procesforbedring, der bygger på, at procesejerskabet er lagt ned til projektgrupperne. De forbedringer, der blev foretaget, var små forbedringer, der alligevel havde en god effekt på projektgruppens arbejde, og var dermed medvirkende til, at projektgruppen fik en bedre reviewproces. Der er dog stadig et åbent spørgsmål om, hvorvidt forbedringerne vil blive fastholdt, noget vi ikke har beskæftiget os med i dette projekt.

I det følgende vil vi besvare de delspørgsmål, der blev opstillet i problemformuleringen.

12.1 Hvilke egenskaber karakteriserer selvforbedrende projektgrupper?

De egenskaber, der kendetegner selvforbedrende projektgrupper har vi identificeret ved at vurdere traditionel SPI og de problemer, der er ved denne tilgang, og efterfølgende ved at vurdere forskellige former for alternativ SPI. Resultatet af dette blev de otte egenskaber, som er vist i tabel 12-1. Disse egenskaber bygger i høj grad på ideen om bottom-up procesforbedring og End-user SPI, og er funderet i samling af udvikling og anvendelse af processer og fokusering på etablering af praksis.

Da udvikling og anvendelse af processer samles ved indførelse af selvforbedrende projektgrupper sidder projektgrupperne med ansvaret for problemerkendelse og procesejerskabet. Herved er det gruppernes ansvar at identificere problemer i deres udviklingsproces og hvorfor problemerne opstår, komme med løsninger på problemerne og efterfølgende anvende løsningerne i praksis. Dette ansvar betyder, at de selvforbedrende projektgrupper er i stand til at foretage double-loop læring. Derfor er der fokus på, at det væsentligste ved procesforbedring ikke er processerne, men udviklerne og ændring af deres adfærd, hvilket sker mest optimalt ved, at de selv har ansvaret for procesforbedringen.

At udviklerne er i fokus ved anvendelse selvforbedrende projektgrupper som forbedringstilgang giver sig også udslag i, at der fokuseres på etablering af forbedret praksis hos udviklerne. Dvs. at selvforbedrende projektgrupper sigter mod at ændre deres adfærd, før de modellerer en forbedret proces. Den modellerede proces kommer herved til at afspejle den faktisk anvendte praksis i projektgruppen.

Da procesforbedringen gennem selvforbedrende projektgrupper tager udgangspunkt i de problemer, som de enkelte projektgrupper konkret sidder med, er det ikke muligt at foretage en detaljeret planlægning af SPI forløbet i organisationen. Dette skyldes dels, at det ikke er muligt at forudsige, hvilke problemer det vil opstå i forbindelse med projekterne. Desuden har projekterne højst

sandsynligt nogle forskellige problemer eller de anser ikke de samme problemer som de væsentligste. Derved er det helle ikke muligt, at fastfryse processen og konstruere et billede af, hvordan udviklingen foregår på et givent tidspunkt, da processen ikke vil være gældende for alle projektgrupperne i organisationen. Derimod skal processen betragtes som værende dynamisk. Den er hele tiden under udvikling i organisationens projektgrupper og derved rekonstrueres den løbende, så den hele tiden afspejler den praksis, som anvendes i øjeblikket.

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen erkender problemerne ved deres udviklingsproces.
<i>Procesejerskab</i>	Ansvar for processen ligger hos projektgruppen.
<i>Planlægning</i>	SPI forløbet improviseres.
<i>Design</i>	Primært fokus på etablering af praksis, derefter modelleres resultatet.
<i>Dokumentation</i>	Procesbeskrivelser skal rekonstrueres løbende.
<i>Normer</i>	-
<i>Fokus</i>	Fokus på udviklerne.
<i>Læring</i>	Double-loop læring hos projektgruppen.

Tabel 12-1: Selvforbedrende projektgruppers egenskaber og kendetegn.

12.2 På hvilken måde kan en softwareorganisation fremme selvforbedrende projektgrupper?

Efter at have identificeret de vigtigste egenskaber ved selvforbedrende projektgrupper opstillede vi en metode for, hvordan de kan indføres i en organisation i praksis, og denne metode har vi afprøvet i Logimatic. Selve metoden er vist på figur 7-2 på side 68.

Det kan virke lidt paradoksalt, at vi har opstillet en metode for, hvordan selvforbedrende projektgrupper kan fungere, når vi gennem hele projektet har argumenteret for, at det er forkert at lægge processer ned over hovedet på andre, da der er stor risiko for, at processerne ikke vil blive accepteret. Der er dog en lille forskel, nemlig, at det vigtige ikke er selve metoden, men nærmere de egenskaber, som vi har identificeret i forbindelse med selvforbedrende projektgrupper. Den opstillede metode er blot vores bud på, hvordan disse egenskaber kan understøttes i praksis, men vi siger ikke, at det er den bedste metode. Derfor må det være op til den enkelte organisation at afgøre, om der er andre metoder, der er bedre til at fremme selvforbedrende projektgrupper.

Hvordan metoden understøtter de otte egenskaber, vi har identificeret som vigtige i forhold til selvforbedrende projektgrupper kan ses i tabel 12-2. Det viste sig, at vores første forslag til en metode, som vist på figur 4-2, indeholdt lidt for mange punkter, som lige så godt kunne slås sammen. Resultatet af det blev den metode, vi benyttede i Logimatic, og som er vist på figur 7-2. Denne metode bakker umiddelbart godt op omkring selvforbedrende projektgrupper og sikrer, at det hele tiden er projektgruppen selv, der er i fokus.

Egenskaber	Kendetegn
<i>Problemerkendelse</i>	Projektgruppen kunne selv identificere problemer ved deres reviewproces, og den anvendte metode gør det til et krav, at det er projektgruppen selv, der erkender problemerne.
<i>Procesejerskab</i>	Flytningen af procesejerskabet foregik smertefrit, og medførte hurtige forbedringer på relevante områder.
<i>Planlægning</i>	Den minimale planlægning gjorde det muligt at bevare fokus på relevante problemer, og gjorde det muligt at gennemføre små forbedringer hurtigt og uden unødvendig bureaukrati.
<i>Design</i>	Metoden lægger vægt på, at processerne skal være praktiske, så de tager udgangspunkt i de problemer, gruppen oplever i hverdagen. Det er igen med til at sikre, at processerne kan bringes til anvendelse hurtigt, og det primære fokus er derfor på etablering af praksis.
<i>Dokumentation</i>	Dokumentation af selve processen frarådes, dog kan "best practices" bruges til at beskrive gode forbedringer. Processerne fastfryses aldrig for at nedskrive dem, men rekonstrueres i stedet løbende.
<i>Normer</i>	Normer, og dermed organisationens behov, er noget sekundært, der kommer efter projektgruppens egne behov.
<i>Fokus</i>	Gennem flytningen af procesejerskabet og nedtoningen af planlægning, design og dokumentation sikres det med selvforbedrende projektgrupper, at fokus flyttes fra processerne til de udviklere, der skal bruge processerne.
<i>Læring</i>	Når projektgrupperne selv skal erkende deres problemer, så leder det frem til double-loop læring, der udgør fundamentet for effektiv procesforbedring.

Tabel 12-2: Hvordan interventionsmetoden understøtter selvforbedrende projektgrupper.

12.3 Hvordan bør en SPI gruppe fungere i en softwareorganisation med selvforbedrende projektgrupper?

Sammen med metoden for selvforbedrende projektgrupper opstillede vi også en protokol for, hvordan forholdet mellem projektgruppen og SPI gruppen bør være. Denne protokol er vist på figur 4-3 på side 47 og lægger op til, at SPI gruppen påtager sig en rådgivende rolle og opgiver procesejerskabet. Det betyder, at SPI gruppen får nogle andre arbejdsopgaver, der sigter mere mod langsigtet planlægning og mod at koordinere samarbejdet mellem projektgrupperne. Tabel 12-3 viser nogle af de væsentligste opgaver, SPI gruppen har i forhold til selvforbedrende projektgrupper. Det drejer sig om at indrette SPI gruppen på en sådan måde, at den støtter bedst muligt op omkring de selvforbedrende projektgrupper, og det gør den efter vores mening bedst ved at fungere som en rådgivende enhed, der kan hjælpe projektgrupperne når de har problemer med procesforbedringerne. Det kan konkret ske ved at formidle kontakt mellem to projektgrupper, hvor den ene projektgruppe har forbedret indenfor det område, den anden arbejder med, eller det kan ske ved at SPI gruppen

SPI gruppens opgaver
<ul style="list-style-type: none"> - Vejledere og rådgivere for projektgrupperne. - Hjælpe med problemlidentifikation. - Målstyring af forbedringsinitiativer. - Erfaringsopsamling og udbredning.

Tabel 12-3: SPI gruppens opgaver når selvforbedrende projektgrupper skal indføres i organisationen.

hjælper med at opstille mål i forhold til forbedringerne.

Afsluttende kan vi konkludere, at vi ser en bottom-up tilgang til SPI som den bedste tilgang til procesforbedring, og selvforbedrende projektgrupper er efter vores mening en måde at gøre det på i praksis. På den anden side er selvforbedrende projektgrupper heller ikke den universelle løsning på softwareindustriens procesforbedringsproblemer, da der er masser af faldgruber, der kan få forbedringerne til at mislykkes. Vi er dog af den overbevisning, at under de rette omstændigheder er selvforbedrende projektgrupper en bedre tilgang end traditionel SPI. Vores arbejde i Logimatic har vist, at de rette omstændigheder i høj grad består af at opbygge et miljø, hvor det er udviklerne selv, der har ansvaret for procesforbedring, da det bringer forbedringerne tæt på dem, der skal bruge forbedringerne, og det er med til at sikre at læringen foregår hos projektgruppen i stedet for i SPI gruppen. Hvorvidt dette er muligt i andre softwarevirksomheder kan vi ikke sige konkret, men vores opfattelse er, at de fleste softwarevirksomheder i forvejen har en forholdsvis flad struktur med en ufomelt miljø, hvor medarbejderne er vant til at tage ansvar, og derfor også kan nyde godt af selvforbedrende projektgrupper.

Litteratur

- [Aae02] Ivan Aaen. *Challenging software process improvement by design*. I Proceedings of European Conference on Information Systems (ECIS), June 2002.
- [AAMN00] Ivan Aaen, Jesper Arent, Lars Mathiassen og Ojelanki Ngwenyama. *A Conceptual MAP of Software Process Improvement*. Department of Computer Science, Aalborg University, Fredrik Bajersvej 7E, DK-9220 Aalborg, Denmark, 2000.
- [APH01] Dennis Andersen, Dorte Pedersen og Finn Hedehus. *Projektledelses- og softwareprocess forbedring i Logimatic Software A/S*. Teknisk rapport, Logimatic Software A/S, Sofiendalsvej 5, DK-9200 Aalborg SV, Danmark, 2001.
- [AS96] Chris Argyris og Donald A. Schön. *Organizational Learning II*. Addison-Wesley, 1996.
- [Bac95] James Bach. *Enough about process: What we need are heroes*. IEEE Software, ss 96–98, March 1995.
- [Bac98] James Bach. *Microdynamics of process evolution*. IEEE Computer, 31(2):111–113, 1998.
- [CKO92] B. Curtis, M. Kellner og J. Over. *Process modeling*. Communications of the ACM 35 9, ss 75–90, 1992.
- [Coc99] Alistair Cockburn. *Characterizing people as non-linear, first-order components in software development*. Teknisk rapport TR 99.05, Humans and Technology, October 1999.
- [Con93] Larry L. Constantine. *Work organization: Paradigms for project management and organization*. Communications of the ACM, ss 35–43, October 1993.
- [DM93] Bo Dahlbom og Lars Mathiassen. *Computers in Context - The Philosophy and Practice of Systems Design*. NCC Blackwell, Cambridge, Oxford, 1993.
- [FHK⁺97] Pat Ferguson, Watts S. Humphrey, Soheil Khajenoori, Susan Macke og Annette Matvya. *Results of applying the personal software process*. IEEE Computer, 30(5):24–31, May 1997.
- [Fow00] Martin Fowler. *The new methodology*. Teknisk rapport, ThoughtWorks, Inc., 2000.
- [FvH95] Bjarne Kastberg Finn van Hauen, Vagn Strandgaard. *Den lærende organisation*. Industriens Forlag, 1995.

- [Got00] Greg Goth. *The team software process: A quiet quality revolution*. IEEE Software, ss 125–127, November/December 2000.
- [Hum90] Watts S. Humphrey. *Managing the Software Process*. Addison-Wesley, Reading, Massachusetts, 1990.
- [Hum00a] Watts S. Humphrey. *Personal software process*. Teknisk rapport CMU/SEI-2000-TR-022, Software Engineering Institute/Carnegie Mellon University, November 2000.
- [Hum00b] Watts S. Humphrey. *Team software process*. Teknisk rapport CMU/SEI-2000-TR-023, November 2000.
- [HZG⁺97] James D. Herbsleb, David Zubrow, Dennis Goldenson, Will Hayes og Mark C. Paulk. *Software quality and the Capability Maturity Model*. Communications of the ACM, 40:30–40, 1997.
- [Jak98] Allan Baktoft Jakobsen. *Bottom-up process improvement tricks*. IEEE Software, (0740-7459):64–68, February 1998.
- [JF97] Jesper Erichsen Jette Frederiksen, Niels Erik Hulgård. *Den lærende organisation: at lære at lære*. Teknisk rapport, Arbejdsmarkedsstyrelsen, 1997.
- [Kol84] D.A. Kolb. *Experimental Learning*. Prentice Hall, 1984.
- [McF96] B. McFeeley. *Ideal : A user's guide for software process improvement*. Teknisk rapport CMU/SEI-96-HB-001, Software Engineering Institute/Carnegie Mellon University, February 1996.
- [Min83] H. Mintzberg. *Structure in fives: designing effective organizations*. Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [Ost87] Leon J. Osterweil. *Software processes are software too*. IEEE Computer Society Press, ss 2–13, 1987.
- [PCCW93] Mark C. Paulk, Bill Curtis, Mary B. Chrissis og Charles V. Weber. *Capability maturity model for software, version 1.1*. Teknisk rapport CMU/SEI-93-TR-024, Software Engineering Institute (SEI), 1993.
- [Pre00] Roger S. Pressman. *Software Engineering - A Practitioner's Approach*. McGraw-Hill, London, 5. udgave, 2000.
- [PUG97] Lutz Prechelt, Barbara Unger og Oliver Gramberg. *Experience report: Teaching and using the personal software process (PSP)*. Teknisk rapport, Fakultät für Informatik, Universität Karlsruhe, January 1997.
- [PWG⁺93] Mark C. Paulk, Charles V. Weber, Suzanna M. Garcia, Mary B. Chrissis og Marilyn Bush. *Key practices of the capability maturity model, version 1.1*. Teknisk rapport CMU/SEI-93-TR-025, Software Engineering Institute (SEI), 1993.

- [RJN01] Joakim Recht, Rasmus Walbum Jensen og Michael Lerskov Munk Nielsen. *Udarbejdelse, udformning og udrulning af procesmodeller*. Teknisk rapport, Institut for Datalogi - Aalborg Universitet, 2001.
- [Zah98] Sami Zahran. *Software Process Improvement - Practical Guidelines for business success*. Addison Wesley Longman, Harlow, 1. udgave, 1998.

Figurer

2-1	Den overordnede struktur af CMM, som kan danne basis for organisationers modningsproces.	11
2-2	Ideal modellen.	13
2-3	Kolbs læringscyklus.	16
2-4	Organisatorisk adfærd.	17
2-5	Læringsprocesser.	18
3-1	James Bachs forslag til en alternativ procesforbedringsmodel.	26
3-2	PSP struktur.	32
4-1	Den nye rollefordeling mellem den selvforbedrende projektgruppe og SPI gruppen når drivkraften flyttes. Den markerede cirkel illustrerer, hvor drivkraften ligger.	43
4-2	Forslag til en model til indførelse af selvforbedring.	46
4-3	Protokol for projekt- og SPI-gruppen.	47
5-1	Organisationsdiagram over Logimatic Holding A/S.	49
5-2	Organisationsdiagram over Logimatic Software A/S.	51
5-3	Organisationsdiagram over IT-Solutions.	52
5-4	Organisationsdiagram over forretningsområdet MARS.	54
6-1	Organisationsdiagram over SPI gruppen.	65
7-1	Forløbet af interventionen hos Logimatic.	67
7-2	Den faktiske model for interventionen, som vi kom frem til på det indledende møde med projektgruppen og SPI gruppen.	68
11-1	De fire kategorier af mål i forhold til en aktivitet.	104

Tabeller

2-1	Forskellige indlæringsloop.	19
2-2	Egenskaber for SPI tilgange.	22
2-3	Traditionel SPI tilgang.	24
3-1	Bachs procesforbedringsmodel i forhold til de otte egenskaber og deres kendetegn.	27
3-2	End-user SPI i forhold til de otte egenskaber og deres kendetegn.	31
3-3	PSP i forhold til de otte egenskaber og deres kendetegn.	33
3-4	TSP i forhold til de otte egenskaber og deres kendetegn.	34
4-1	Selvforbederende projektgrupper i forhold til de otte egenskaber og deres kendetegn.	43
5-1	Ansatte i Logimatic Software A/S.	52
5-2	Oversigt over designparametre i Logimatic.	59
5-3	Oversigt over situationsfaktorer for Logimatic.	62
6-1	Logimatics SPI tilgang.	66
8-1	CMM mål for review.	73
8-2	Resultater af deltagelse i det første review.	74
8-3	Retningslinjer for review i forhold til Logimatics review.	76
8-4	De problemer som gruppen identificerede ved deres reviews.	78
8-5	De overordnede mål for reviews som gruppen opstillede.	78
8-6	De mål som gruppen opstillede for indledende reviews.	78
8-7	Målene for reviews af færdige dokumenter, som gruppen opstillede.	79
8-8	Gruppens løsningsforslag til forbedring af reviews.	79
8-9	Vigtigste iagttagelser i forbindelse med opfølgningen.	81
9-1	Gruppens dagsorden for andet review.	83
9-2	Observerede ændringer i forhold til første review.	84
9-3	Retningslinjer for review i forhold til Logimatics andet review.	86
9-4	Positive og negative aspekter ved andet review ifølge projektgruppen.	87