

16. Prosessforbedring gjennom årsaksanalyse: Erfaringer fra Fjellanger Widerøe

Fjellanger Widerøe ønsket å forbedre utviklingsprosessen for å få mer fornøyde kunder. Forandringene måtte ikke ødelegge den fleksibiliteten avdelingen hadde og som var en viktig del av deres konkurransefortrinn.

For å identifisere mulige forbedringsmuligheter brukte bedriften GQM, Paretoanalyse og Ishikawadiagram.

Ut fra de data som ble samlet inn identifiserte utviklerne de viktigste problemområdene og kom opp med forslag til sjekklister og standarder som vil redusere feil- og rettekostnadene med ca. 20%

16.1. Introduksjon

Dette caset beskriver Fjellanger Widerøes erfaringer med å bruke årsaksanalyse for å forbedre utviklingsprosessen. Denne forbedringsprosessen ble gjennomført ved at de:

- Identifiserte viktige problemer på bedrifts- eller avdelingsnivå.
- Samlet inn data for å forstå disse problemene.
- Identifiserte viktige problemer på prosess- eller produktnivået.
- Analyserte disse problemene for å finne botemidler.

GQM ble brukt for å identifisere hvilke data som var nødvendige å samle inn, Paretoanalyse ble brukt for å identifisere de viktigste problemene på prosessnivået og til slutt ble årsaksanalyse kombinert med Ishikawadiagram for å finne forbedringstiltak.

16.2. Bakgrunn

Fjellanger Widerøe har totalt ca. 240 ansatte, blant disse en avdeling på 14 personer for utvikling av programvare. Avdelingen som utvikler programvare lager hovedsakelig systemer for behandling og presentasjon av geografiske data. Avdelingens størrelse er en del av dens styrke – ved å være liten kan den raskt tilpasse seg endrede omgivelser og kundekrav.

Konkurransen er hard og det å ha få eller ingen feil levert til kunden er viktig for å få tillit i markedet. Av den grunn valgte Fjellanger Widerøe – ikke overraskende – å fokusere på å redusere antall feil levert til brukerne. Forskere fra SINTEF deltok i opplæring og bruk av GQM, Pareto og Ishikawa.

Erfaring:

For små og mellomstore bedrifter er det ofte ikke nødvendig å ha noen stor, kostbar prosess for å finne ut hva som er problemene på bedrifts- eller avdelingsnivå – vanligvis holder det å spørre utviklerne.

16.3. Forbedringsprosjekt

Fjellanger Widerøe ønsket å gjennomføre et forbedringsprosjekt for å få en mer effektiv utviklingsprosess. Forbedringsprosjektet ble gjennomført med følgende trinn:

1. Identifiser de viktigste problemene innen programvare utvikling hos Fjellanger Widerøe.
2. Identifiser hvilke data som var nødvendige for å kunne forstå – og dermed gjøre noe med – disse problemene.
3. Saml inn de definerte dataene.
4. Identifiser de viktigste feilårsakene.
5. Analyser disse feilårsakene for å finne de bakenforliggende årsakene i prosessen.
6. Identifiser et sett av tiltak for å fjerne eller redusere disse årsakene

Forbedringsprosjektet ble gjennomført av utviklerne i Fjellanger Widerøe med støtte fra SPIQ-forskere ved SINTEF. Fjellanger Widerøe bestemte seg for å samle inn data fra to prosjekter med ganske forskjellige karakteristika. Dette var viktig for at dataene skulle gjenspeile spennvidden i den virkeligheten utviklerne i bedriften opplever. De to prosjektene viste ytterpunktene i bedriftens prosjekter.

- Prosjekt 1: Et velstrukturert prosjekt. Det fantes en god kravspesifikasjon, en fullstendig utviklingsplan og kunden hadde definert kravene til kvalitetssikring.
- Prosjekt 2: Dette prosjektet brukte prototyping. Kravspesifikasjonen ble utviklet etter hvert gjennom samarbeid mellom kunde og utviklerne hos Fjellanger Widerøe.

16.4. Resultater

16.4.1. Trinn 1 og 2: Identifisere problem og data

Trinn 1 – identifisere det viktigste problemet – var allerede gjort da arbeidet begynte. Under påvirkning fra SPIQ personell valgte Fjellanger Widerøe å fokusere på å legge inn færre feil under utvikling heller enn å forbedre testprosessen. Den første oppgaven bedriften sto overfor var å identifisere de data som utvikleren skulle samle inn i løpet av prosjektet. Etter råd fra SINTEF valgte bedriften å bruke:

- En brainstorming for å få frem interessante data.
- GQM prosessen og arbeidsarket til å strukturere resultatene fra brainstormingen.
- GQM feedback-møter for å validere og tolke de data som ble samlet inn.

Ved å bruke GQM prosessen og feedback-møter håpet bedriften å oppnå to ting, nemlig å få:

- Samlet inn data som ble oppfattet som interessante og viktige av de utviklerne som skulle samle dem inn. Dette skulle føre til at dataene virkelig ble registrert og på en måte som var nyttig for å finne ut hvordan bedriften skulle redusere antall feil. GQM prosessen er nyttig her.
- Kvalitetssikret de data som ble samlet inn slik at de ikke bygde konklusjonene på feil eller misforståelser hos utviklerne eller SPIQ-forskerne. Feedback-møtene hjelper til med å oppnå dette.

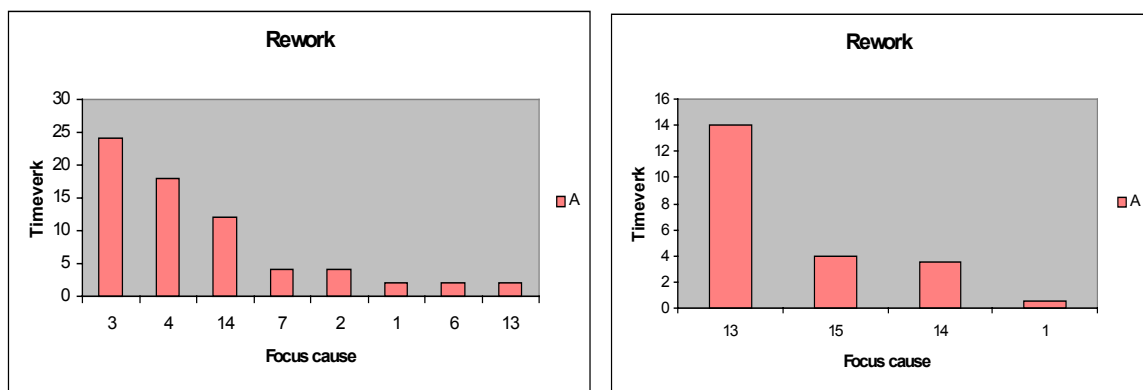
Erfaring:

GQM fungerte godt for å definere hva slags data som skulle samles inn – også for bruk i Paretoanalyse og årsaksanalyse.

16.4.3. Trinn 4: Identifisere viktige feilårsaker

For å identifisere de viktigste feilårsakene summerte vi antall timeverk brukt til retting pr. årsakskategori og viste resultatene i Paretoagrammer. Disse diagrammene er vist i figur 16.2. Vi ser at i prosjekt 1 er det fokusårsak 3 (feil bruk av språk) som dominerer, mens rework kostnadene i prosjekt 2 er dominert av fokusårsak 13 (lav detaljeringsgrad i kravspesifikasjonen). For å få en bedre forståelse av feilårsakene må vi også se på årsaker knyttet til prosjektomgivelsene. Videre analyse av data fra årsaker knyttet til omgivelsene viser at:

- Prosjekt 1 – feilårsak 3 (feil bruk av språk) har bare en årsak knyttet til omgivelsene – tidspress. Konklusjonen her er derfor enkel – tidspress gir feil i bruk av programmeringsspråket.
- Prosjekt 2. For feilårsak 13 (lav detaljeringsgrad i kravspesifikasjonen) er bildet noe mer sammensatt. De innsamlede data viser at omgivelsesårsak 6 (manglende brukerkontakt) dominerer med 13 timeverk. I tillegg ble det registrert en time på eksterne forstyrrelser – omgivelsesårsak 3.1.



Figur 16.2 Paretoanalyse av fokusårsaker for prosjekt 1 (til venstre) og prosjekt 2.

Erfaring:

Ved å bruke GQMs ide om fokus og omgivelser fikk vi frem to typer av feilårsaker. Disse to til sammen ga et godt bilde av hvordan feil kom inn i systemet og hva bedriften kunne gjøre for å få færre feil i fremtiden.

16.4.4. Trinn 5: Analysere de viktigste feilårsakene

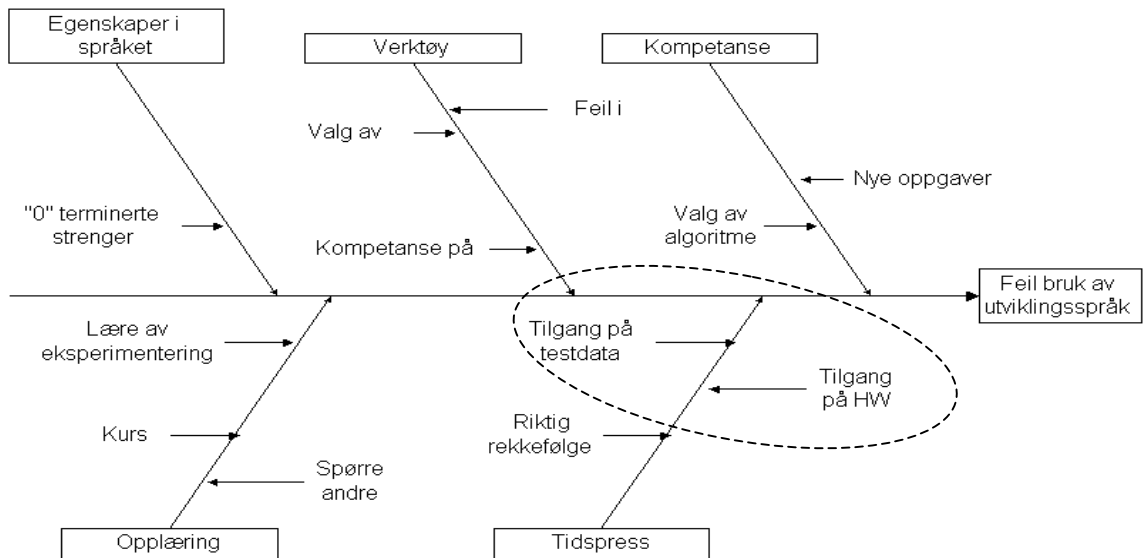
Vi valgte å starte med de viktigste feilene:

- ”For lav detaljeringsgrad i kravspesifikasjonen” og dennes viktigste årsak i prosjektomgivelsene – ”Manglende brukerkontakt”.
- ”Feil bruk av språk” og dennes viktigste årsak i omgivelsene ”Tidspress”.

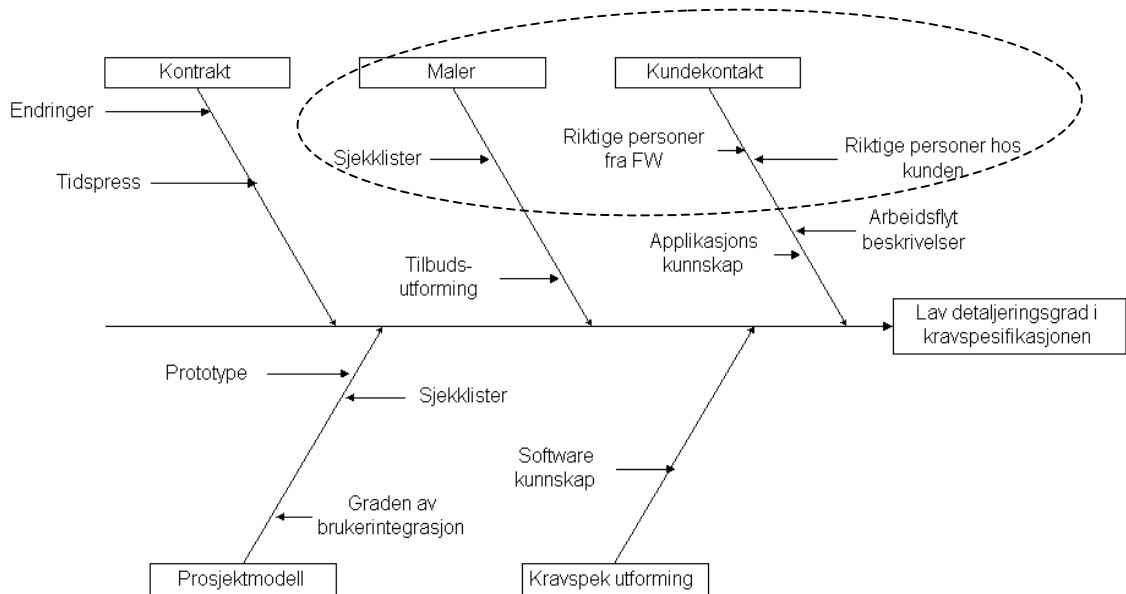
For å finne de bakenforliggende feilårsaker brukte vi brainstorming og dokumenterte fortløpende de ideene som kom frem i et Ishikawadiagram. Dette diagrammet ble også brukt til å strukturere – og delvis styre – idegenereringen. For at brainstormingen ikke skulle flyte ut i en generell diskusjon var det viktig å styre prosessen slik at man hele tiden var mest mulig konkret. Dette gjaldt både feilårsaker og mulige tiltak. Resultatet fra denne prosessen hos Fjellanger Widerøe er vist i figurene 16.3 og 16.4.

Erfaring:

Kombinasjonen brainstorming og Ishikawadiagram krever en god del styring for at den skal konsentrere seg om konkrete årsaker og tiltak.



Figur 16.3 Diagram for feilårsaken "Feil bruk av (utviklings)språk".



Figur 16.4: Diagram for feilårsaken "Lav detaljeringsgrad i kravspesifikasjonen".

Erfaringer:

Bruk av Ishikawadiagrammer er lett å lære og fungerer godt som støtte for årsaksanalyse også for en prosess i programvareutvikling. Det resulterte i en målrettet diskusjon som resulterte i forslag til konkrete forbedringer.

16.4.5. Trinn 6: Tiltak

Ut fra brainstormingen og arbeidet med Ishikawadiagrammene identifiserte deltakerne følgende årsaker:

- Tidspres: for sen tilgang på maskinvare og testdata.
- Problemer med utforming av kravspesifikasjon: behov for maler og sjekklister, samt bedre kundekontakt.

Dette ledet til følgende forslag til forbedringstiltak:

- Utviklerne skal lage et sett av sjekklister og maler til bruk ved skriving av kravspesifikasjoner.
- Alle bedriftens sjekklister og maler må være tilgjengelige over Intranett.
- Rutinemessige aktiviteter som mange gjør må standardiseres og få dokumentmaler.
- Bedriften skal sette i gang tiltak for å dokumentere, spre og gjenbruke de erfaringer innenfor programvareutvikling som hver enkelt utvikler satt inne med.
- Bedriften må ta med betingelser som angår tilgang på maskinvare (target) og testdata i utviklingskontraktene.
- Bedriften må bli mer bevisst på å koble sammen riktig person i bedriften med riktig person hos kunden.

16.5. Erfaringer

Prosjektets erfaring kan oppsummeres som følger:

- For små og mellomstore bedrifter er det ofte ikke nødvendig å ha noen stor, kostbar prosess for å finne ut hva som er problemene på bedrifts- eller avdelingsnivå – vanligvis holder det å spørre utviklerne.
- GQM fungerer godt for å definere hva slags data som skal samles inn – også for bruk i Paretoanalyse og årsaksanalyse.
- Feedback-møter var nyttige både for å skape tillit til de data som ble samlet inn og for å definere nye årsakskategorier når dette var nødvendig i løpet av prosessen.
- Ved å bruke GQMs ide om fokus og omgivelser fikk vi frem to typer av feilårsaker. Disse to til sammen ga et godt bilde av hvordan feil kom inn i systemet og hva vi kunne gjøre for å få færre feil i fremtiden.
- Bruk av Ishikawadiagrammer er lett å lære og fungerer godt som støtte for årsaksanalyse i programvareutvikling. Det resulterte i en målrettet diskusjon som resulterte i forslag til konkrete forbedringer.
- Kombinasjonen brainstorming og Ishikawadiagram krever en god del styring for at den skal konsentrere seg om konkrete årsaker og tiltak.

16.6. Konklusjoner

Det store spørsmålet er: Lønner dette seg? Svaret kommer an på om man ser det på kort (dette prosjektet) eller lang (noen år) sikt og på hva man tar med av kostnader og nytte. Det etterfølgende er en mulig måte å se det på.

Kostnader:

- Beslutningen om å prøve ut denne måten å forbedre prosessen på – 2 timeverk.

- Kurs i GQM og utarbeiding av skjema for datainnsamling – fire personer i en dag – 32 timeverk.
- Datainnsamling – 10 minutter pr. rapportert feil – pluss to feedback-møter på en time – ca. 12 timeverk.
- Paretoanalyse og rapportering – fire timeverk.
- Kurs i bruk av Isikawadiagram og analyse av de innsamlede data fire personer i en dag – 32 timeverk.

Totalt brukte Fjellanger Widerøe altså 82 timeverk eller ca. to ukeverk på å gjennomføre forbedringsprosjektet.

Nytten vil avhenge av hvor mange feil de kan unngå i senere prosjekter. De feilårsakene vi prøvde å gjøre noe med sto for i alt 37 timeverk med feilretting. Det synes rimelig å anta at vi kan redusere dette med 50% ved de forslåtte tiltakene. Innsparingen er altså ca. 18 timeverk pr. prosjekt, gitt at prosjektene er av samme type og størrelse som de prosjektene som er brukt i dette caset. Dette tilsvarer 20% av all tid som gikk med til feilretting. Grovt regnet skulle vi altså trenge seks prosjekter for å ”tjene inn” de medgåtte kostnadene. Dersom senere prosjekter er større vil det antakelig gå kortere tid.

I tillegg kommer nytte som det er vanskelig å tallfeste, men som på lang sikt antakelig vil være viktigere enn de rene besparelsene i feilrettingskostnadene, f.eks.:

- Forståelse og interesse for prosessforbedring blant utviklerne og ledelse.
- Kunnskap om GQM, Paretoanalyse og Ishikawadiagram og deres rolle i problemløsning generelt og i prosessforbedring spesielt.
- Et mer kritisk syn på egne arbeidsmetoder – hvorfor gjør vi dette slik?

16.7. Videre planer

Arbeidet ble avsluttet selv om det var stor interesse blant utviklerne. Ledelsen i Fjellanger Widerøe valgte imidlertid å nedprioritere dette arbeidet på grunn av andre, mer presserende aktiviteter. Avdelingen er nå solgt ut av Fjellanger Widerøe.

16.8. Anbefalt litteratur

Hans Jørgen Lied og Tor Stålhane

Experience from Process Improvement in a SME
EuroSPI'99 i Pori, Finland 26.-27. oktober, 1999.

Denne artikkelen beskriver hele prosjektet, inklusiv en detaljert beskrivelse av dataanalysen og de metodene som ble brukt der.

Forbedring gjennom årsaksanalyse – SPIQ bruksnotat, inneholder en god beskrivelse av hvordan man kan gjennomføre en årsaksanalyse.

Etablering av måleplaner – SPIQ bruksnotat, inneholder en god beskrivelse av hvordan man bruker GQM til å definere, samle inn og validere data som en del av programvareutvikling.