SL 16-06-2009

Use Case Experiment

Investigator: Soren Lauesen, slauesen@itu.dk

We - the IT professionals - often write some kind of use cases. Our "use cases" may be quite different, e.g. UML-style, tasks, scenarios, or user stories. Which kind is best? It is hard to tell when we compare across different projects. In this experiment we will look at a single real-life project and compare how various professionals specify the use cases in their preferred way.

If you work professionally with IT, we invite you to participate in the experiment. We expect that you spend between 3 and 20 work hours.

We will compare the specifications and publish the result. You will of course see the result before it is published. You may comment on it and decide whether you want your name and a short profile to be published.

The project is this: A company wants to acquire a support system for their hotline. Your job is to specify part of the requirements. (The project details are shown on the next pages.)

1. If you want to participate in the experiment, send an email to Soren Lauesen <slauesen@itu.dk>. We will tell you whether we have capacity to handle your later submission. Provide your name and the number of years you have worked as an IT professional. You may also ask questions for clarification.
2. Write the use cases (or the like) that are necessary for the acquisition project.
3. Choose the kind of use cases you prefer or are familiar with. Tell us why you chose it.
4. You may add diagrams or descriptions of data if you find it useful.
5. Don't attempt to write quality requirements such as security, response time, or ease-of-use.
6. The project details reflect interviews and studies in the domain. They are not supposed to be complete. You may include something you believe is missing, e.g. the need to delete a request. However, don't add information from related domains, for instance multi-national hotlines. The customer cares only about support of his present situation.
7. Your use cases may not take up more than 10 pages. If necessary, write some of the more interesting ones in detail and provide a list of the rest.
8. Send the result to slauesen@itu.dk. Send also an estimate of the time used and any additional comments.
9. Deadline for your reply is Monday 28th September 2009.

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:

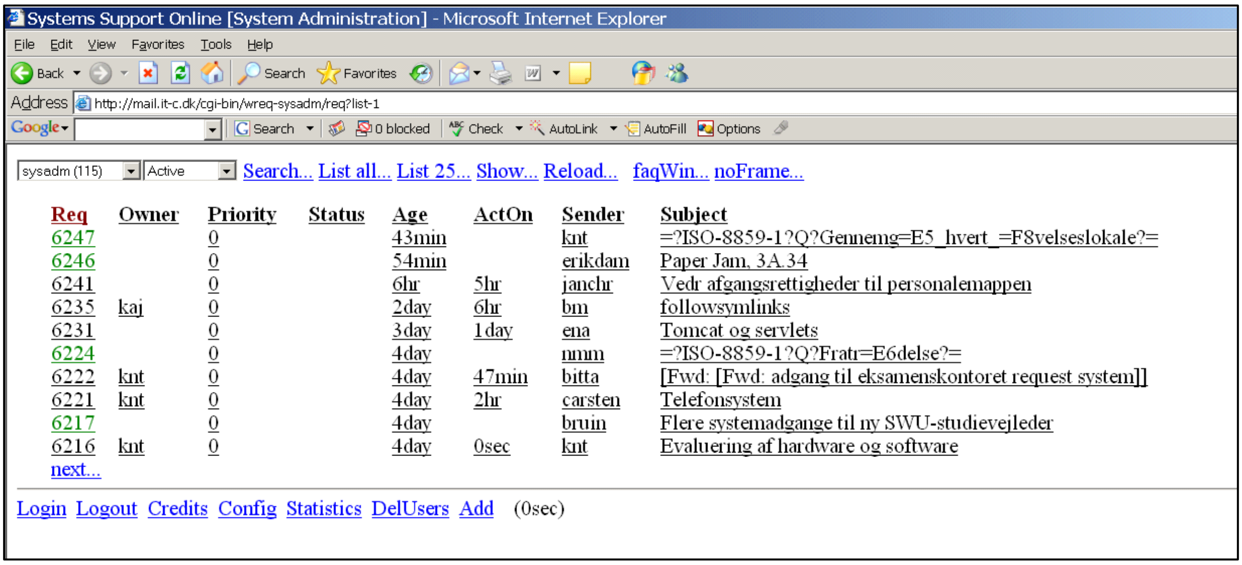
First line .... A first-line supporter must take on the request, for instance because it just arrived.
Second line .... A second-line supporter must take on the request.
Taken .... The request is handled by a supporter (the owner). The owner may change from one line to another while he is handling the request.
Parked .... The request awaits something, for instance an external delivery, and hotline need not do anything meanwhile.
Reminder .... The 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.



Systems Support Online [System Administration] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Google
Search
0 blocked
Check
AutoLink
AutoFill
Options

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?Gennemg=E5_hvert_=F8velseslokale?=
6246		0		54min		erikdam	Paper Jam, 3A.34
6241		0		6hr	5hr	janchr	Vedr afgangsrettigheder til personalemappen
6235	kaj	0		2day	6hr	bm	followsymlinks
6231		0		3day	1day	ena	Tomcat og servlets
6224		0		4day		nmm	=?ISO-8859-1?Q?Fratr=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)

Figure 1. A list of hotline requests in an existing support system.