

Månedsbrev 2

Kravspecifikation med use cases – en god ide?

Use cases (eller på dansk: Brugsmønstre) beskriver vigtige egenskaber ved IT-systemer, og erfaringen viser, at brugerne kan forstå use cases. Så det er en god ide at lave use cases, når man analyserer IT-systemer.

Nogle metoder anbefaler at lade use cases indgå i kravspecifikationen. Men det er en sammenblanding af dokumenter, som skaber flere problemer, end det løser.

Lad os først præcisere begreberne og se på et eksempel. **Kravspecifikationen er aftalen** mellem udviklerne og brugerne om, hvad der skal leveres. Sådant en aftale skal være kort og klar. Der kan for eksempel stå i den:

Krav 3.11: En kunde skal via www kunne se status på sine ordrer.

En **use case** er en *detaljeret* beskrivelse af en afgrænset interaktion mellem systemet og en bruger (eller et andet system). Use cases kan beskrives i tekst eller i diagrammer. For eksempel kan man i detaljer beskrive, hvordan en bruger klikker sig frem til at se status på sine ordrer, herunder variationsmuligheder ved fejl og afbrydelser, osv.

Jamen, så må det da også være godt at have use cases i kravspecifikationen? – **NEJ!** For så er kravspecifikationen ikke længere kort og klar. Alle detaljerne betyder i praksis:

- at brugere ikke gider (nær)læse kravspecifikationen, der nu er blevet 5-10 gange større og er fyldt med oplysninger, der er selvfølgelig for dem,
- at det tager meget længere tid at skrive kravspecifikationen,
- og at brugere og udviklere låser sig fast på forhold i brugergrænsefladen, som først kan fastlægges gennem afprøvning.

”Jamen, jeg er nødt til at skrive disse detaljer i kravspecifikationen. Når den er godkendt, kommer der en programmør på opgaven, og han snakker ikke med brugerne. Så hvis kravspecifikationen ikke er detaljeret, så kan han ikke lave systemet” – sådan hørte jeg engang en erfaren analytiker indvende.

Metodicas månedsbrev: SUBSCRIBE & UNSUBSCRIBE

- vil du slettes af listen, ændre adresse, tilføje en kollega til listen?
- eller ønsker du fremover modtage månedsbrevet per e-mail?

Så send en e-mail til metodica@post4.tele.dk med de relevante data.

Hertil vil jeg sige, at hvis det virkelig ikke er muligt for programmøren at snakke med brugere, så er analytikerens nødt til at give programmøren detaljerede informationer. Men de skal stå i **et andet dokument** end kravspecifikationen. For eksempel et analyse-dokument. Her kan analytikerens skrive sine use cases, efterhånden som de bliver lavet. Og dette analysedokument kan programmøren så bruge til at opbygge sin forståelse.

Kravspecifikationen og analysedokumentet

Pointen er altså, at der er brug for to dokumenter. Forskellene mellem dem er vist i nedenstående tabel.

<i>Dokument</i>	<i>Formål</i>	<i>Indhold</i>	<i>Form</i>
Kravspecifikation	Fastholde aftalen mellem udviklere og brugere	Aftalte beslutninger	Tekst
Analysedokument	Fastholde udviklernes forståelse	Beskrivelser af systemet og dets omgivelser (beslutninger, skitser, ideer)	Tekst og diagrammer (heriblandt use cases)

Dokumenterne hænger sammen på den måde, at kravspecifikationen er et ekstrakt af aftalerne i analysedokumentet.

Hvis man blander de to dokumenttyper sammen – for eksempel ved at indlægge use cases i kravspecifikationen – bliver resultatet et analysedokument. Hermed har man i praksis afskaffet kravspecifikationen.

Andreas Munk-Madsen

P.S. I vores bog "Objektorienteret Analyse & Design" (tredje udgave, 2001) er der på side 301 en diskussion af forskellen mellem en kravspecifikation og et analyse-dokument - illustreret ved et eksempel.

Kursus: Kravspecifikation og kravstyring. 28.-29. okt. 2002

På kurset lærer deltagerne:

- at skrive praktisk anvendelige kravspecifikationer,
- at afdække krav,
- at håndtere ændringer til kravene,
- og at sikre at kravene indfries.

På www.metodica.dk kan du læse mere om kurset, og om vores bøger og kurser i Projektledelse og Objektorienteret analyse og design.