

Use-case eksperiment

Forsøgsleder: Søren Lauesen, slauesen@itu.dk

Vi IT-folk skriver ofte en eller anden slags use-cases. Vores use-cases kan være meget forskellige, fx UML-stil, tasks, scenarier eller user stories. Hvad er bedst? Det er svært at vide hvis vi sammenligner på tværs af projekter. I dette eksperiment vil vi se på ét virkeligt projekt og sammenligne hvordan forskellige professionelle specificerer use-cases på deres foretrukne måde.

Arbejder du professionelt med IT, inviterer vi dig til at deltage i eksperimentet. Vi regner med at du skal bruge mellem 3 og 20 arbejdstimer.

Vi vil sammenligne løsningerne og publicere resultatet. Du ser selvfølgelig resultatet inden det publiceres, kan kommentere det, og kan bestemme om du vil have dit navn og en kort profil offentliggjort.

Her er projektet: En virksomhed vil anskaffe et støttesystem til deres hotline. Din opgave er at skrive dele af kravene. (Detaljerne står på de næste sider). Der er både en dansk og en engelsk version. Valget er dit, men det gør international publikation lettere for os hvis det er på engelsk.

1. Send en email til Søren Lauesen <slauesen@itu.dk>, hvis du har tænkt dig at besvare opgaven. Så fortæller vi om vi har kapacitet til at behandle den løsning du senere indsender. Opgiv dit navn og antal år du har arbejdet professionelt med IT. Du kan også stille afklarende spørgsmål.
2. Skriv de use-cases (eller lignende) der er nødvendige ved dette projekt.
3. Vælg den form for use cases du bedst kan lide eller den du er vant til at bruge. Fortæl os hvorfor du har valgt den.
4. Tilføj gerne diagrammer eller beskrivelser af data hvis du finder det hensigtsmæssigt.
5. Forsøg ikke at skrive kvalitetskrav, fx om sikkerhed, svartider, brugervenlighed, etc.
6. Projektets detaljer afspejler interviews og feltstudier. De forsøger ikke at være fuldstændige. Du kan tilføje ting du mener mangler, fx behov for at kunne slette en henvendelse til hotline. Men lad være med at tilføje ting fra beslægtede områder, fx en multi-national hotline. Kunden interesserer sig kun for støtte til den nuværende situation.
7. Dine use-cases må ikke fylde mere end 10 sider. Skriv evt. nogle af de mere interessante i detaljer og skriv en liste over resten af dem.
8. Send resultatet til slauesen@itu.dk. Medsend gerne kommentarer og et omtrentligt tidsforbrug.
9. Deadline for din besvarelse er mandag 28. september 2009.

Støttesystem til hotline

Et firma med omtrent 1000 IT-brugere har sin egen hotline (help desk). De er utilfredse med deres nuværende open-source system til hotline-støtte, og vil anskaffe et nyt system. De ved ikke om de skal modificere det de har eller købe et nyt.

En planlægger har interviewet interessenterne og observeret hvad der faktisk sker. Du finder hans rapport nedenfor. Baseret på den er det din opgave at specificere dele af kravene til det nye system, nemlig use-cases (eller lignende) og om nødvendigt datakravene.

IT-brugerne

Brugerne løber ind i problemer af mange slags. De har fx glemt deres password og kan overhovedet ikke komme i gang med dagens arbejde, eller printereren mangler toner, eller de kan ikke huske hvordan de får Word til at skrive i to spalter. Der kan også være tale om at noget er gået i stykker, fx en printer, eller at bestille et program som man har brug for.

Det letteste er at ringe til hotline eller besøge deres kontor. Så kan man i mange tilfælde få løst sit problem med det samme. Hotline vil dog helst have sagen pr. email til *hotline@* ... Sommetider er det dog umuligt, fx hvis problemet er at man har glemt sit password til computeren.

Hvis man ikke får løst sit problem med det samme, er det irriterende at man ikke ved hvornår det bliver løst. Hvor tit skal man fx ud til printereren for at se om der nu er kommet toner i? Ofte sker det at problemet faktisk er blevet løst, men brugeren ved det ikke.

Med det nuværende system kan brugeren finde sin sag i støttesystemet og se hvor langt sagen er kommet, men det er ubekvem, og hvor tit skal man kigge der?

Firmaet har en medarbejderdatabase med medarbejderens navn, telefon, email-adresse, brugernavn og password. Støttesystemet skal trække på den og ikke vedligeholde sin egen brugerliste.

Hotline

Hotline er ofte organiseret sådan at nogle medarbejdere er *første niveau* og andre er *andet niveau*. Første niveau modtager sagerne over telefon eller email, eller når brugeren kommer personligt.

I travle perioder kan en første-niveau medarbejder modtage 50 sager pr. dag. Ca. 80% af sagerne kan klares med det samme, og netop for disse sager kniber det med at få registreret dem til statistisk brug.

De resterende 20% af sagerne gives videre til andet niveau. Ofte kan første niveau ud fra meddelelsen og snak med brugeren give sagen en prioritet og evt. et skøn over hvornår sagen er færdig. (Erfaringen viser at brugerne ikke selv skal kunne vælge en prioritet, fordi de er tilbøjelige til at give alting høj prioritet).

Halvdelen af andet niveau sager er i princippet lette, men kan ikke klares med det samme. Man skal ud at se på sagen, fx skifte toner i printeren eller hjælpe brugeren ved hans egen PC. Som regel er det afslutningen på sagen, men det kan også blive til en længere sag, fordi der skal skaffes en ekspert eller eksterne ressourcer.

Omkring 10% af sagerne tager længere tid fordi de skal overtages af en andet-niveau medarbejder med særlig ekspertise, eller fordi der skal bestilles dele eller ekspertise udefra. Overdragelsen af sagen går let galt. Hotline medarbejderen lægger en gul seddel på ekspertens bord, men ofte forsvinder sedlen eller eksperten misforstår hvad sagen handler om. Det er derfor en god idé at eksperten personligt eller pr. telefon kan forhøre sig om sagen hos den oprindeligt modtog den eller hos brugeren selv.

Der er 10-15 medarbejdere der lejlighedsvis eller hele tiden tager sig af hotline. Man kender hinanden og ved hvem der er ekspert i hvad. Medarbejderne skifter ofte mellem at være første og andet niveau, bl.a. for at få lidt afveksling. Det sker desværre at en første-niveau medarbejder flytter til andet niveau uden at bemærke at ingen andre tager sig af første niveau.

Nogle gange tabes en sag fordi en hotline-medarbejder har taget fat på sagen, men inden den er færdigbehandlet bliver han syg eller skal på ferie.

Lederne vil have statistikker over hvad der giver flest problemer og tager mest tid, så man kan forebygge problemerne. Med det nuværende system er det desværre besværligt at registrere de oplysninger der er nødvendige til statistik. Med disse data ville man også kunne måle hvor lang tid hotline er om at behandle sagerne.

Når der er travlt, kan der være 100 sager der skal gøres noget ved (uafsluttede sager). Så er det svært for den enkelte at overskue hvilke sager han er i gang med og hvilke der haster mest.

Det eksisterende støttesystem

Det nuværende system samler alle mails sendt til *hotline@* ... og anbringer dem i en liste af igangværende sager. Figur 1 viser et eksempel på sådan en liste. Man kan se sagsnummeret (Req), brugeren (Sender, fx hans email adresse), hvilken hotline-medarbejder der har sagen (Owner), og hvor gammel sagen er (Age).

Man kan også se hvornår nogen sidst har set på sagen (ActOn). Det er dog ikke så nyttigt. Det ville være vigtigere at se hvornår sagen burde være afsluttet ud fra hvad brugeren forventer. Man kunne evt. lade systemet give en advarsel (en "rykker") hvis sagen ikke er afsluttet på det forventede tidspunkt.

Når en bruger henvender sig telefonisk eller personligt, opretter hotline-medarbejderen en ny sag. Den vil så optræde i den normale liste af sager. Når hotline-medarbejderen klarer sagen straks, lader han desværre ofte være med at registrere den, fordi det er for besværligt. Det giver misvisende statistikker.

Som man kan se, kan systemet ikke altid håndtere danske bogstaver, og det er ikke intuitivt hvad de forskellige funktioner betyder.

Princippet er dog godt nok. En hotline-medarbejder har hele tiden listen på skærmen så han kan følge med i hvad der sker. Medarbejderen kan tage en indkommende sag ved at åbne sagen (omtrent som man åbner en email), evt. påtage sig at behandle den, fx ved at give løsningen med en email, klassificere sagen efter årsag (printer, login, etc.), give den en prioritet, overdrage den til en anden, osv. Når sagen er afsluttet markerer medarbejderen det, og sagen står så ikke længere på den normale liste af åbne sager.

Som man kan se bliver *Status* ikke brugt. Det er for besværligt og de nuværende status-navne er forvirrende. Nogle hotline medarbejdere har dette forslag til hvad status for en sag kunne være:

- 1. niveauSagen skal tages af en 1. niveau medarbejder, fx fordi den lige er modtaget af systemet.
 - 2. niveauSagen skal tages af en 2. niveau medarbejder.
 - BehandlesSagen behandles af en medarbejder (Owner). Medarbejderen kan godt skifte mellem niveauerne i løbet af behandlingen.
 - Parkeret.....Sagen afventer noget, fx en leverance udefra, og hotline skal ikke gøre noget foreløbig.
 - RykkerSagen er ikke afsluttet til tiden eller den ventede leverance udefra er ikke kommet til tiden.
 - Afsluttet.....Sagen er afsluttet, men den kan genoptages, fx hvis brugeren ikke mener problemet er løst.
- Åbne sager er dem der hverken er parkeret eller afsluttet.

Af hensyn til statistik og selve sagsbehandlingen ville det være nyttigt at kunne se hvornår en sag har skiftet status.

Det eksisterende system kan indstilles til at registrere sagens årsag, men det bevirker at systemet kræver en årsag lige fra starten, selvom den virkelige årsag ofte først kendes senere. Desuden er der nogen der skal definere en liste af mulige årsager, hvilket ikke er så let. Derfor registrerer man ikke årsager, og statistikkerne bliver dårlige.

For længere sager kan der undervejs komme nye oplysninger, både fra den oprindelige bruger og fra hotline-medarbejderne. I det nuværende system er det besværligt at registrere oplysningerne, og resultatet er at de ikke er tilgængelige for sagsbehandleren.

En hotline-medarbejder kan indstille systemet så det sender en almindelig email til ham, når der er kommet noget han skal kigge på. Det er især nyttigt for andet-niveau medarbejdere, som prøver at lave andre opgaver når der ikke lige er bud efter dem.

Systems Support Online [System Administration] - Microsoft Internet Explorer

Address <http://mail.it-c.dk/cgi-bin/wreq-sysadm/req?list-1>

sysadm (115) Active Search... List all... List 25... Show... Reload... faqWin... noFrame...

Req	Owner	Priority	Status	Age	ActOn	Sender	Subject
6247		0		43min		knt	=?ISO-8859-1?Q?Gennem=E5_hvert_=F8svelseslokale?=&
6246		0		54min		erikdam	Paper Jam. 3A.34
6241		0		6hr	5hr	janchr	Vedr afgangsmappen til personalemappen
6235	kaj	0		2day	6hr	bm	followsymlinks
6231		0		3day	1day	ena	Tomcat og servlets
6224		0		4day		mmm	=?ISO-8859-1?Q?Frat=E6delse?=&
6222	knt	0		4day	47min	bitta	[Fwd: [Fwd: adgang til eksamenskontoret request system]]
6221	knt	0		4day	2hr	carsten	Telefonsystem
6217		0		4day		bruin	Flere systemadgange til ny SWU-studievejleder
6216	knt	0		4day	0sec	knt	Evaluering af hardware og software

[next...](#)

[Login](#) [Logout](#) [Credits](#) [Config](#) [Statistics](#) [DelUsers](#) [Add](#) (0sec)

Fig. 1. En liste af hotline sager i et eksisterende støttesystem.
A list of hotline requests in an existing support system.

Hotline support system

A company with around 1000 IT users has its own hotline (help desk). They are unhappy with their present open-source system for hotline support, and want to acquire a better one. They don't know whether to modify the system they have or buy a new one.

An analyst has interviewed the stakeholders and observed what actually goes on. You find his report below. Based on this, your task is to specify some of the requirements to the new system: use cases (or the like) and if necessary the data requirements.

IT users

The users encounter problems of many kinds. For instance they may have forgotten their password, so they cannot start their work; or the printer lacks toner; or they cannot remember how to make Word write in two columns. The problem may also be to repair something, for instance a printer, or to order a program the user needs.

The easiest solution is to phone hotline or walk into their office. In many cases this solves the problem right away. However, hotline prefers to receive the problem request by email to *hotline@* ... Sometimes this is impossible, for instance if the problem is that the user has forgotten his password.

If the user cannot have his problem solved right away, it is annoying not knowing when it will be solved. How often will he for instance have to go to the printer to check whether it has got toner now? In many cases the problem has been solved, but the user doesn't know.

The present support system allows the user to look up his problem request to see what has happened, but it is inconvenient and how often should he look?

The company has an employee database with the user's name, phone number, email address, user name and password. The support system should retrieve data from it and not keep an employee file of its own.

Hotline

Hotline is staffed by *supporters*. Some supporters are *first line*, others are *second line*. First-line supporters receive the requests by phone or email, or when the user in person turns up at the hotline desk.

In busy periods, a first-line supporter may receive around 50 requests a day. Around 80% of the requests can be dealt with right away, and for these problems it is particularly hard to ensure that supporters record them for statistical use.

The remaining 20% of the requests are passed on to the second line. Based on the problem description and talks with the user, first line can often give the request a priority and maybe an estimated time for the solution. (Experience shows that users shouldn't be allowed to define the priority themselves, because they tend to give everything a high priority.)

Half of the second-line requests are in principle easy, but cannot be dealt with immediately. The supporter may have to move out of the office, for instance to change toner in the printer or help the user at his own PC. Usually this ends the request, but it may also turn into a long request because a specialist or external resources are needed.

Around 10% of all problems are long requests because the problem has to be transferred to a hotline person with special expertise, or because spare parts and expertise have to be ordered from external sources. Transferring the problem often fails. The supporter places a yellow sticker on the expert's desk, but the stickers often disappear. Or the expert misunderstands the problem. For this reason it is important that the expert in person or by phone can talk with the supporter who initially received the request, or with the user himself.

There are 10-15 employees who occasionally or full time serve as supporters. They know each other and know who is expert in what. The supporters frequently change between first and second line, for instance to get variation. It happens, unfortunately, that a supporter moves to second line without realizing that nobody remains in first line.

The request is sometimes lost because a supporter has started working on it, but becomes ill or goes on vacation before it is finished.

Managers ask for statistics of frequent and time-consuming requests in order to find ways to prevent the problems. However, with the present system it is cumbersome to record the data needed for statistics. Gathering this data would also make it possible to measure how long hotline takes to handle the requests.

In busy periods, around 100 requests may be open (unresolved). Then it is hard for the individual supporter to survey the problems he is working on and see which problems are most urgent.

The existing support system

The present system automatically collects all emails sent to *hotline@ ...* and put them in a list of open requests. Figure 1 shows an example of such a list. You can see the request number (Req), the user (Sender, e.g. his email), the supporter working on it (Owner), and how long ago the request was received (Age).

You can also see when someone last looked at the request (ActOn). However, this is not really useful. It would be better to see when the request should be closed according to what the user expects. It would be nice if the system warned about requests that are not completed on time.

When a user calls by phone or in person, the supporter creates a new request. It will appear in the normal list of requests. However, when the supporter can resolve the request right away, he often doesn't record it because it is too cumbersome. This causes misleading statistics.

As you can tell from the figure, the system cannot handle national letters (Danish), and it is not intuitive what the various functions do.

Anyway, the basic principle is okay. A supporter keeps the list on the screen so he can follow what is going on. He can open an incoming request (much the same way as you open an email), maybe take on the request (for instance by sending a reply mail), classify the case according to the cause of the problem (printer, login, etc.), give it a priority, transfer it to someone else, etc. When the request has been completed, the supporter closes it, and the request will no longer be on the usual list of open requests.

As you can see on the figure, *Status* isn't used at all. It is too cumbersome and the present state names are confusing. Some of the supporters have proposed to distinguish between these request states:

First lineA first-line supporter must take on the request, for instance because it just arrived.

Second lineA second-line supporter must take on the request.

Taken.....The request is handled by a supporter (the owner). The owner may change from one line to another while he is handling the request.

Parked.....The request awaits something, for instance an external delivery, and hotline need not do anything meanwhile.

ReminderThe request hasn't been closed in due time, or the external delivery wasn't received in due time.

Closed.....The request has been handled. However, it may be opened again, for instance because the user doesn't think the problem has been solved.

Open requests are those that are neither parked, nor closed.

For statistical purposes and to support the resolution of the request, it would be useful to keep track of when a request has changed state.

The present system can be configured to record a problem cause, but then the system insists that a cause be recorded initially, although the real cause may not be known until later. In addition somebody must set up a list of possible causes, and this is a difficult task. As a result, causes are not recorded, and statistics are poor.

While a long request is handled, it may receive additional information from the original user as well as from supporters. In the present system it is cumbersome to record this, and as a result the information may not be available for the supporter who later works on the request.

A supporter can set the system to send an ordinary email to himself when he has to look at some request. This is particularly useful for second-line supporters who concentrate on other tasks until they are needed for support.