Requirements specification for
Hotline Support System
(below called the Hotline system)

Customer
Z-Department

Supplier
...

The delivery comprises
Software, support and maintenance for the Hotline system

Contents

Change log

Version 1: 08-04-2019. First version.
Version 2: 20-04-2019. Minimum requirements changed. (Might let excessively expensive solutions pass and discard good solutions.)
B5 Highest net benefit, replaced by "most score points per dollar".
B5. Example prices adjusted.
Various misprints corrected.
Version 3: 13-10-2020: D6 CREDO added.

A. Background and supplier guide

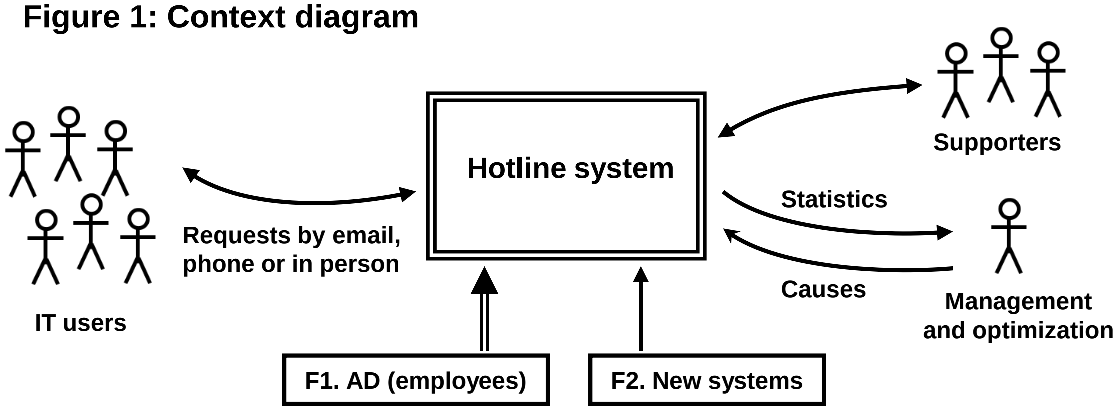
A1. Background and vision

The Z-Department is a government organization. It has its own IT hotline (help desk) with 10-15 supporters and around 1000 IT users. They are unhappy with their present open-source system for hotline support and want to buy a better one.

The customer expects that the supplier has a COTS system (Commercial-Off-The-Shelf system) that can meet most of the requirements. In return, the customer is willing to change his work processes to a reasonable extent, as long as the business goals are met (see section B1).

The future situation is illustrated with this context diagram. The data flow on the arrows in the direction of the arrow. Double lines show the supplier's responsibilities: The box with double-line border shows the system to be delivered. Double-line arrows show integrations to be delivered. Improvements are expected in communication with IT-users and supporters, in access to statistics and in maintenance of a catalog of request causes.

Figure 1: Context diagram



A2. Supplier guide

This section explains the requirements format. Everything written by the supplier must be in red.

All requirements are written in tables:

- Column 1 is the requirement (the customer's demand - what he wants the system to support).
- Column 2 may contain the customer's solution example. In the supplier's reply, column 2 is a short description of the proposed solution. It must be in red.
- Column 3 (Code) may be the customer's rating of the proposal, test references, etc.

The requirements are organized in chapters according to their kind, e.g. Chapter C about user tasks to be supported, Chapter H about security. Within each chapter, the requirements are written in tables, e.g. a table with requirements relating to a specific task. A reference to requirement 3 in section L2 looks like this: L2-3.

The customer's solution examples are only for inspiration. The supplier is welcome to suggest completely different solutions. They become legal requirements when both parties have accepted them. However, if the accepted solution does not meet the demands stated in column 1 in a reasonable manner, column 1 has priority. See contract §9.1.

Text outside tables

Text outside the tables can serve several purposes:

A. **Assumptions** behind the requirements, for instance that the task must be supported for this kind of users, this frequency of use, etc.

B. **Requirement notes** that elaborate column 1 in the table. In principle, they should be inside the table, but they don't fit well. One example is a list of access rights to the system.

C. **Solution notes** that elaborate column 2 in the tables. They are not requirements but example solutions. One example is various ways a user can look up a code in a table.

D. **Examples** and other information to help the reader understand the requirements.

Options

Customers often write requirements that turn out to be very expensive to meet. In such cases, the supplier is welcome to offer options: an expensive one that fully meets the customer's requirements and one or more that only partly meet them. The requirement in the table below is an example.

When the proposal has several areas, e.g. availability and response time, each with several options, it is important that the customer can assess them independently.

Open target

Chapter L has many "open target" requirements. As an example, the customer may ask for high system availability, but isn't sure what it will cost. So he states what he expects and leaves it to the supplier to suggest something. In the proposal, it becomes requirement L2-2 with two supplier options:

L2. Availability

Availability requirements:	~~Example solution~~ Proposed solution:	Code:
2. In the period from 8:00 to 17:00 on weekdays, the system must have high availability.	In these periods the availability is at least ____%. (The customer expects 99.5% or better). Base version: 99.0% Option A5-1: 99.8% (around 2 m$/year, see app. 2) Option A5-2: 99.95% (around 3 m$/year, see app. 2)	

Notice that the customer has written "99.5 *or better*". It means that the supplier earns additional points for both options. If the supplier had omitted "*or better*", none of the options would earn more than 99.5%.

The template format

The template is an MS-Word document. It uses *Heading 1, Heading 2* and sometimes *Heading 3*, plus a special heading style, *Heading no number.* They automatically generate the table of contents. In order to improve the overview, some headings have a forced page break. It may be changed through

Home → Paragraph → Line and Page Breaks → Page break before

Tables use the embedded table style *Requirement Table*. It has borders of 3/4 point. The cells have top and bottom cell margins of 0.5 mm. Column 1 has a hanging indent of 0.75 cm. Within a table cell, you tabulate with Ctrl+Tab, since Tab alone moves the cursor to the next cell.

A3. Customer options

1. Conversion of existing data on open support requests. See J4.

A4. Overall solution

. . .

A5. Supplier options

. . .

B. High-level demands

This chapter explains how the customer's business goals are met through the requirements, how to mitigate high-risk requirements, and how to compare proposals.

B1. Flow: Handling a request

The main flow is handling a support request. The table below is the general, logical flow of the process. Many of the steps can be omitted or repeated several times during the process (e.g. step 3, 4 and 5).

The logical flow is carried out in physical tasks, where a user or a supporter for a short period of time works with the request without essential interruptions. Column 2 shows the related tasks and subtasks for each step in the flow. Chapter C shows the details.

Steps in handling a request for support	**Tasks and subtasks**
1. The IT user reports a problem.	C1
2. A first-line supporter has a first look at the request, and in 80% of the cases solves the problem immediately and replies to the user.	C10-2, 4, 6
3. A second-line supporter handles the case and replies to the user in 10% of the cases.	C11-3, 4, 5, 7q
4. A supporter with special expertise or an expert outside the organization handles the request.	C11-5, C12
5. The IT user may provide more information.	C2-3
6. The IT user receives the reply and checks that the problem is solved.	C2-1
7. The IT user may reopen the request.	C2-5

In the general flow above, we haven't mentioned time monitoring at the various steps. It is described in tasks and subtasks.

The flow description is not requirements, but a cross check between the logical flow and the tasks.

B2. Business goals

The customer's reason to acquire the system is to reach some business goals. The customer expects that the system contributes to the goals as stated below. The supplier can rarely reach the goals alone. Customer contribution is needed too. This means that the goals are **not requirements** to the supplier. They are shown in a table only to provide overview.

All goals are important and the sooner they can be met, the better. Some goals are crucial to meet at a specific date, for instance for business or legal reasons. Such deadlines are shown in the table.

Goals for the new system	**Solution vision**	**Related requirements**	**Deadline**
1. Better service to users so that they know what happens and when.	Send a mail about the expected time for problem resolution, escalation, etc. Send a mail when the problem is resolved.	C1-2p, C2-1p, C10-4. C2-1p, C11-7q. C10-2q, C11-8p.	(none)
2. Avoid that requests are lost or delayed.	Generate reminders to the hotliners. Warn when nobody is on 1st line. Warn when transferring to someone off duty. Transfer requests when a supporter leaves.	C10-6, C11-9.	
3. Prevent problems.	Collect statistics about problem frequencies and causes. An existing, proven catalogue of causes.	C10-2p, C11-8q, C20, C21-2.	

B3. Early proof of concept

Some requirements are high-risk and the supplier may not be able to deliver what he promised in his proposal. If this is detected late in the project, the customer may terminate the contract, but this is a disaster to both parties. Usually the customer chooses to accept the inadequate system, possibly with compensation from the supplier. To reduce the risk, the customer requires an early proof of concept for the high-risk requirements.

According to the contract, both parties can terminate the contract if the early proof fails.

The following requirements are considered high-risk. Deficiencies here can hardly be rectified late in the project. In his reply, the supplier must state how he will carry out the proof of concept and when.

Areas where an early proof of concept is required:	**Example of proof:**	**Code:**
1. Efficient support of the request flow.	Customer and supplier walk through a number of situations in a hotline, assessing the supporter's part as well as the IT-user's. The customer expects to do this as part of the supplier selection.	
2. Usability (all requirements in section I1).	The customer expects to test usability within a few weeks after signing the contract.	
3. Cause catalogue.	The customer inspects the catalogue and tries to map some of his own examples into the catalogue.	

B4. Minimum requirements

Since the Z-department is a government organization, it must comply with the EU rules. This means that the suppliers must know the minimum requirements and the selection criteria before writing a proposal. This applies only to acquisition prices above a certain limit. We assume this example is above the limit.

Scores: The customer gives each proposal scores for the requirement areas shown in the table below. The table has space for several suppliers (columns A, B and C). The detailed requirements in Chapters C to I explain what the system provides today and where the problems are. Below, the customer states the scores he would give his present system.

The scores use this scale: -2 (not supported or very inconvenient), -1 (inconvenient), 0 (just sufficient), 1 (efficient), 2 (very efficient). Options get score 0 if they are not provided. We have shown an imagined example of scores for supplier A (in blue). Notice that although supplier A scores lower on F1 than the customer's present system, he still meets the minimum scores.

Minimum score: For each requirement area, the customer has stated the minimum scores below.

Minimum requirements: The system must meet the minimum scores in all requirement areas. Further, product cost+operating costs for 5 years must not exceed 80,000 $.

Requirement area	**Score for present system**	**Minimum score**	**Score**		
			Sup A	**Sup B**	**Sup C**
C1-C2. Service to users so they know what happens and when.	-1	0	1		
C10-C13. Avoid that requests are lost or delayed.	-1	0	1		
C10-C11 and C20-C21. Ease of cause registration and continuous improvement.	-2	-1	1.5		
D5. Initial cause catalogue.	-2	-1	2		
F1. Integration with AD (user list).	1	0	0		
I. Usability Windows user interface.	0	0	1		
Usability Mac/Mobile user interface.	-1	0	1		
K. Support of the acquisition process.	N/A	-2	1		
A3-1. Data conversion. Converting existing request data is an option. It gets score 0 if not offered.	N/A	0	1		

B5. Selection criteria: Most score points per dollar

Scores: The scores are those the customer assessed for the minimum criteria in B4. We have shown an imagined example of scores and weighted scores for supplier A (**in blue**).

Weight: Each requirement area has a weight that reflects the impact of the area. The weights add up to 100, excluding the options.

Requirement area	**Weight**	**Score**			**Weighted score**		
		Sup A	**Sup B**	**Sup C**	**Sup A**	**Sup B**	**Sup C**
C1-C2. Service to users so they know what happens and when.	20	**1**			**20**		
C10-C13. Avoid that requests are lost or delayed.	20	**1**			**20**		
C10-C11 and C20-C21. Ease of cause registration and continuous improvement.	20	**1.5**			**30**		
D5. Initial cause catalogue.	10	**2**			**20**		
F1. Integration with AD (employee list).	0	**0**			**0**		
I. Usability Windows user interface.	10	**1**			**10**		
Usability Mac/Mobile user interface	10	**1**			**10**		
K. Support of the acquisition process	10	**1**			**10**		
Total weight and total weighted score points excluding options	100				**120**		

For each proposal, the customer computes the total weighted score and the costs of deploying and operating the system for a period of 5 years, excluding options. Finally the score per 1000 $ is computed.

Score per 1000 $	**Sup A**	**Sup B**	**Sup C**
Total weighted score points excluding options	**120**		
Product cost	40		
Customer hardware costs	0		
Customer project cost	14		
Staff training	7		
Operating costs for 5 years	15		
Total costs for 5 years	**76**		
Basic score points per 1000 $	**1.58**		

For each option and each proposal, the customer computes the changes in total cost and in weighted scores. Next, he computes the new score points per 1000 $. If the result is higher than the basic score points per 1000 $, the customer accepts the option.

Option A3-1: Data conversion	**Weight**	**Sup A**	**Sup B**	**Sup C**
Cost change		**15**		
Weighted score change	10	**10**		
New score points per 1000 $		**1.43**		
Accept or reject option		**Reject**		

The customer selects the proposal and options with the highest score points per 1000 $.

C. Tasks to support

The system must support all user tasks in this chapter, including all subtasks and variants, and mitigate the problems. Column 1 of the tables describe what user and system will do together. Who does what depends on the chosen solution.

A task is carried out from start to end without essential interruptions. If necessary, the task must be parked and resumed later. Although subtasks are numbered, they don't have to be carried out in this sequence, and many of them are optional. The user decides what to do and in which sequence. A subtask may also be repeated during the same task.

Some subtasks may be performed in alternative ways. It is shown with a, b, etc. Letters p, q, etc. indicate something that today is a problem with this subtask.

Work area 1: The IT user

C1. Report a problem

Users: IT users. Around 1000 in total.
Start: The user has an IT problem, often urgent.
End: Hotline has received the request and maybe solved it.
Frequency: Around twice a year for the average IT user. Around 2000 a year in total.

Subtasks and variants:		Example solutions:
1.	Call hotline or meet in person.	
1a.	Send mail to hotline.	
2.	The problem may be solved on the spot.	
2p.	Problem: When not solved on the spot, when can I expect a reply?	The system reminds the supporter to set *reminderTime* and helps him send a mail about it to the user.

C2. Follow up on a problem

Users: IT users.
Start: The user is impatient or gets a mail from hotline.
End: The user knows the status of the request and may have provided more information.
Frequency: Unknown (part of the problem).

Subtasks and variants:		Example solutions:
1.	Check whether the problem has been solved.	
1p.	Problem: Hotline forgets to tell when the request is completed.	The system helps the supporter send a mail when hotline closes the request.
2.	See how far the request has come.	See hotline's web site.
3.	Provide more information.	
4.	Maybe close the request.	
5.	Maybe resume the request (open it again). This can be by mail, phone or in person.	

Work area 2: Hotline

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.

We might describe 1st and 2nd line with the same task because many subtasks are the same. But the two lines have different triggers and other details differ too. To avoid losing important details, we describe them separately.

C10. Handle a request in first line

Users: Hotline staff, sometimes also developers.
Start: This task can be started in many ways, for instance: A user calls or sends a mail; a reminder arrives because a request is overdue; or the supporter has done something else and now looks for the next request to handle. This is described below as variants of subtask 1.
End: The supporter cannot do more about the request right now.
Frequency: On average 10 per day for a supporter in hotline, sometimes 50 per day. Around 80% of the requests can be handled right away.

Subtasks and variants:		Example solutions:
1.	The user phones or meets in person. Record the request, particularly the user's phone or email. (See data details in section D1 and D2.)	The system warns if user or phone isn't specified when the request is created.
1p.	Problem: Cumbersome, particularly when it is an on-the-spot solution.	Fields allowed being blank as far as possible.
1a.	The request arrives by email through the support system.	The system records sender details, subject, etc. automatically.
1b.	The user provides new information about an open request. Find it and record new inf.	
1q.	Problem: Today it is difficult to add notes to an existing request.	
1c.	The user wants to reopen a closed request. Find it.	
1d.	A reminder arrives because a request has not been dealt with in due time.	The system sends the reminder.
1e.	The supporter has finished doing something else and now looks for a request to handle.	
1r.	Problem: In busy periods there may be 100 requests to handle and it is hard to spot the important and urgent ones.	Can restrict the list to relevant requests. Can sort according to reminder time, priority, etc.
2.	Maybe solve the problem on the spot and close the request.	The system warns if the cause hasn't been set.
2q.	Problem: The user isn't informed when the request is closed.	The system sends a mail when the request is closed. The supporter may write an explanation in the mail.
2p.	Problem: To gather statistics, a cause should be specified, but this is difficult and cumbersome today.	A list of frequent causes to choose from.
3.	If necessary, contact the source or sender to obtain more information.	The system shows the related office and phone number.
4.	Update request data, including adding notes, setting priority, etc.	
5.	Order something from an external supplier and park the request. Give it a priority and a reminder time. Inform the user.	The system warns if no reminder time has been set.
6.	Maybe transfer the request to second line or a supporter with special expertise. Give it a priority and a reminder time. Inform the user.	The system helps sending a mail about it to the user.
6p.	Problem: Hard to see which supporters are present now.	
6q.	Problem: Today it is done with a yellow sticker on the expert's desk, but it often disappears.	
7.	Maybe leave the request in the "in-tray" for later treatment.	The system warns if a reminder time hasn't been set.

C11. Handle a request in second line

Users: Usually more experienced supporters.
Start: The supporter gets an email about a request or looks for pending requests.
End: The supporter cannot do more about the request right now.
Frequency: Around 2 per day for the entire hotline. Half of them can be handled right away.

Subtasks and variants:		Example solutions:
1.	Look at open second-line requests from time to time, or receive an email.	
1p.	Problem: In busy periods it is hard to spot the important and urgent requests.	Can restrict the list to relevant requests. Can sort according to reminder time, priority, etc.
1a.	The user provides new information about an open request. Find it and record new inf.	
1q.	Problem: Today it is difficult to add notes to an existing request.	
2.	Maybe solve the problem by moving to the problem location.	The system shows the sender's office.
2p.	Problem: Would be nice if the supporter could handle several problems on his trip.	Mark some of them and print them. Or put them on a special list for mobile access.
3.	Work for some time on the problem, see its notes. Inform others that they don't have to look at it.	Put the request in state *taken*.
4.	If necessary, contact the source or sender to obtain more information.	The system shows the related office and phone number.
5.	Update request data, including adding notes, setting priority, etc.	
6.	Order something from an external supplier and park the request. Inform the user.	The system warns if no reminder time has been set.
7.	In case of a reminder, contact the supplier and set a new reminder time.	
7p.	Problem: The user doesn't know about the delay.	The system sends a mail when the reminder time is changed.
8.	Maybe close the case.	
8p.	Problem: The user isn't informed when the request is closed.	The system sends a mail when the request is closed. The supporter may write an explanation in the mail.
8q.	Problem: To gather statistics, a cause should be specified, but this is difficult and cumbersome today.	The system warns if the cause hasn't been set. A proven list of causes to choose from.
9.	Maybe transfer the request to a supporter with special expertise. Give it a priority and a reminder time. Inform the user.	The system helps sending a mail about it to the user.
9p.	Problem: Hard to see which supporters are present now.	
10.	Maybe leave the request in the "in-tray" for later treatment.	The system warns if a reminder time hasn't been set.

C12. Handle message from an external supplier

Users: Usually more experienced supporters.
Start: A message or a delivery is received.
End: The supporter cannot do more about the request right now.
Frequency: Rare. Maybe once a week.

Subtasks and variants:		Example solutions:
1.	Find the request.	
2.	(Continue as in C11)	

C13. Change role

Users: Any supporter.
Start: A supporter has to change line, leave the hotline, etc.
End: New role recorded.
Frequency: Around 6 times a day for each supporter.

Subtasks and variants:	Example solutions:
1. Change own settings for line, absence, etc. (see section D3).	
2. Change other supporter's settings, for instance if they are ill.	
3. When leaving for a longer period, transfer any taken requests to 1st or 2nd line.	
3p. Problem: The supporter forgets to transfer them.	The system warns and suggests changing status for all of them.
4. When leaving 1st line, check that enough 1st line supporters are left.	
4p. Often forgotten.	The system warns.

Work area 3: Management

C20. Study performance and statistics

Users: In principle any supporter.
Start: Occasionally.

Subtasks and variants:	Example solutions:
1. Get access to history data.	
2. Look at data in various ways. See E2.	

C21. Update basic data

Users: In principle any supporter.
Start: Occasionally.

Subtasks and variants:	Example solutions:
1. Update the supporter list.	
2. Update the catalogue of problem causes and cures.	

D. Data to record

The system must record the data described in this chapter. The user can create, view, and change the data through the tasks described in Chapter C. In some cases, data has to be exchanged with external systems as specified in Chapter F. Figure 3 illustrates the data in the existing system.

Figure 2 illustrates data in the new system. It is a data model (an Entity/Relationship diagram, E/R) that gives an overview of the data. Each box is a class of data. Imagine a pile of file cards behind the box (also called Records or Rows). The box symbolizes one of the cards. As an example, D3 is a pile that holds a card for each supporter. Next to each box is a list of the fields on the card (or just some of them).

There are relationships between the boxes, shown as crow's feet. A crow's foot shows that a card relates to one or many cards in another pile. As an example, an employee can be the sender of many requests, but a request has only one sender.

As another example, a request step may have zero or one cause. The small circle indicates that zero is okay. Reading the crow’s foot the other way, a cause may be the cause of many request steps.

Data need not be structured this way in the system, but it must be handled in some way.

Dotted boxes show data that are (partly) shared with an external system.

D1. Request: Each record contains basic data about a request: the ID that supporters use, the time the request was created, and the subject the IT-user stated.

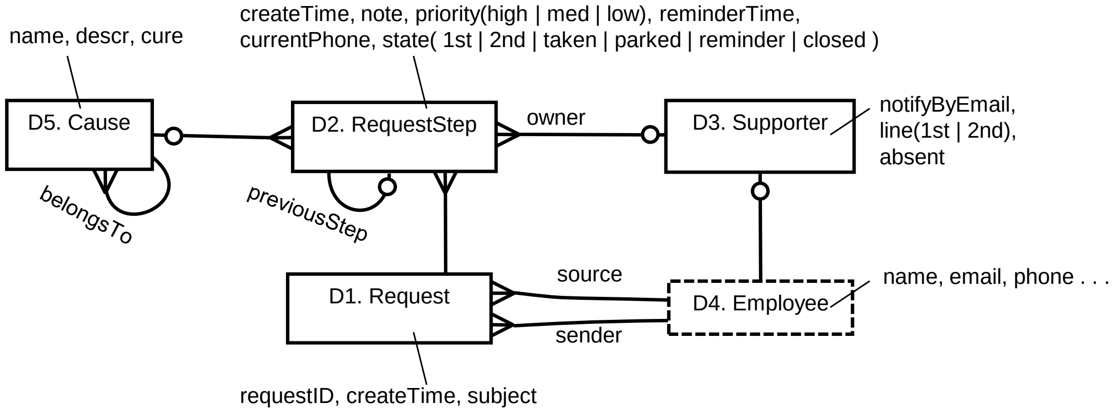
D2. RequestStep: A request has one or many request steps, which keep track of the history. The last step shows the current state of the request. Each step has a note written by the person who created the step, the current priority of the request, the time when the system should send a reminder, a state of the request (either 1st, 2nd, taken, etc.), etc. When handling a request, the supporter will often look at the entire history, for instance all the notes written on the way.

D3. Supporter: Each record contains data about a supporter, whether he currently is 1st or 2nd line, and a relation to his employee record which holds his name, etc.

D4. Employee: Each record holds data about a person: name, email, etc. The table is shared with the customer's employee list (Active Directory, AD, see F1).

D5. Cause: Each record holds data about a potential request cause: Its name (e.g. “password forgotten”, “printer out of toner”), a description, and maybe a cure for the problem. There may be a hierarchical structure in the causes.

Figure 2. Data model for hotline



D0. Common fields

Each data class records history, i.e. each change creates a new version of the "file card" and preserves the old one. It is recorded in these fields.

Fields and relationships:	Example solutions:	Code:
1. Change Time: The date and time when the "file card" was created or changed.		
2. Source: The person who created or changed the "file card".		
3. History: Relation to earlier versions of the "file card" (not shown in the diagram).		

D1. Request

A request is a problem report sent by the IT user or recorded by a supporter on behalf of a user. The request is recorded as a Request record and a RequestStep. Whenever somebody deals with the request, the system records it as a new RequestStep.

Fields and relationships:	Example solutions:
1. requestID: The identification of the request.	Generated by the system.
2. createTime: The point in time where the request step was recorded.	Generated by the system.
3. subject: The title given by the IT user or set by a supporter.	

D2. RequestStep

Whenever somebody deals with the request, the system records it as a new RequestStep.

Fields and relationships:	Example solutions:
1. createTime: The point in time where the request step was recorded.	Generated by the system.
2. note: An explanatory text written by the person who created the request step.	
3. priority: The supporter's estimate of how urgent the request is.	
4. reminderTime: The point in time when the system will set the request in reminder state.	The system offers some default times.
5. currentPhone: If the sender works away from his usual place, we may record his current phone.	
6. state: The current state of the request.	See solution notes below.
7. cause: A relationship to what the supporter currently believes is the cause.	
8. **o**wner: A relationship to the supporter who works on the request. Null if nobody works on it.	Generated by the system.
9. previousStep: A relation to the previous request step (if any).	Generated by the system.

Solution notes
The system might support these 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 taken or 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.

D3. Supporter

Data about the current role of a supporter.

Fields and relationships:	Example solutions:
1. notifyByEmail: Whether the system sends an email to the supporter when there is a request for him.	
2. line(1st | 2nd): Whether the supporter currently works in 1st or 2nd line.	
3. absent: Whether the supporter is away, for instance for lunch or on a course.	
4. employee: A relation to the supporter's employee record.	

D4. Employee

Data about an employee.

Fields and relationships:	Example solutions:
1. name, email, phone, office.	The customer's employee data (e.g. from Active Directory, AD). See F1.

D5. Cause

The cause records may make up a hierarchical list of possible request causes. It is a maintenance task to update this list according to changing needs.

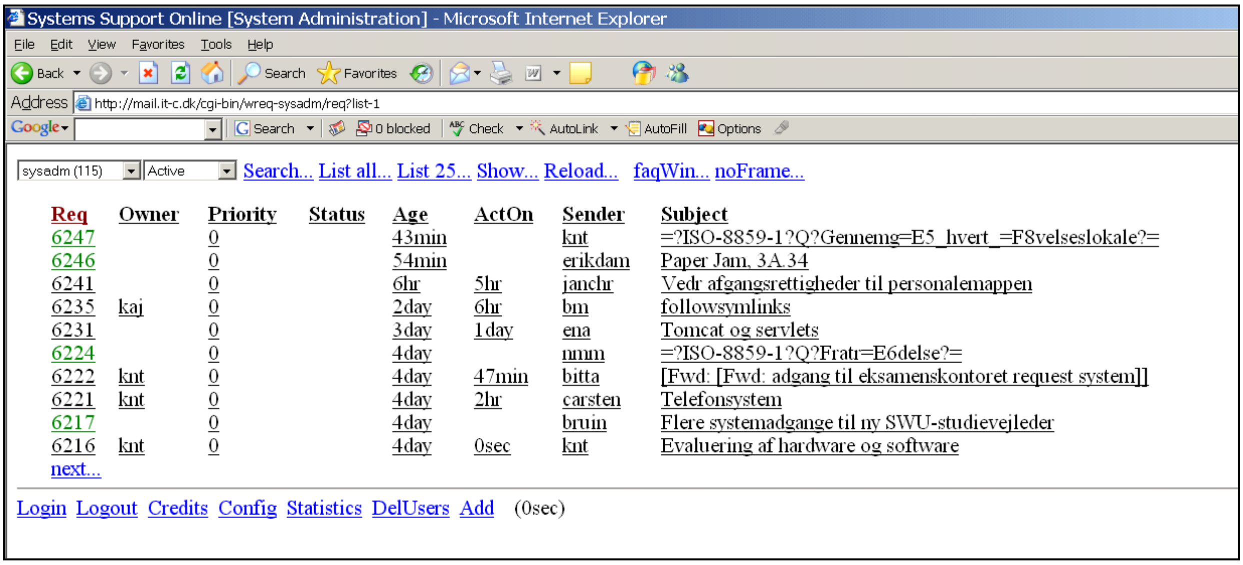
Fields and relationships:	Example solutions:
1. name: A short name of the cause, e.g. "Password lost".	
2. descr: A longer explanation of the cause.	
2p. It is difficult to create a suitable list.	The system provides a proven list as a starting point
3. cure: Ways to deal with the problem, e.g. some URL's.	
4. belongsTo: A relationship to a higher-level cause in the hierarchy.	

D6. CREDO check

This section is not requirements, but a check that tasks and data match and nothing is missing. CREDO means Create, Read, Edit, Delete and Overview. Each row (record) in a table must in principle through a task be created, read, edited, deleted and seen in some overview.

CREDO check	D1. Request	D2. Step	D3. Supporter	D4. Empl.	D5. Cause
C1. Report problem	C---O	C----	-----	-R---	-----
C2. Follow up					
C10. Handle 1st line					
C11. Handle 2nd line					
C12. Handle supplier					
C11. Change role					
C20. Study performance					
C21. Update basic data					
Missing					

Comments:



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

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?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 3. A list of hotline requests in the existing support system.

E. Other functional requirements

Most system functions are simple creations, deletions, edits, and queries that need no further specification. They are implicitly given by the task descriptions (Chapter C), system integrations (Chapter F), etc. In addition, the system must be able to perform the functions specified in this chapter.

E1. System generated events

The system must generate these reminders:	Example solutions:	Code:
1. When reminderTime for a request is passed, the request must appear as a reminder to supporters.	The request appears as a reminder to be handled by first line, which always should be staffed.	

E2. Reports

Some reports are needed in connection with the tasks described in Chapter C. The report formats are not essential as long as the tasks are supported well. These reports are not described in this chapter. There is also a need for reports with ad hoc purposes. They are specified here.

Report requirements:	Example solutions:	Code:
1. The system has a report generator that is easy to use for supporters. It can combine data from all classes in the data model.		
2. The system can show all reports on the screen, in print, and store them as a file.		
3. Supporters can explore all data in the data model in an ad-hoc way.	The system can transfer data to a spreadsheet or provide direct access to its own database.	

E3. Business rules and complex calculations

Not used.

E4. Expansion of the system

Not used.

F. Integration with external systems

The system must integrate with the systems mentioned below. The box headers D, G, K, etc. are part of the headers defined in SL-07

F1. AD (employees)

E-data (external data): The customer's Active Directory (Microsoft AD).

Tasks:	A supporter may need the IT user's name, email address, phone and room number when handling a request (C10 and C11).
E-support:	The customer can provide AD details.
E-data updates:	The customer updates AD on a monthly basis and sometimes ad-hoc.
S-data updates:	The hotline system is not expected to update anything in AD.
Data volume:	Around 1000.

D. Integration responsibility:	Example solutions:	Code:
1. The supplier must integrate with AD.	Direct .NET access to the AD or daily copying of the relevant data.	

G. Data recency:	Example solutions:	Code:
2. The employee data should not be older than 24 hours.		

F2. Integration with new external systems

The customer expects that he can integrate new external systems with the hotline system - with little or no help from the supplier. Integration means that the external system can report requests to the hotline system.

External system:	In principle any system. Examples: A system that monitors IT-operations, a system that keeps track of employer's loans of equipment and tells hotline when a loan is overdue.
Data volume:	Less than 100 requests per day.

D. Integration responsibility:	Example solutions:	Code:
1. The customer is responsible for the integration. The supplier assists on the same conditions as maintenance.		

K. Other functions:	Example solutions:	Code:
2. The new system can report requests to the hotline system.		

G. Technical IT architecture

G1. The customer operates the system

The customer intends to operate the system and buy new equipment as needed.

Platform requirements:	Example solutions:	Code:
1. In order to meet the requirements in L1, L2 and L3, the customer may need new IT equipment.	The customer needs this equipment _____.	
2. The supporters need to use the system from Windows, Mac and mobile devices.		

H. Security

H1. Login and access rights for users

Login is not a separate user task, but subtasks that occur in many tasks. The system must support the following subtasks relating to the user's access rights.

Subtasks for user access:	Example solutions:	Code:
1. Identify the user with the existing user identification, login method, and time-out method, which is AD.		
2. Check that only authorized users get access to systems and data. (See the requirement note below.)		

Requirement note: Possible access rights

1. Right to access hotline requests. All supporters have this right.

H2. Security management

Supporters can create and remove other supporters according to task C21. This seems sufficient for security management.

H3. Protection against data loss

Since the customer intends to operate the system, his existing processes take care of basic data protection. See H6 for other threats.

H4. Protection against unintended user actions

An unintended user action means that the user happened to do something he didn't intend to do, e.g. hitting the wrong key or using a command that does something he didn't expect.

Requirements:	Example solutions:	Code:
1. Unintended user actions may not cause the system to close down, neither on the client nor on the server.	May be hard to test at delivery, but the supplier's issue log and a description of the supplier's test methods may provide evidence.	
2. All data entered must be checked for format, consistency and validity. In case of doubt, the user must be warned and asked what to do.		
3. The user must be able to correct mistakes easily.	The system provides extensive use of undo.	
4. Prevent mistaken use of undo-able functions.	Position the button so that it is not hit accidentally - or ask for confirmation.	
5. The user must be able to interrupt functions that take a long time, e.g. a long data transfer, without compromising data integrity.	The customer doesn't expect any operations to take a long time.	

H5. Privacy requirements

The customer must meet the European Union’s privacy rules (GDPR, General Data Protection Regulation).

Requirements:	Example solutions:	Code:
1. The system provides functionality that enables the customer to meet GDPR, e.g. deletion of an IT-user's support requests at the user's request.		
2. In case of a GDPR dispute, the customer must be able to document which personal data the system uses and for what.	The supplier provides the necessary documentation.	

H6. Protection against threats

Since the customer intends to operate the system, his existing processes take care of most threats. However, the hotline system needs to protect against some other threats.

The system must protect against:	Example solutions:	Code:
1. SQL injection (the intruder types a database command where the system expects e.g. a person name; as a result the system carries out the database command).		
2. DoS attack (Denial of Service). An attacker sends so many requests to the system that it is paralyzed.		
3. Other threats the supplier knows of.		

I. Usability and design

I1. Ease-of-learning and task efficiency

It is important that the system has adequate usability. The customer will test this as part of the early proof (POC).

Requirements for early proof of concept (POC):	Example solutions:	Code:
1. After a short course (see J2), the supporters must be able carry out the tasks in work area 2 and 3 without critical usability problems (see definition in the requirement note below).	Usability testing (think-aloud testing) is carried out. The supplier may improve the user interface and test again. Two new supporters participate in each round of testing.	
2. They must be able to do so from their preferred Mac, Windows and mobile.		

Although the system has passed the test, it may turn out to give the users considerable trouble in areas that were not tested. The customer wants to avoid the situation where the supplier rejects the problem with reference to the customer having approved the system, or it being a COTS system.

Requirements for handling usability problems later:	Example solutions:	Code:
3. Critical usability problems (see definition in the requirement note below) must be handled as system errors in the same way as other errors in the system.	The error is handled by the support organization and eventually transferred to maintenance.	

Requirement note: Serious and critical usability problem
A **serious** usability problem is a situation where the user:
a. is unable to complete the task on his own,
b. or believes it is completed when it is not,
c. or complains that *it is really cumbersome*,
d. or experts observe that the user doesn't use the system efficiently.

A **critical** usability problem is a serious usability problem that is observed for more than one user.

I2. Accessibility and Look-and-Feel

The Z-department has a policy of using English in written communication.

Requirements:	Example solutions:	Code:
1. The user interface must be in English.		
2. The user interface must show Danish letters correctly.		

J. Other requirements and deliverables

J1. Other standards to obey

None.

J2. User training

The customer wants to deliver most of the training himself. The idea is to train two supporters first, and let them train others.

Requirements:	Example solutions:	Code:
1. The supplier must train 2 supporters, making them able to train other supporters. The training must cover all tasks in work area 2 and 3.	Training takes __ hours. (The customer expects 3 hours).	
2. The supplier must train 2 supporters, making them able to handle the customer's part of system operation and support.	Training takes __ hours. (The customer expects 3 hours).	
3. The training must be carried out within the last month before system delivery in order that users can use the system immediately and haven't forgotten what they learned. If necessary, the training must be repeated and the delivery suspended.		

J3. Documentation

The customer expects that hot-line staff don’t need user documentation. However, documentation might be necessary for operation and for exploring the database.

Requirements:	Example solutions:	Code:
1. Before system delivery, sufficient documentation must be available for the customer to handle his part of IT operation and support.		
2. In order to explore request data, the customer needs documentation of tables and fields.		

J4. Data conversion

Data conversion is an option. If offered, the supplier converts the following data:	Example solutions:	Code:
1. Data about open requests in the existing system in such a way that they can be handled by the new system.		

J5. Installation

Requirements:	Example solutions:	Code:
1. The supplier must install all parts of the delivery, software as well as data.		
2. The supplier must install all converted data.		

J6. Testing the system

Requirements for the customer's own testing:	Example solutions:	Code:
1. The customer needs to test the system before accepting the delivery.	The supplier makes a test version available to the customer.	
2. Special situations must be tested.	The customer can insert special test data.	
3. There is also a need for testing with realistic data.	The supplier converts parts of the customer's existing data and inserts them in the test version.	

J7. Phasing out

In this section "customer" means the customer's own staff or third party authorized by the customer.

Requirements:	Example solutions:	Code:
1. On request, the supplier must extract all data described in Chapter D in a format that is suited for import in other systems.		
2. The customer must be able to extract all data described in Chapter D in a format that is suited for import in other systems.		
3. At phasing out, the supplier must provide or update the descriptions of all tables and fields, cf. J3-2.		
4. The supplier must loyally assist with phasing out the system and transferring data to another supplier.		
5. The supplier must carry out the work at a fair price that covers time and material.		

K. The acquisition process

K1. Acquisition plan

The customer is willing to follow the supplier's recommendations on the acquisition process, but wants to ensure that the essentials are covered. The customer imagines the activities below during the process.

The supplier is asked to comment on the plan and/or state the plan he proposes, based on his own experiences. He is asked to state the expected end time as the number of work days after signing the contract. For test activities, it is when the customer has had reasonable time to check and approve the test results. As in the rest of the requirements, everything written by the supplier must be in red.

Activities and participants:		Users	Customer's IT and management	Supplier	External parties	Workdays after signing contract
1.	Market screening and prequalification.	v	v	v		N/A
2.	Introductory supplier meetings (see requirement note below).	v	v	v		N/A
3.	Revise requirements and send for tender.		v			-70
4.	Write and send proposal.		(v)	v		-30
5.	Supplier selection.	v	v	v		-10
6.	Signing the contract.		v	v		0
7.	POC (Proof-Of-Concept, see B3).	v	v	v		10
8.	Option: Data conversion (see J4).		v	v		
9.	System test (see req. note below).		v	v		
10.	Pilot test.	v	v	v		
11.	Operational test (see Chapter L and req. note below).		v	v		
12.	Evaluate the business results.	v	v			
13.	Warranty period (one year, see req. note below)		v	v		
14.	Support and maintenance (see Chapter L).		v	v		

Requirement notes

Introductory supplier meetings: The customer sends the requirements to selected suppliers and have an informal meeting with each of them. At the meeting, the supplier should show how his system can meet the requirements. The supplier is not expected to deliver anything in writing. The customer welcomes comments on the requirements, e.g. unsuitable or missing requirements. The customer stresses that the supplier not presents his own solution only, but shows how it can be used for the customer's purpose.

Supplier selection: When the customer compares the proposals, there is a need to see the system in operation, get answers to questions, talk to references, etc. The supplier is expected to support this.

System test: The purpose of the system test is to check that requirements are met, screens work correctly, etc. Special test data and database contents are used to allow testing all the special situations (see also J6).

Pilot test: A pilot test is real work being done. The customer wants the supplier's experts to observe the users and see whether they use the system as planned.

Acceptance test: The customer hasn't indicated an acceptance test in the plan. Acceptance consists of an approved system test and an approved pilot test.

Operational test: Starts after acceptance of the full system. The purpose of the operational test is to check those requirements that can be verified only after a period of daily operation. It might be the response time under real load, availability (breakdown frequency), user's task time, the supplier's hotline quality, etc. See the requirements in Chapter L.

Warranty period: The warranty period starts when the first (partial) delivery is approved and ends one year after delivery of the entire system. The supplier must remedy all significant defects detected in the warranty period. After the warranty period, defect correction is covered by the maintenance requirements (L5).

K2. Project management

During the acquisition process, the customer's project management must at all times know how far the process is, what is missing, and maybe where we must change something. It is important for the customer to get good support for this, but often the supplier's reports are insufficient or obscure. We can describe project management's needs with this task:

Users: Customer's project managers and members of the steering committee.
Start: When project management need an overview of the project status, e.g. before and during meetings with the supplier, project team meetings, or steering committee meetings.
End: When there is nothing more to do right now.
Frequency: Weekly or monthly.

Subtasks and variants:		Example solution:	Code:
1.	How far are we in the schedule? What is missing?	The supplier maintains a Gantt diagram that the customer can access.	
2.	How many hours has the supplier spent on each activity and how many are needed to complete it?	The supplier has a system that shows estimated hours per activity, how much is spent and how much is still needed. The customer has access to this system.	
3.	Which visible results do we have, e.g. user screens, integrations that work, test reports?	Each activity is terminated with a result that is visible to the customer.	
4.	How much do the test cases cover?	See J6.	
5.	Which open issues do we have and what happens to them?	The system can show overviews of the issues and what has changed since a specific date.	
5p.	**Problem:** The supplier uses an issue tracking system with poor overview for the customer, so he has to make his own.	See the supplier's system and assess how well it supports the customer.	
6.	Can we approve the tests? Can we deploy the system, maybe as a partial delivery? (See the requirement note below).		
7.	In case of serious troubles: What can we do? Re-plan? Change staffing? Change organization and decision authority? Get help from outside? Increase funding? Change scope? Close the project?		
8.	Is the business case still valid?		
9.	What are the most important risks? The probability? The consequence if it happens anyway?	Joint risk analysis for customer and supplier.	

Requirement notes

Approve the tests and deploy the system

The customer may reject a test although no serious issues were found. It may for instance happen if the test doesn't sufficiently cover all situations that may occur.

However, the parties may define a partial delivery that the customer can approve and deploy. As an example, some user groups may use the system, while others have to wait. In this case, the payment for the partial delivery is reduced accordingly.

K3. Update issue list

Even small projects may have more than hundred issues, requests for change as well as things not working as expected. The parties can spend lots of time discussing what to do with them, and it delays the development process. The customer wants efficient management of the issues. What to do now, what can wait, what to ignore, who pays? This can be handled by recording the issues when they turn up, and maybe not deal with them until later. We can describe it as this task:

Users: The customer's project team and the supplier's developers.
Start: When an issue turns up and it cannot be dealt with immediately. Or when issues change state (e.g. after test). Or at meetings between supplier and customer, where the parties review the list of open issues.
End: When nothing more needs to be done with the issues right now.
Frequency: Weekly. In the test periods often daily.

Subtasks and variants:		Example solution:	Code:
1.	New issues: Record the problem with date, source, a short description, maybe screen dumps, etc. Often you don't have to do more right now and can park the issue.	See the supplier's system, e.g. in an appendix with screen shots, and assess how well it supports the customer.	
1a.	Find a specific issue and update its data.		
1b.	Find the next issue to deal with.		
2.	Is it duplicate, i.e. something we have recorded already?		
3.	Analyze the issue. How important for the customer? How expensive to rectify? How urgent? Is there a work-around for the customer?		
4.	Is it a defect (the supplier has to rectify it) or a new requirement (the customer has to pay)? This is sometimes hard to determine (see the requirement note below). For minor changes, you can make the change and postpone the pay decision until later.		
5.	What is the consequence of the change on price, delivery time, documentation and maintenance?		
6.	Maybe reject the issue or postpone it with a deadline.		
7.	Maybe add your own notes.		
8.	When the change has to be made: Sometimes it is necessary to change the requirements specification, but often it is a matter of details that are not mentioned in the requirements. If so, it is sufficient to record the decision in the issue tracking system.		
9.	For changes to be made: Is it done? Tested? Deployed?		
10.	Maybe inform the parties.		

Requirement notes
Defect or request for change?

a. When a programmer can see that the system doesn't work as intended, it is a defect and the supplier covers the cost. Most issues are of this kind.
b. When the system doesn't meet the requirements or the proposed solution description, it is also a defect and the supplier covers the costs.
c. When the system can do what the user wants, but the user cannot figure out how, it is a usability issue. Whether to rectify it and who pays, depends on the requirements in I1.
d. If the system doesn't meet the customer's reasonable expectations, it is also a defect. A reasonable expectation means that the supplier knew or should have known that this would become a problem for the customer.
e. In other cases it is a request for change and the customer has to pay.

K4. Work place and the customer's deliverables

Work place:	Example solution:	Code:
1. Physical meetings between customer staff and supplier staff improve the process.	The supplier's staff work in the customer's offices.	

The following list of the customer's deliverables and services must be complete. The supplier cannot expect more from the customer. If necessary, the supplier must add to the list in his proposal.

The customer delivers:	Supplier comments:	Code:
2. Hardware, software, and external systems that the new system requires (see the details in Chapter G). The equipment must be available when testing starts.		N/A
3. Office with two IT work places from one month before the planned system test to one month after system delivery.		N/A
4. Samples of production data for testing purposes and the full data set in case of conversion.		N/A
5. Test cases for system testing.		N/A
6. Expertise in the application area corresponding to a half-time employee during the entire project.		N/A
7. Test subjects for usability tests.		N/A
8. A half-time project manager.		N/A
9. IT staff who will support the system at the customer site.		N/A

L. Operation, support, and maintenance

This chapter specifies the supplier's responsibilities after delivery of the system itself. The requirements can only partly be verified (tested) at the pilot test. The full verification takes place later, at the operational test.

L1. Response times

The system must be able handle the peak load specified below, with the specified response times.

Peak load
The peak load is that three supporters work concurrently with 100 requests that they try to resolve. One of them work from a PC, one from Mac and one from a mobile phone.

Solution note: Measuring response time
The response time is the period from the user sends his command to the result is visible and the user can send a new command. A command means a key press, a mouse click or a tap. All measurements are made in a peak load period.

Measurements are made with a setup according to Chapter G.

Response time requirements:	Example solutions:	Code:
1. **Fractile.** The times specified below must apply in almost all cases.	__% of the response times must be within the limits. (The customer expects 98%.)	
2. When moving from one field to the next, the user's typing speed must not be reduced.	Typing is possible within ___ s. (The customer expects 0.5 s.)	
3. When moving from one screen to the next, data must be visible and typing possible within the mental switching time (around 1.3 s).	Data is visible and typing possible within ___ s. (The customer expects 1.3 s.)	
4. Lookup in drop-down lists must allow selection from the list within the mental switching time.	Selection is possible within ___ s. (The customer expects 1.3 s.)	
5. Reports used occasionally must be visible before the user loses patience.	The report must be visible within ___ s. (The customer expects 10 s.)	
6. Login must be completed before the user loses patience.	The user can start working within ___ s. in addition to the time he spends typing name and password. (The customer expects 10 s.)	
7. Repeated login when the user temporarily has left the system must be completed before the user loses patience.	The user can start working within ___ s in addition to the time he spends typing his password. (The customer expects 4 s.)	

L2. Availability

The system is out of operation when it doesn't support some of the users as usual. The cause of the breakdown may be:

1. The customer's issues, e.g. errors in the customer's equipment.
2. External errors, e.g. power failure.
3. The supplier's issues, e.g. errors in software or configuration.
4. Planned maintenance.
5. Insufficient hardware capacity.

In this project, the customer wants to operate the system himself. As a result the supplier is responsible for only point 3 above.

Solution note: Measuring availability
A breakdown is counted as at least 20 minutes, even if normal operation is resumed before. If the following period of normal operation is less than 60 minutes, it is considered part of the breakdown period.

The **operational time** in a period is calculated as the total length of the period minus the total length of the breakdowns for which the supplier is responsible. The **availability** is calculated as the operational time divided by the total length of the period. When only some of the users experience a breakdown, the availability may be adjusted. One way is to calculate the availability for each user and take the average for all users.

Availability requirements:	Example solutions:	Code:
1. The availability must be calculated periodically. The calculation should compensate for the number of users experiencing breakdowns.	The availability is stated monthly and calculated as described above.	
2. In the period from 8:00 to 18:00 on weekdays, the system must have high availability.	In these periods the total availability is at least ___%. (The customer expects 99.5%)	
3. In other periods, the availability may be lower.	In these periods the total availability is at least ___%. (The customer expects 99%)	

L3. Data storage

The data volume is specified in Chapter D. Data must be stored as follows:

Data storage requirements:	Example solutions:	Code:
1. The system must give access to data for the last 5 years.		

L4. Support

In this section, “user” means one of the customer’s hotline staff. "Supplier" means the supplier's organization. A "supporter" means a qualified supplier employee. The support covers all hardware and software delivered under this contract.

Support requirements:	Example solutions:	Code:
1. **Fractile.** The response times specified below must apply in almost all cases.	__% of the response times must be within the limits. (The customer expects 95%.)	
2. The supplier must handle user requests for help. See the requirement note below.		
2p. Problem: Users find it hard to decide which product a specific problem relates to. It is even harder to mediate between several suppliers.	The supplier involves the necessary other parties on his own initiative.	
3. Direct contact: In the period from 8:00 to 18:00 on weekdays, users can quickly contact a supporter by phone or mail.	In this period, contact is available within __ minutes. (The customer expects 10 minutes.)	
4. For a direct contact, the supporter handles the request on the spot as far as possible.	On the spot means what can be done within 5 minutes.	
5. Indirect contact: Requests sent by email, sent by web, or escalated from the direct contact. The user gets a reply within a few hours.	The supplier replies within __ work hours (8:00 to 18:00 on weekdays). (The customer expects 3 hours.)	
6. The supplier monitors request handling to see that requests are closed and response times met.		

Requirement note: Handle a request

When a supporter receives a request, he can perform one or more of the following subtasks. All subtasks except e (escalation) end with a **reply** to the user. The request is **closed** when nothing more can be done about the request (subtask f).

a. Help user: Assist the user in solving the problem or circumventing it. If needed contact the user for clarification. Assistance is considered a valid reply.

b. Change configuration: E.g. start servers, change settings, replace printer cartridges, install software. Reply to the user when it has been done.

c. Order equipment or help from another organization: Reply to the user about the expected delay.

d. Defect: The support organization cannot solve the problem. Report it to the maintenance organization. Reply to the user that it has been done.

e. Escalate request: The supporter cannot fully solve the problem himself. Pass the request on to another supporter. This person may again perform one or more of the subtasks.

f. Close the request: Nothing more can be done about the request. This may happen at the first point of contact. The request may also escalate several times, wait for external delivery or wait for a reply from maintenance before it can be closed. Reply to the user that the request has been closed.

L5. Maintenance

Maintenance includes defect removal and system updates.

Requirements for defect removal:	Example solutions:	Code:
1. **Fractile.** The response times specified below must apply in almost all cases.	__% of the response times must be within the limits. (The customer expects 95%.)	
2. The supplier keeps a log of reported defects.		
3. For all reported defects, the customer's supporters decide whether the defect is business critical. The supplier assesses whether it is possible to circumvent the defect temporarily, or circumvent it permanently (i.e. reject it).	In the period from 8:00 to 18:00 on weekdays, the supplier completes the assessment within __ hours. (The customer expects 3 hours.)	
4. Business-critical issues are rectified quickly.	Business-critical issues are rectified within __hours. (The customer expects 24 hours.)	

Requirements for system update:	Example solutions:	Code:
6. When third-party software on which the system depends, is changed, the supplier must if needed, adapt and install new versions of the system without unduly delay.	Installation takes place within ___ days after release of the third-party software in Denmark. (The customer expects 30 days.)	