**Problem-oriented EHR requirements SL-07 version 9**

IT developers and IT consultants often ask for an exemplary requirements specifica­tion as a starting point in their own project. This document is such a specifica­tion. It is a template filled out with a complex example: *Requirements for an Elec­tronic Health Record system (EHR)*. Only a few points had to be illustrated with examples from other areas. Large parts of the specification may be re­used in other projects. Parts that are too special for reuse are shown in blue.

All the requirements are written in tables. Column 1 is the customer's *demands*. Column 2 is his *example solution* and later the supplier's *proposed* *solution*.

There is a **contract** that matches the requirements. Requirements template, contract, requirements and supplier’s proposal for real projects are available here:

<http://www.itu.dk/people/slauesen/SorenReqs.html#SL-07>

You may freely use the template as long as **the source and copyright notice are clearly stated**, for instance as in the footer of the next page.

Copy this document, delete the first page, and adjust the rest to your project. Blue parts should always be replaced with something else - or deleted. Parts with yellow background are warnings or alternatives. Other parts may often be reused.

The guide to the template and the contract are published as a booklet:

Soren Lauesen: Problem-oriented requirements SL-07 – Guide and contract.

It is available on Amazon and on the author's web site:

<http://www.itu.dk/people/slauesen/SorenReqs.html#SL-07>

The guide comments the template page by page, explains why the requirements are stated as they are, what to avoid, how to elicit and test the requirements, and how it relates to the contract.

I wrote large parts of the template and the guide on request from the Danish Min­istry of Research, Technology and Development in 2007 (hence the name SL-07), as part of their new standard contract for software acquisition (K02).

Earlier versions of the template have been used with success in 150+ very different projects. EU tender processes as well as in-house projects, agile as well as waterfall, e.g. home care in a municipality, including route optimization; a pharmaceutical company's innovative document management system; electronic health records; stock management for movie production; claims management for car insurance with GIS as documentation.

Experiences from these 150+ projects helped me write this version of the template. See the change log on page 3.

I have experienced that SL-07 works extremely well in practice - once you have learned how to use it. Although it looks easy, most people get it all wrong the first time, particularly the tasks in Chapter C. With a bit of help they get it right.

Any comments - positive as well as negative - are most welcome and will help me improve future versions.

Soren Lauesen

The IT-University of Copenhagen, February 2022 [slauesen@itu.dk](mailto:slauesen@itu.dk) [http://www.itu.dk/people/slauesen](%20http://www.itu.dk/people/slauesen)

Version 9 24-02-2022, 15:54

Last changed by: slauesen

**Requirements specification for**

Electronic Health Record System

(below called the EHR system)

**Customer**

Midland Hospital

**Supplier**

…

**The delivery comprises**

Software, operation and maintenance for an EHR system

**Contents**

A. Background and supplier guide 4

A1. Background and vision 4

A2. Supplier guide 5

A3. Customer options 6

A4. Overall solution 6

A5. Supplier options 6

B. High-level demands 7

B1. Patient treatment (future flow) 7

B2. Business goals 8

B3. Early proof of concept 9

B4. Minimum requirements 10

B5. Selection criteria: Highest net benefit 11

B6. Selection criteria: Most score points per dollar 12

C. Tasks to support 13

Work area 1: Patient management 13

C1. Admit patient before arrival 13

C2. Admit immediately 13

Work area 2: Patient treatment 14

C10. Perform clinical session 14

C11. Order medicine for the patient (long subtask) 15

C20. Perform clinical session, mobile 15

D. Data to record 16

D0. Common fields 17

D1. Diagnosis 17

D2. Diagnosis Type 18

D3. Service 18

D4. CREDO check 20

E. Other functional requirements 21

E1. System generated events 21

E2. Reports 21

E3. Business rules and complex calculations 22

E4. Expansion of the system 23

F. Integration with external systems 24

F0. Common integration requirements 25

F1. SKS 26

F2. LabSys 27

F10. Integration with new external systems 28

G. Technical IT architecture 29

G1. Existing hardware and software Alternative 1: Use what we have 29

G2. New hardware and software Alternative 2: Supplier suggests 29

G3. The supplier operates the system Alternative 3: Supplier's problem 29

H. Security 30

H1. Login and access rights for users 30

H2. Security management 31

H3. Protection against data loss 32

H4. Protection against unintended user actions 32

H5. Privacy requirements 33

H6. Protection against threats 34

I. Usability and design 35

I1. Ease-of-learning and task efficiency 35

I2. Accessibility and Look-and-Feel 36

J. Other requirements and deliverables 37

J1. Other standards to obey 37

J2. User training 37

J3. Documentation 38

J4. Data conversion 38

J5. Installation 38

J6. Testing the system 39

J7. Phasing out 39

K. The acquisition process 40

K1. Acquisition plan 40

K2. Project management 42

K3. Update issue list 43

K4. Workplace and the customer's deliverables 44

L. Operation, support, and maintenance 45

L1. Response times 45

L2. Availability 46

L3. Data storage 46

L4. Support 47

L5. Maintenance 48

#### Change log (replace with change log for your own project)

**Version 1**: 22-06-2007. Pre-release version. Very close to the Danish version.

**Version 5.0**: August 2012. Changes:

1. Improvements of the selection method, integration requirements and usability requirements.
2. Many minor improvements all over, for instance to make the requirements better follow the author's own advices.

**Version 5.1**: October 2012. Changes:

1. Further improvements to the selection method, integration requirements, E/R explanation, and response times.
2. Several examples in column 2 have been moved to column 1 (they were not solution examples).

**Version 5.3**: November 2013. Changes:

1. New security requirement (H5-7; from version 5.5: H6-7).

**Version 5.4**: January 2016.

1. The template is now in docx format only.
2. The supplier guide is now only one page (A2).
3. B1 is a new section that describes the flows to be supported, and how they relate to the tasks in Chapter C. Business goals have become section B2.
4. Minimum requirements are changed in order to match legal practice better.
5. A short version of integration requirements are shown for F1. Many improvements in all of Chapter F.
6. Browser requirements are included in the architectures.
7. New requirements for stolen passwords (H1-4) and using an existing security management (H2).
8. New requirement for monitoring the operations management (H3-5p).
9. New requirement for protection against DoS attacks (H5-4; from version 5: H6-4).
10. Alternative requirements for usability (I1, Alternative 1).
11. Requirements for the customer's testing (J6) and phasing out the system (J7).

**Version 5.5**: November 2017.

1. The use of colors has changed: Red is used for the supplier’s proposal only. Warnings etc. have yellow background.
2. New section A3 added for the supplier’s alternatives and large-scale solution.
3. New section H5 deals with privacy requirements, particularly the European Union’s GDPR.
4. E2 adds “exploring data” as a report option.

**Version 6.0**: December 2018.

1. Contract SL-07 is published. It is 8 pages with 4 appendices. Appendix 1 is the requirements spec.
2. Section A3 now contains customer options as well as the supplier’s suggested options. The supplier guide is changed accordingly.
3. Chapter K now contains time schedule and requirements to the development process, issues traditionally contained in the contract.
4. Many changes to requirements for security, usability, response time, maintenance, etc.

**Version 7.0**: May 2019.

1. B4 (Minimum requirements) now shows points for the customer’s existing solution.
2. Small changes to E2, H4, H5 and I1.
3. K1 (Acquisition plan) is adapted to EU procurement rules.
4. L1 (Response time) may use “number of users” to define the nominal load.

Version 8.0: October 2020

1. B3 (POC) has got a requirement about the supplier abiding by his proposal for 100 work days.
2. D4 CREDO is a check task and data match.
3. F (integration) and F0-6 now mentions the need for checking the integrations.
4. Minor changes in K1, K2 and K3.

#### Guide booklet

Guide booklet version 9 is based on requirements template 9 with a matching SL-07 contract. The contract is presented and explained in the guide booklet, Chapter 6.

# Background and supplier guide

## Background and vision

Presently the customer has several old EHR systems that he wants to replace with one system to obtain:

1. More efficient support for the clinical work,
2. Better possibilities for integration with future systems,
3. Lower cost of operation.

Midland Hospital has around 5,000 employees, 800 of which are doctors. The hospital has around 50,000 in-patients a year and around 250,000 outpatients.

The customer expects that the supplier has a COTS system (Commercial-Off-The-Shelf system) that can meet many 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 present and future situations are illustrated with these context diagrams. The supplier's responsibilities are indicated: The box with double-line border shows the system to be delivered. Double-line arrows show integrations to be delivered. There is presently insufficient integration between the EHR system and the medication system. The customer wants an EHR system that includes a medication system. It may be the customer’s present medication system or a new one delivered as part of the EHR system.

## 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, and to requirement 4 in D3.1 like this: D3.1-4.

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:

1. **Assumptions** behind the requirements, for instance that the task must be supported for this kind of users, this frequency of use, etc.
2. **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.
3. **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.
4. **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 *Requirements 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.

## Customer options

The customer wants a proposal for these options . . .

1. Integration with two old MR scanners

. . .

## Overall solution

The solution is based on . . .

## Supplier options

The supplier proposes these additional options:

1. High availability

99.8%. See details in L2-2 and prices in contract appendix 2.

2. Ultra availability

99.95%. See detalails in L2-2 and prices in contract appendix 2.

3. Integration with the WHO DNA database

WHO provides possibility for …

# 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.

## Patient treatment (future flow)

The system shall support only one kind of flow: treatment of a patient. The table below is the general, logical flow of a treatment. Many of the steps can be omitted (e.g. step 2 and 8) or repeated several times during the treatment (e.g. step 3 to 9).

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

| **Steps in patient** treatment | **Tasks and subtasks** |
| --- | --- |
| 1. Admit the patient either through GP (General Practitioner), the patient in person or acutely (e.g. traffic accident with unconscious patient). | C1, C2 |
| 2. Call the patient to make an appointment. | C1-4 |
| 3. The patient arrives to the ward. Examine the patient to make a diagnosis, including making tests on the spot or through a lab. | C10-1, 2, 3  C12 |
| 4. Plan the treatment, including ordering medicine, booking time, order implants, etc. | C10-6, C11, C13 |
| 5. Maybe transfer the patient to another ward, for instance in case of several diagnoses. | C3 |
| 6. Treat the patient. | C10-3, C14 |
| 7. Evaluate the result. Maybe perform further tests and treatments. | C10 |
| 8. Make appointments for check-ups. | C10-6 **?** |
| 9. The patient arrives for check-up. Perform various tests and maybe supplementary treatments. | C10 |
| 10. Arrange home care. | **?** |
| 11. Discharge the patient and inform relevant parties, e.g. own GP or social services. The patient may also have died. | C6, C7 |
| 12. Settle accounts | C8 |

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. In the case above, it revealed some flaws in the tasks, marked by question marks.

## 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. Efficient support of all user tasks. | All relevant data are available during the task without switch­ing between several systems. All par­ties can see the health record. | Support for all tasks in Chapter C. System integration, particularly F2. Adequate response times in L1. |  |
| 2. Reduce medica­tion errors from 10% to 2%. | Avoid manual steps - record the order immediately.  The system checks for validity, drug interaction, etc. | Support for task C10 (clinical session), in particular problem 2p (assess the state of the patient) and 6q (errors at hand-over).  Support for task C11 (order medicine), almost all the subtasks. |  |
| 3. Continuous improvement of the work proc­esses. | Easy to set up and modify standard treatment plans.  Easy to integrate the system with new lab systems, etc. | Requirements in sections E4 and F10 (system expansion and integration with new systems). |  |
| 4. Lower opera­tional costs. | Acquire a new, hopefully cheaper, system. | All the requirements and the selection criteria in B5. |  |
| 5. Meet the new EU rules on ... | … | … | 1-1- 2020 |

## 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. The date must be stated as the number of workdays after signing the contract. The customer expects 40 workdays or less.

| **Areas where an early proof of concept is required:** | **Example of proof:** | **Code:** |
| --- | --- | --- |
| 1. Efficient support of clinical sessions (task C10). | A prototype of the necessary computer screens (maybe a paper mockup) is assessed by expert users. Can be done within \_\_ workdays.  (See also area 5 below.) |  |
| 2. Usability (all requirements in section I1). | A prototype (maybe a paper mockup) is usability tested with ordinary users.  Can be done within \_\_ workdays. |  |
| 3. Response times with the required number of users (all requirements in section L1). | A test setup is used to simulate the required number of users. The response times are meas­ured. Can be done within \_\_ workdays. |  |
| 4. Possibility for third-party expansion of the system (sections E4 and F10). | An independent software house studies documentation of parts of the system and the technical interfaces in order to assess whether it is adequate for expanding the system.  Can be done within \_\_ workdays. |  |
| 5. Integration with other systems. | A test setup which demonstrates the data exchange. Can be done within \_\_ workdays. |  |

| In case the contract is terminated: | Example solution: | Kode: |
| --- | --- | --- |
| 10. In case the customer contracts with another supplier, but the contract is terminated (e.g. during POC), the customer would like to contract with the next supplier in priority sequence, without a new request for proposal. | The supplier abides by his proposal for \_\_\_ workdays after the proposal deadline. The customer expects 100 work days. |  |

In case this need disappears, the customer will as soon as possible inform the supplier.

## Minimum requirements

In public tenders according to EU rules, the suppliers must know the minimum requirements and selection criteria before writing a proposal. In commercial acquisitions, the customer needs not state any criteria.

**Scores:** The customer gives each proposal scores for the requirement areas shown in the table below. To provide better overview, the tables have space for several proposals (columns A, B and C). The detailed requirements in Chapter C to L explain where the problems are today. 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).

**Minimum score**: For each requirement area, the customer has stated the minimum scores below. A system that that doesn't meet the minimum scores in all areas, will be useless in practice.

**Minimum requirements**: The system must meet the minimum scores below in all requirements areas.

Notice that a minimum score may be -2 or -1. This means that a proposal may be acceptable even if it is worse than the present system in this area. As an example, area C1-C4 has a minimum score of -2 because the customer can use his existing admission system. The table shows an example where supplier A scores -1 (worse than today) for area H2-H6, but this is acceptable because the minimum score is -1.

| **Requirement area** | **Points today** | **Minimum**  **score** | **Score** | | |
| --- | --- | --- | --- | --- | --- |
| **Sup A** | **Sup B** | **Sup C** |
| C1-C4. Admit and discharge patients (considered one area). | 0 | -2 | 1 |  |  |
| … |  |  |  |  |  |
| C10. Perform clinical session. | -1 | 0 | 1 |  |  |
| C11-C… Medication (considered one area). | -1 | 0 | 2 |  |  |
| … |  |  |  |  |  |
| D. Data. Assessed through the task support. | N/A | N/A | N/A |  |  |
| … |  |  |  |  |  |
| F10. Integration with new systems. | -1 | 1 | 1 |  |  |
| H1. Login and access rights for users. | 0 | 0 | 0 |  |  |
| H2-H6. Other security (one area). | 0 | -1 | -1 |  |  |
| I. Usability and design. | 0 | 0 | 1 |  |  |
| J2. User training. | 0 | 0 | 0 |  |  |
| J4. Data conversion. | N/A | 0 | 1 |  |  |
| K. Acquisition process (project cost) | N/A | -1 |  |  |  |
| L1. Response times. | 0 | 0 | 0 |  |  |
| … |  |  |  |  |  |

## Selection criteria: Highest net benefit

Use either section B5 or B6 as selection criteria.

The total benefit of the proposal is based on a financial value for each business goal. The table shows an example with fictitious figures for supplier A.

**Potential:** The customer's estimate of the potential benefit for a 5-year period. Measured in million $.

**Fraction:** For each proposal, the customer estimates the fraction of the potential benefit that this proposal can reach if the supplier delivers as promised. It is stated as a number with one decimal, normally in the range from 0.0 to 1.0. Example: The potential cost saving of efficient task support is estimated to one hour per day per employee. Proposal A seems to save only half an hour and gets the fraction 0.5.

**Risk:** For each proposal, the customer estimates the risk that the fraction will not be met. The risk is estimated based on how detailed the solution is, whether the relevant part of the solution exists, whether it is used elsewhere, the supplier's domain knowledge, and the time proposed for the proof of concept. Example: Supplier A has sketched a detailed solution but it doesn't exist yet. However, he has good domain knowledge. The risk is estimated to 30%.

**5-year value:** Computed as Potential \* Fraction \* (1-Risk)

| **Business goal** | **5-year poten­tial** | **Fraction** | | | **Risk** | | | **5-year value** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SupA** | **SupB** | **SupC** | **SupA** | **SupB** | **SupC** | **SupA** | **SupB** | **SupC** |
| 1. Efficient support of clinical tasks | 200 | 0.5 |  |  | 30% |  |  | 70 |  |  |
| 2. Reduce medication errors | 50 | 1.0 |  |  | 10% |  |  | 45 |  |  |
| 3. Continuous improvement | 50 | 1.0 |  |  | 40% |  |  | 30 |  |  |
| 4. Lower operating costs (included below) |  |  |  |  |  |  |  |  |  |  |
| **Total benefit for 5 years (million $)** | **300** |  |  |  |  |  |  | **145** |  |  |

The customer estimates the net benefit for each proposal. The total benefit for a period of 5 years is computed above. The costs of deploying and operating the system are subtracted. The result is the net benefit for 5 years. Notice that all the figures may differ between proposals.

The customer selects the proposal with the highest net benefit for 5 years.

| **Benefit for 5 years, million $** | **Sup A** | **Sup B** | **Sup C** |
| --- | --- | --- | --- |
| **Total benefit for 5 years** | **145.0** |  |  |
| Product cost | 20.0 |  |  |
| Customer hardware costs | 10.0 |  |  |
| Staff training | 5.6 |  |  |
| Operating costs for 5 years | 20.0 |  |  |
| **Total costs for 5 years** | **55.6** |  |  |
| **Net benefit for 5 years** | **89.4** |  |  |

## Selection criteria: Most score points per dollar

With this alternative the customer doesn't have to specify the benefit in $, and he doesn't have to reveal to the supplier how much he expects to gain. Risks are not included below, but it could be done.

**Scores:** The scores are those the customer assessed for the minimum criteria in B4. Since one of the areas has a very high weight, the decision is very sensitive to this area getting score 1 or 2. For this reason, we give it a score with one decimal here.

**Weight:** Each requirement area has a weight that reflects the impact of the area. For instance, the number of staff affected, the impact on the customer's service quality, or the contribution to the business value. The weights add up to 100.

| **Requirement area** | **Weight** | **Score** | | | **Weighted score** | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sup A** | **Sup B** | **Sup C** | **Sup A** | **Sup B** | **Sup C** |
| C1-C4. Admit patient (considered one area). | 5 | 1 |  |  | 5 |  |  |
| … |  |  |  |  |  |  |  |
| C10. Perform clinical session. | 50 | 1.5 |  |  | 75 |  |  |
| C11-C… Medication (considered one area). | 15 | 2 |  |  | 30 |  |  |
| … |  |  |  |  |  |  |  |
| D. Data. Assessed through the task support. | N/A | N/A |  |  |  |  |  |
| … |  |  |  |  |  |  |  |
| F10. Integration with new external systems. | 15 | 1 |  |  | 15 |  |  |
| H1. Login and access rights for users. | 0 | 0 |  |  |  |  |  |
| H2-H6. Other security (one area). | 0 | -1 |  |  |  |  |  |
| I. Usability and design. | 10 | 1 |  |  | 10 |  |  |
| J2. User training (training cost below). | N/A | 0 |  |  |  |  |  |
| J4. Data conversion. | 0 | 1 |  |  |  |  |  |
| K. Acquisition process (project cost below). | N/A |  |  |  |  |  |  |
| L1. Response times. | 5 | 0 |  |  |  |  |  |
| … |  |  |  |  |  |  |  |
| **Total weight and total weighted score points** | 100 |  |  |  | 135 |  |  |

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. Finally the score per million $ is computed.

The customer selects the proposal with most score points per million dollars.

| **Score per million $** | **Sup**  **A** | **Sup**  **B** | **Sup**  **C** |
| --- | --- | --- | --- |
| **Total weighted score points** | 135.0 |  |  |
| Product cost | 20.0 |  |  |
| Customer hardware costs | 10.0 |  |  |
| Customer project cost | 0.3 |  |  |
| Staff training | 5.3 |  |  |
| Operating costs for 5 years | 20.0 |  |  |
| **Total costs for 5 years** | **55.6** |  |  |
| **Score points per million $** | **2.4** |  |  |

# 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 case 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: Patient management

This work area comprises calling in patients, monitoring waiting lists …

**User profile: Doctor's secretaries**. Most of them are experienced IT users with good domain knowledge. They communicate easily with medical staff.

**User profile: Clerical staff** …

**Environment:** Office …

## Admit patient before arrival

This task creates an admission or continues a parked admission. Most admissions can be handled as one piece of work. The rest have to be parked, e.g. because some information is missing. It is important that the system ensures that parked admissions are not forgotten (see E1-1).

**Users:** Initially a doctor's secretary, but the case may be transferred to someone else.

**Start:** Message from medical practitioner, message from another hospital … message with missing data, or a reminder about a parked admission.

**End:** When the patient has been admitted or recorded on the waiting list, or when the admission has been parked while the missing data is on its way.

**Frequency:** In total: Around 600 admissions per day. Per user: A maximum of 40 per day.

**Difficult:** (never)

|  |  |  |
| --- | --- | --- |
| Subtasks and variants: | Example solutions: | Code: |
| 1. Record the patient. (See data description D5). |  |  |
| 1a. The patient is in the system. Update data. |  |  |
| 2. Admit also a healthy companion. |  |  |
| 3. Record the admission, including the initial diagnosis. (See data description D1 and D6). |  |  |
| 3a. Transfer data from medical practitioner, etc. | The system uses the MedCom formats. |  |
| 3p. **Problem**: Some electronic messages don't follow the MedCom format. | The system allows manual editing of the transferred message. |  |
| 3q. **Problem**: The patient may have several admissions at the same time at different hospitals and departments. It is hard to see who is responsible for nursing and where the bed is. |  |  |
| 4. Find a meeting time for the patient and send an admission letter or a confidential email. |  |  |
| 4a. Put the patient on the waiting list. |  |  |
| 4b. Essential data is missing. Park the case with time monitoring. |  |  |
| 4c. Transfer the case to someone else, possibly with time monitoring. |  |  |
| 4d. Maybe reject the case. |  |  |
| 5. Request an interpreter for the meeting time. |  |  |

## Admit immediately

This task creates an admission for a patient who arrives in an emergency without prior notice …

Work area 2: Patient treatment

This work area comprises …

## Perform clinical session

A clinical session may comprise diagnosis, planning, treatment, evaluation, etc. Usually several of these are carried out, but it may also happen that only planning, for instance, is carried out.

**Users:** …

**Start:** Contact with the patient or a conference about the patient.

**End:** When nothing else is to be done about the patient right now.

**Frequency:** In total: Around 15,000 per day. Per user: A maximum of 20 per day.

**Difficult:** Disaster with many injured. (Better describe it as a separate task. See the guide booklet.)

| Subtasks and variants: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Identify the patient. | The system can read an electronic brace­let, e.g. for uncon­scious patients. |  |
| 2. Assess the state of the patient. See open diagno­ses and related indications. See notes. See results of services ordered earlier and compare them with expectations. The data to overview comprises D0 … | The system shows an overview of everything on one screen, e.g. with a Gantt-like time dimension. The user can drill down to details from the overview. |  |
| 2p. **Problem**: Today clinicians have to log in an out of several systems to see all relevant data. |  |  |
| 3. Provide services that can be given on the spot, e.g. blood pressure and SAT. | The system makes it easy to record the results on the spot. |  |
| 4. Follow up on planned services and results. Check for violated deadlines. | The overview shows ordered services and their state, e.g. deadline violation. |  |
| 5. Adjust diagnoses (modify, add, delete, prioritize). Check against standard recommendations. Write notes. |  |  |
| 5p. **Problem**: Cumbersome to see standard recommendations. | The system can show recommenda­tions and checklists based on a selected diagnosis. |  |
| 6. Plan and order new services. Check against available time for all parties - including the patient. (See the long subtasks C11, C12 … for order medicine, booking …). | For bookings, the system shows available dates and times for all parties. |  |
| 6p. **Problem**: Parts of the request are forgotten. | The system can book standard packages of services. |  |
| 6q. **Problem**: Errors when data are written on paper and recorded later. | The system makes it easy to record on the spot. |  |
| 7. Maybe discharge the patient. (See task C6). |  |  |

## Order medicine for the patient (long subtask)

This is not a separate task but a long subtask carried out during a clinical session. For this reason "start", "end", and "user" are unnecessary.

**Frequency:** In total: Around 30,000 times per day. Per user: A maximum of 20 times per day.

| Subtasks and variants: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Assess the entire medication pattern of the patient, in this admission as well as other admissions. | The system shows an overview of all medications, CAVE and diagnoses. |  |
| 1p. **Problem**: Cumbersome to see standard recommendations | The system can show recommenda­tions and checklists based on diagnosis and drug type. |  |
| **…** |  |  |
| 6. Calculate dose. Check that it is reasonable. Check for interaction with other drugs. | The system offers a calculation based on body weight retrieved from the health record. |  |
| 6p. **Problem**: Translation between various units. There may be a difference between the unit of ordering (e.g. mg) and the unit of dose (e.g. number of tablets). | The system shows the dose in order units as well as dose units. |  |
| **…** |  |  |

**…**

## Perform clinical session, mobile

Clinical sessions may be performed when medical staff is moving around from patient to patient, e.g. with a PDA or mobile phone. In principle, we have the same subtasks as in C10, but they cannot be supported in the same way. In order to allow the supplier to specify his solution for the mobile situation, we repeat the clinical session task here.

**Users:** …

**Start:** When …

**End:** When …

**Frequency:** …

# 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 many cases data has to be exchanged with external systems as specified in Chapter F.

Figure 3 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, D5 is a pile that holds a card for each person the system deals with. Next to the box is a list of the fields on the card.

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, a person can have many admissions, but an admission relates to only one person. Data need not be structured this way in the system, but it must be handled in some way.

The dotted boxes show data that are (partly) shared with external systems.

**D1. Diagnosis:** Each record contains data about one of the patient's diseases. It corresponds to the National Health Classification (SKS), but there is also a need for recording diseases that are not in SKS or cannot be classified until later.

**D2. Diagnosis type:** Each record contains data about a type of diagnosis - independent of the patient: the diagnosis name and SKS code (where possible), recommendation, standard treatment packages (through the relationship to the catalogue of service types) … The clinicians will choose diagnoses from this catalogue of diagnosis types.

**…**

**D5.** **Person:** Each record holds data about a person: name, address … A person may be a clinician, a patient, or a relative.

**D6.** **Admission:** Each record holds data about an admission: start time, related person …

## 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). |  |  |

## Diagnosis

A diagnosis is a disease or a symptom for a specific patient.

**Examples**: There is a fuzzy distinction between diseases and symptoms. As an example, cholera as well as coughing are "diagnoses".

**Data source**: Diagnoses are recorded during clinical sessions (C10) and often during admission (C1).

**Data use**: Diagnoses are shown in patient overviews, for billing and for government reporting.

| Data volume: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Around 800,000 diagnoses are recorded a year. |  |  |

| Fields and relationships: | Example solutions: | Code: |
| --- | --- | --- |
| 2. Diagnosis Code: Relation to Diagnosis Type. The patient's primary diagnosis may change during the admission. The primary diagnosis type is used for billing and government reporting. |  |  |
| 2p. Problem: Very hard to select the right SKS code from the 20,000 possible ones. | See solution notes below. |  |
| 3. Admission: Relation to the Admission, which in turn refers to the patient (Person). | The system records it automatically based on the currently selected patient. |  |
| 4. Name: Usually the name from Diagnosis Type, but may be a name entered for this specific patient. | Field length: 100 characters. |  |
| 5. State: A diagnosis may be in these states: Obs, valid, canceled, closed. |  |  |
| 6. Start Time: The date and time from which the diagnosis is in this state. Usually it is the same as the Change Time, but not always, e.g. if you record that the patient started coughing yesterday. | The system makes it easy to choose the Recording Time as the Start Time. |  |
| **…** |  |  |
| 17. Recommendation: The recommendation valid at the time of creating the diagnosis. |  |  |

Solution notes

The user might for instance select a diagnosis code in these ways:

1. Browsing a hierarchy (corresponding to the SKS super and subclasses)
2. A reduced hierarchy so that the department as a default see only the diagnoses relevant for them.
3. "Live search" where the user enters part of the diagnosis name, and the system shows possible matches keystroke by keystroke.

## Diagnosis Type

The collection of diagnosis types makes up the diagnosis catalogue.

**Examples**: DA009: Cholera vibrio eltor; DR059: Coughing.

**Data source**: Imported from the SKS web site.

**Data use**: The user selects a diagnosis type when recording a patient diagnosis.

|  |  |  |
| --- | --- | --- |
| Data volume: | Example solutions: | Code: |
| 1. There will be around 30,000 diagnosis types. SKS has presently around 20,000 types. |  |  |

| Fields and relationships: | Example solutions: | Code: |
| --- | --- | --- |
| 2. Diagnosis code: SKS code or a temporary code. |  |  |
| 3. Name: The full name of the diagnosis, e.g. "Cholera without specification". |  |  |
| 4. State: A diagnosis type can be in one of these states: Considered, valid, outdated. |  |  |
| 5. Parent: Relation to a more general diagnosis type, e.g. "Cholera, DA00”. | Example: "Cholera, DA00".  WRONG - not a **solution** example. |  |
| 6. Description: A longer text, but not more than one or two lines. Even longer descriptions may be found in the "Recommendation". | Field length: 160 characters. |  |
| 7. Service types: Relation to service types that may be used to treat this diagnosis. | The system may extract the informa­tion from the Recommendations. |  |
| **…** |  |  |
| 10. Recommendation: A long text describing indications, medical practice, etc. | Might be a URL. |  |

## Service

A service is something measured or given to the patient. There are many subclasses of service, e.g. measurements, surgery and medication. At present they are stored in separate tables or even in external systems to be integrated.

| Fields and relationships common for all services: | Sample solutions: | Code: |
| --- | --- | --- |
| 1. Service code: Relation to Service Type. |  |  |
| 2. Admission: Relation to the Admission, which in turn refers to the patient (Person). | The system records it automatically based on the currently selected patient. |  |
| 3. Start time: The date and time the service was given. |  |  |
| 4. State: In the normal flow a service may be in these states: Ordered, confirmed (by the service provider), started (e.g. sample taken), comple­ted, assessed (by the clinician). Exceptionally, the state may be: Canceled, changed. |  |  |
| 5. Consists of: Relation to services that are part of this service, e.g. surgery that consists of several services. |  |  |

### Patient measurement

**Examples**: Blood pressure; Body Weight; B-glucose; Gamma globulin; X-ray.

**Data source**: Some are recorded during a clinical session; others are imported from an external system, e.g. lab results.

**Data use**: Used in patient overview and detail view to support diagnosing and treatment.

| Data volume: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Around 100,000 measurements are recorded a day. Of these 5,000 are pictures. |  |  |

| Fields: | Example solutions: | Code: |
| --- | --- | --- |
| 2. A patient measurement should include the data from the present table, see Figure 4, tblPatient­Measurement. Notice that the present table doesn't have these common service fields: admissionID and state. |  |  |

### Patient surgery

**Examples**: Heart Bypass Operation; Photodynamic Therapy (PDT).

**Data source**: Recorded during and after surgery.

**Data use**: Used in patient overview and detail view to support diagnosing and treatment.

| Data volume: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Around 100 surgeries are recorded a day. |  |  |

| Fields: | Example solutions: | Code: |
| --- | --- | --- |
| 2. A patient surgery record should include the data from the present table, see Figure 4, tblPatientSurgery. Notice that the present table doesn't have the common service fields: admissionID and state. |  |  |



### Patient medication

**Examples**: Ibumetin, 400 mg\*3; Furix, 40 mg\*2.

**Data source**: Recorded as medicine orders during clinical sessions.

**Data use**: Used in patient overview and detail view to support diagnosing and treatment.

| Data volume: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Around 30,000 medicine orders are recorded a day. |  |  |

| Fields: | Example solutions: | Code: |
| --- | --- | --- |
| 2. A patient medication record should include the data that the present system shows. See Figure 5, screen shot from the present medication system. |  |  |



## CREDO check

This section is not requirements, but a check to see that tasks and data match and nothing is missing. CREDO means Create, Read, Edit, Delete and Overview. For each row (record) in a table, it must through some task be possible to Create it, Read it, Edit it, Delete it and see it in some Overview/list.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CREDO check** | D5. Person | D6. Admission | D1. Diagnosis | D2. DiagnosisType | . . . |
| C1. Admit | C---O | C---- | -R--- | -R--O |  |
| C2. AdmitImmed. | C---- | C---- | -R--- | -R—-O |  |
| C6. Discharge | -RE-- | -RE-- | -RE-O | -R--- |  |
| . . . |  |  |  |  |  |
| C10. ClinicalSession | -RE-O | -RE-O | CRE-O | -R—-O |  |
| C11. . . . |  |  |  |  |  |
| F1. SKS integration | ----- | ----- | ----- | C-ED- |  |
| Missing | ---D- | ---D- | ---D- | ----- |  |

As you can see, no task can Delete Person, Diagnosis and Admission. It is because these data are kept for 20 years, and then deleted automatically. Requirements about this are missing. DiagnosisType will be created, changed and deleted automatically as part of the SKS-integration F1.

# 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.

## System generated events

| The system must generate these reminders: | Example solutions: | Code: |
| --- | --- | --- |
| 1. If an admission has been parked for X days, the doctor's secretary must be reminded. System administration must be able to define X. | X is typically 4 days, but may vary between departments. |  |
| 2. If a LabSys service has been ordered but not completed within 24 hours, the clinicians must be reminded. |  |  |

## 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, cross-task purposes, and reports with a precise format. They are specified here.

| Report requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Checks must be printed on preprinted forms with the format shown in … |  |  |
| 2. The system can show an overview and a forecast of the bed occupation (used for instance in task …). | Figure … shows an example of such a report. |  |
| 3. The supplier can develop up to 100 new reports at a fixed price as part of the maintenance. | See appendix 2. (The price may depend on the complexity.) |  |
| 4. The system has a report generator that is easy to use. It can combine data from all classes in the datamodel. | \_\_% of super users can develop reports like those in appendix X. The customer expects 50%. |  |
| 5. The system can show all reports on the screen as well as on print, and save them as a file. |  |  |
| 6. Super users can explore data in an ad-hoc way. | The system can transfer data to a spreadsheet. |  |
| … |  |  |

## Business rules and complex calculations

Some business rules are specified in the task steps, e.g. *Check that* … (example in C11-6). Other business rules are specified in the data descriptions (example in D3-4), and some are specified as access rights (section H1). Here are additional business rules and complex functions:

| Function: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Waiting list priority must be calculated as described in … |  |  |
| 2. Salary calculations must at any time follow the collective agreements (see also the maintenance requirements in …). |  |  |
| 3. Normally, a diagnosis may change state only as described in Figure 6. In case of mistakes, the user must be able to deviate from the rules (see also H4-2). | A user who tries to deviate from the rules will be asked whether it is intentional. If so, the change is made and logged in … |  |
| 4. Inside the system, a service requested from LabSys changes state as described in Figure 7. |  |  |

Requirement note: State-transition diagrams

Figure 6 shows that a clinician creates the diagnosis. It is created in either state *Obs* or state *Valid*. Clinicians can change the state according to the diagram. The diagnosis disappears when the system automatically cleans up the data after 20 years.

Figure 7 shows how the state of a LabSys service changes inside the system. A clinician creates a LabSys service in state *Ordered*. During the creation, the system sends a LabRequest to LabSys. When LabSys sends a *LabConfirm* message to the system, it changes the service state to *Confirmed*. A clinician takes a sample from the patient, sends it to the lab and tells the system, which changes the service state to *Started*. The service can change state in other ways as specified in the diagram.

## Expansion of the system

The system shows and maintains data through the user screens. In this section, "customer" means the customer's own IT staff or a third party authorized by him. The customer expects to be able to modify the screens and add new ones in order to create overview for medical specialties, new work procedures, etc.

The system handles many types of medical services, often with special combinations of data. The customer expects to be able to add new types of services. The requirements below state the demands.

| Expansion requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The customer can define new types of services based on data in Chapter D. |  |  |
| 2. The customer can define screens that combine data from the entire data model in Chapter D (arbitrary views of data). |  |  |
| 3. A screen can activate functionality in the EHR system and in external systems integrated with the EHR system. E.g. request of a service, notification, print of a report. |  |  |
| 4. A screen can be composed of many types of components (controls) and their color can reflect data values. E.g. text boxes, tables, buttons, graphs, pictures. |  |  |
| 5. The customer can add new types of components for use in the screens. |  |  |
| 6. Screens can be defined for several kinds of equipment, e.g. PC, PDA, Smartphone. |  |  |

| Documentation and rights: | Example solutions: | Code: |
| --- | --- | --- |
| 7. The tools for composing screens, adding new component types, etc. must be documented in such a way that the customer's IT staff or a third party can understand them and use them for the intended purpose. | A course of \_\_\_ days is necessary to use the tools. |  |
| 8. The customer must have the right to use the tools and the data stored in the system. |  |  |

# Integration with external systems

The system must integrate more or less closely with the external systems shown in Figure 8 (context diagram). The integration comprises data sharing or replication, and the ability for the user to activate functionality in the external systems.

In this Chapter, "customer" means the customer's own IT staff or a third party authorized by him.

**S-Data** (System data) are the integrated data stored locally in the EHR system, S.

**E-Data** (External data) are the integrated data stored in the external system, E.

Here is a short explanation of the external systems:

F1. SKS: The National Health Classification system. The National Health Organization updates it regularly.

F2. LabSys: The customer's present lab system for …

F3. A medication system. It may be the customer’s present medication system or a new one delivered as part of the EHR system (the description is omitted below).

F10. An external system that the customer will buy later and integrate.



**Requirement note: Response times**

The response times specified in this chapter must be interpreted in the same way as in L1, i.e. with L1's fractile, measured in peak load periods, etc.

**Integration aspects**

For each integration there are many aspects to consider:

1. Access rights to data.
2. Protection against loss of data.
3. Documentation and means for integrating the system with other systems, including testing the integration and migrating data. See also testing in J6.
4. Integration responsibility, e.g. the supplier, or the customer with support from the supplier.
5. Tasks the integration must support.
6. Data import from E (the external system). Which data to import.
7. Data recency (how old may the local copy of the data be).
8. Response time at import.
9. Data export to E. Which data to export.
10. Response time at export.
11. Other functions, e.g. warnings to the user or E.

For practical reasons the requirements in group A, B and C are written as common integration requirements, which means that they are valid for all integrations where relevant.

## Common integration requirements

The requirements in this chapter apply for all the integrations unless explicitly stated.

| A. Access rights to data: May be moved to H1 | Example solutions: | Code: |
| --- | --- | --- |
| 1. The system may only transfer E-data to the user's PC when the user has the right to see it according to H1. |  |  |

| B. Protection of data: May be moved to H3 | Example solutions: | Code: |
| --- | --- | --- |
| 2. The system must protect against loss or duplication of data transferred between the systems, e.g. because one or both systems have been off-line or closed down. |  |  |
| 3. The system must protect against concurrency problems, e.g. that user A sees and then updates E-data, while user B does the same. Neither A nor B will notice that the data may become inconsistent. |  |  |
| 4. The system must support error tracing at data transfers. | Logging all transfer errors. |  |

| C. Documentation and rights: | Example solutions: | Code: |
| --- | --- | --- |
| 5. It must be easy to add new interfaces, e.g. SOA services, database queries, or API's. | The system provides an OData interface that allows the client to define services.  Or: The supplier can do it at a fixed price. |  |
| 6. The customer needs to test the integration, e.g. creating and changing test data in the E-system. See also J6. |  |  |
| 7. The customer must have the means and rights to extract and use all data described in Chapter D, e.g. for converting the data to another system. |  |  |
| 8. The technical interfaces to S must be documented. The documentation must be understandable to a typical software house and found suited for integration and data retrieval. | A course of \_\_\_ days is necessary to use the documentation and make the integration. Documentation samples are delivered early (see B3-4). |  |
| 9. The customer must have the right to use the documentation and the interfaces themselves. |  |  |
| 10. The supplier must loyally support the customer in the integration or migration effort with qualified staff at a fair price. |  |  |

## SKS

**E-data (e**xternal data):The SKS tables comprise codes and corresponding names for diagnoses, services, health departments, etc.

**Tasks:** The codes and names are used in most of the tasks. However, the department codes are retrieved from another system.

**E-support:** The tables are publicly available from the web site of the National Health Organization. They are zip text files with fixed field spacing. They are documented on the same web site.

**E-updates:** The department data are updated on a monthly basis, the other codes every three months.

**Data volume:** The SKS tables comprise around 100,000 records, each around 100 characters.

Alternative 1: All points considered

| D. Integration responsibility: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier must integrate the system with the SKS tables. |  |  |

| E. Task support: No special requirements. | Example solutions: | Code: |
| --- | --- | --- |

| F. Data import: | Example solutions: | Code: |
| --- | --- | --- |
| 2. All codes and names are needed, except the department data. |  |  |

| G. Data recency: | Example solutions: | Code: |
| --- | --- | --- |
| 3. S-data should not be older than a week. | The system imports E-data every \_\_ days.  Or: IT support starts a transfer when the Health authorities announce that data are available. |  |
| 3p. Sometimes new SKS codes conflict with local codes or cause other problems. | IT support can roll SKS data back to the previous version.  Or: Local codes may have a tag so that they don't conflict. |  |
| 4. In special cases, there may be demand for more recent data. | IT support can start a data transfer. |  |

| H. Response time at import: No requirements. | Example solutions: | Code: |
| --- | --- | --- |

| I. Data export: None. | Example solutions: | Code: |
| --- | --- | --- |

| J. Response time at export: N/A. | Example solutions: | Code: |
| --- | --- | --- |

| K. Other functions: No requirements. | Example solutions: | Code: |
| --- | --- | --- |

Alternative 2: The short version

| Integration requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier must integrate the system with the relevant SKS tables. |  |  |
| 2. The system must use the new codes and names shortly after their release. | The system or the maintenance staff transfer the tables within a week after their release. |  |

## LabSys

**E-data** (external data):LabSys version yyy. Users can request lab tests from LabSys. The sample itself is delivered by … and the reply comes on fax and electronically. One reply may contain several results.

**Tasks:** LabSys is used in connection with task C10, Perform clinical session.

**E-support:** The technical interface to LabSys is described in … MediData supports LabSys in Denmark and can provide integration support. The customer has contracted the rights with MediData.

**E-data updates:** LabSys generates replies continuously by fax, but at present the electronic replies are only sent as a batch over night.

**S-data updates:** The entire hospital requests around 8000 tests a day, mainly between 8:00 and 16:30.

**Data volume:** Each reply consists of one or more results, each of around 500 characters.

| D. Integration responsibility: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier must integrate with LabSys. |  |  |

| E. Task support: | Example solutions: | Code: |
| --- | --- | --- |
| 2. The integration must support C10 in an efficient manner. | Requests and replies are handled in the same way as other services - without retyping patient ID. |  |

| F. Data import: | Example solutions: | Code: |
| --- | --- | --- |
| 3. All E-data that can match the data in section D3. |  |  |

| G. Data recency: | Example solutions: | Code: |
| --- | --- | --- |
| 4. S-data should not be older than 3 hours. | The system imports E-data every \_\_ hours.  Or: Data is imported at E request when they are available.  Or: Data is always retrieved from E. |  |
| 5. Sometimes the latest results are needed for a specific patient, e.g. during surgery. | The system retrieves data on the user's request.  Or: Data is always retrieved from E. |  |

| H. Response time at import: | Example solutions: | Code: |
| --- | --- | --- |
| 6. When the user requests a lab reply, it must be so fast that the user doesn't lose patience. | The result is visible within \_\_ s plus the time LabSys needs to send the reply. (The customer expects 3 s.) |  |

| I. Data export: | Example solutions: | Code: |
| --- | --- | --- |
| 7. The user can send LabSys requests through the EHR system (S). |  |  |

| J. Response time at export: | Example solutions: | Code: |
| --- | --- | --- |
| 8. A lab request can be sent and the user continue typing within the mental switching time (around 1.3 s). The confirmation from LabSys should be visible a bit later. | Typing is possible within \_\_ s. (The customer expects 1.3 s.)  The confirmation from LabSys appears \_\_ s after LabSys has sent it. (The customer expects 3 s.) |  |

| K. Other functions: | Example solutions: | Code: |
| --- | --- | --- |
| 9. S can notify the user about new or missing LabSys replies. |  |  |
| 10. S can notify LabSys (E) about missing LabSys replies (reminders). |  |  |

## Integration with new external systems

The customer expects that he can integrate new external systems (E) with S - with little or no help from the supplier of S. This section lists the demands such integrations might have and asks for the supplier's suggestion for what he can deliver to meet the needs.

**External system:** In principle any system. Examples: X-ray system, mobile applications, specialist system for intensive care.

**Tasks:** Defined later.

**E-support:** The customer's responsibility.

**E-data updates:** Defined later.

**Data volume:** Defined later.

| D. Integration responsibility: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The customer is responsible for the integration. The supplier must assist as specified in F0-10. |  |  |

| E. Task support: | Example solutions: | Code: |
| --- | --- | --- |
| 1. For mobile applications E may in some periods be off-line. When E connects to S again, data synchronization is needed. | The customer can configure S to automatically synchronize data at reconnect. |  |

| F+G. Data import and data recency: | Example solutions: | Code: |
| --- | --- | --- |
| 3. S can import data from E assuming that they fit into S's existing data tables. | The customer can configure S to import at S's request or E's request. |  |
| 4. S can periodically import data from E. | The customer can configure S to do this. |  |
| 5. S can optimize the import by asking only for data younger than a certain point in time. | The customer can configure S to do this. |  |
| 6. S can optimize the import by asking for data about a specific patient only. |  |  |

| H+J. Response time at import and export: | Example solutions: | Code: |
| --- | --- | --- |
| 7. S can scale up to carry a significantly higher load than specified in L1 with the response times specified in L1. The customer may use this additional load for data transfers. | The system can scale up to handle a load \_\_\_ times as high as required in L1. (The customer expects 2 times.) |  |
| 7p. When a long transfer is in progress, it may block for shorter transfers so that they have a very long response time. | The system can handle several concurrent transfers. |  |

| I. Data export: | Example solutions: | Code: |
| --- | --- | --- |
| 8. S can export data to E assuming that the data exist in S's existing data tables. | The customer can configure S to export at S's request or E's request. |  |
| 9. S can periodically export data to E. | The customer can configure S to do this. |  |
| 10. S can optimize the export by sending only data younger than a certain point in time. | The customer can configure S to do this. |  |
| 11. S can optimize the export by sending data about a specific patient only. |  |  |

| K. Other functions: | Example solutions: | Code: |
| --- | --- | --- |
| 12. S can use functionality in E, e.g. request services or warn about missing requests. | The supplier is asked to specify the functionality S can use. |  |
| 13. E can use functionality in S, e.g. notifying the user or printing on printers managed by S. | The supplier is asked to specify the functionality S provides. |  |

# Technical IT architecture

## Existing hardware and software Alternative 1: Use what we have

At present, the customer has the following IT equipment, which is intended for operating the new system:

1. 2 servers of type …
2. 300 PCs with Windows XP and at least 100 GB disks.
3. Optical fiber net …
4. Oracle database …

The equipment is used by other applications at the same time, but within these limits:

1. Within any 1 second period, servers leave 50% of the speed capacity for the EHR system.
2. Within any 1 second period, the optical fiber net leaves 50% of the capacity for the EHR system.
3. No other applications run on a PC when it runs the EHR system.

| Platform requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Initially the system must run on the existing equipment and meet the requirements in L1, L2 and L3 for a limited number of users. | On these conditions the system can serve \_\_\_ users.  The customer expects 20 users. |  |
| 2. In order to reach the full peak load (see L1) the system must be expanded to meet the requirements in L1, L2 and L3. | The customer has to add this equipment \_\_\_\_. |  |
| 3. The browser-based parts must be able to run on common browsers. | MS-Internet Explorer, Chrome, Safari |  |

## New hardware and software Alternative 2: Supplier suggests

The customer intends to buy new equipment to operate the system.

| Platform requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. In order to meet the requirements in L1, L2 and L3 the customer needs new IT equipment. | The customer needs this equipment \_\_\_\_\_. |  |
| 2. When the peak load grows by a factor of two, the system must be expanded to meet the requirements in L1, L2 and L3. | The customer has to add this equipment \_\_\_\_. |  |
| 3. As far as possible, only equipment from the list in appendix X should be used. |  |  |
| 4. The browser-based parts must be able to run on common browsers. | MS-Internet Explorer, Chrome, Safari |  |

## The supplier operates the system Alternative 3: Supplier's problem

| Platform requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier operates the system and uses the necessary equipment to meet L1, L2 and L3. |  |  |
| 2. The browser-based parts must be able to run on common browsers. | MS-Internet Explorer, Chrome, Safari |  |

# Security

## 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.

Alternative 1: Login as today

| Subtasks for user access: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Identify the user with the existing user identification, login method, and time-out method, which is LDAP and LA … |  |  |
| 2. Check that only authorized users get access to systems and data. (See the requirement note below.) | The database system checks the rights.  Or: The user screens show only the authorized functions and data. |  |

Alternative 2: Better and more convenient security wanted

| Subtasks for user access: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Identify the user. (See section H6-3 about the length of passwords.) | A user identifies himself with a user name and a password; preferably also an alternative identification such as voice or finger print recognition. |  |
| 2. The user has been away from the system for some time. |  |  |
| 2p. Problem: Another user may access the system with the rights of the first user. | The system times out after 10 minutes of non-use. |  |
| 2q. Problem: If the system logs out automatically, it is cumbersome to log on again. | The system requires password only. The timeout period may depend on the physical location, for instance a long timeout in the operating room. |  |
| 2r. Problem: If the system logs out automatically, entered data may be lost. |  |  |
| 3. Check that only authorized users get access to system and data. (See the requirement note below.) | The database system checks the rights.  Or: The user screens show only the functions and data he is allowed to use. |  |
| 3p. Problem: Today the users have a password for each system. It is cumbersome to switch between systems and hard to change passwords regularly. As a result, users tend to post passwords where everyone can see them. | Each user has only one user name and one password (single sign-on). |  |
| 4. Stolen passwords are often traded by criminals. Limit the possibility. | Users must change passwords regularly. If a leak has been detected, all passwords can quickly be blocked. |  |

Requirement note: Possible access rights

1. Right to order drugs in department M.
2. Right to see patient data in department M.
3. Right to record clinical data (diagnoses and services) in department M.
4. Right to see data according to patient permission (see H5-4).

**…**

A physician in department M might for instance have rights 1, 2, and 3, while a supervising physician for department M has rights 2 and 3 only.

## Security management

Each department has its own security management.

Or: Security management is centralized for the entire hospital.

The work in security management includes the following subtasks.

Alternative 1: Use the existing security management

The customer uses LDAP and AD and wants to manage all rights in this way.

| Subtasks for security management: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Create and remove users. | Leave it to the existing security management. |  |
| 2. Assign or remove rights for a user. | Leave it to the existing security management. |  |
| 3. Check that the user has the necessary rights. (Strictly speaking, this is a subtask in H1). | The EHR system retrieves the rights data from the customer's existing system. |  |

Alternative 2: The new system has its own security management

| Subtasks for security management: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Assign or remove rights for a user. |  |  |
| 1a. First, create the user. |  |  |
| 1p. Problem: A lot of users need access rights when they start the first day in the month. | The system transfers data from the personnel system once a month. |  |
| 1q. Problem: A temporary employee has been appointed in a hurry and is not yet in the personnel system. Needs access rights anyway. | Possibility for temporary registration in the department, bypassing the central department. |  |
| 1r. Problem: Security management must keep track of the relationship between 4000 users and 300 rights. | Each user is assigned one or more roles, e.g. physician in department M and supervising in department N. Each role has one or more rights, e.g. order medicine and diagnosing. |  |
| 1s. Problem: Security management forgets to assign and remove rights on the right dates, e.g. in connection with hiring and resigning. | Rights and roles can be defined ahead of time and be valid for a certain period, e.g. from the day the person is employed. |  |
| 2. Create new roles with new combinations of rights. |  |  |
| 3. Get an overview of who has which rights and whether some rights have not been assigned to anyone. |  |  |

## Protection against data loss

Data may unintentionally be lost or misinterpreted in many ways.

| Protect against: | Example solutions: | Code: |
| --- | --- | --- |
| 1. (See F0-2 for protection of data against loss or replication during transfer between systems.) |  |  |
| 2. (See F0-3 for protection against concurrency problems with external systems.) |  |  |
| 3. Local concurrency problems, for instance that user A orders medicine, but before the system has recorded it, user B orders something that interacts. Neither A nor B will notice the conflict. |  |  |
| 4. Disk crash | Periodic backup or RAID disks. |  |
| 5. Fire and sabotage | Remote backup at least 10 km away … |  |
| 6. Disc full | Capacity management. |  |
| 6p. **Problem.** The system operator doesn't store the data properly, as an example stores the backup data on the same drive as the database. Doesn’t detect that the disk is running full. Often observed for subcontractors. | The main contractor regularly audits whether it is done properly.  Or: The customer gets a weekly backup of all his data for his own storage. |  |

## 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 indicate the test coverage. |  |
| 2. Check all data entered, 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. |  |  |

## Privacy requirements

The customer must meet the European Union’s privacy rules (GDPR, General Data Protection Regulation). The customer has little idea what is involved, but expects that the supplier knows and provides the necessary functionality and advice.

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Provide functionality that enables the customer to meet GDPR, e.g. deletion of a client’s personal data on request, sending personal data in electronic form on request, and automa­tically deleting personal data when they have served their purpose. |  |  |
| 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. |  |
| 3. The customer doesn’t know what to do about GDPR. | The supplier advices the customer about administrative measures to take. |  |
| 4. When the supplier operates the system, and a security breach occurs, the supplier informs the controllers, and in cooperation with the customer, informs the persons concerned. |  |  |

Patients have the right to decide who will see their diagnoses and other clinical data. This is done during the clinical sessions C10 and C20.

| Subtasks in C10 and C20: | Example solutions: | Code: |
| --- | --- | --- |
| 5. Ask the patient for permission to see his clinical data from other organizations, e.g. the patient’s general practitioner. Record the permission. |  |  |
| 6. The system shows only data that the patient has permitted. See also access rights in H1, requirement note. |  |  |

## Protection against threats

Alternative 1:

A risk assessment has shown that the following threats are the most serious. The system must protect against them.

| Protect against: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Unauthorized persons obtaining manager rights through the internet (hacking). | The rights can only be used on the internal network. |  |
| 2. Wire-tapping of passwords or data. | Encryption. |  |
| 3. An intruder tries all possible passwords with a special program. | Passwords must be at least 9 characters, Caps as well as …  Or: at least 5 seconds between login attempts.  Or: Block access after 3 attempts. |  |
| 4. 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). |  |  |
| 5. DoS attack (Denial of Service). An attacker sends so many requests to the system that it is paralyzed. |  |  |
| 6. Unauthorized persons getting access to personal data. Too open-ended, see the guide booklet. |  |  |
| 7. The system meets ISO 15408 (Common Criteria) and ISO 17799. Okay, but check that all risks are covered. See the guide booklet. |  |  |
| 8. The supplier follows developments in the security area and delivers safeguards. |  |  |
| . . . |  |  |

Alternative 2:

The customer has not made a security risk assessment.

| Threat protection: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier shall list the threats that are most serious for this kind of system and specify the safeguards he proposes. |  |  |

# Usability and design

## Ease-of-learning and task efficiency

Although the system has a finished user interface, it may turn out to give the users considerable trouble in some places. The customer wants to avoid the situation where the supplier rejects the problem with reference to that the customer has approved the system or it being a COTS system.

| Requirements for handling usability problems: | Example solutions: | Code: |
| --- | --- | --- |
| 1. 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:

1. is unable to complete the task on his own,
2. or believes it is completed when it is not,
3. or complains that *it is really cumbersome*,
4. 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.

Relevant when essential parts of the user interface have to be developed:

It is important that the system obtains adequate usability. This is best done through early, iterative design and usability test of the user interface (before any programming). It takes place in two stages: Essential parts of the user interface at the early proof (POC), and the rest early in the main development.

If the parties cannot agree on the detailed requirements, they may terminate the contract (section B3-2).

| Requirements for early proof of concept (POC): | Example solutions: | Code: |
| --- | --- | --- |
| 2. The parties must test essential parts of the user interface for usability soon after signing the contract. The critical usability problems (see the requirement note above) must be corrected until usability testing gives acceptable results. In addition, the parties must agree on the detailed usability requirements for later use. | Usability testing (think-aloud testing) is carried out for existing parts of the system in a suitable setup. For parts that don't exist yet, essential parts of the user interface are designed and usability-tested (with prototypes). Three new users participate in each round of testing. |  |

| Examples of requirements to be agreed in detail during POC, and verified before delivery: |  |  |
| --- | --- | --- |
| 3. After a short instruction by super users, the ordinary users must be able to carry out all tasks in Chapter C within their own work areas with few critical usability problems. | Within each work area, thinking-aloud testing is done with three randomly selected users.  Either: A maximum of \_\_ critical usability problems may be observed.  Or: All essential usability problems are handled as system defects. |  |
| 4. Error messages must be understandable and helpful. | During the usability test, a selection of error messages is shown to the user, who tries to explain what the message means and what to do about it.  \_\_% of the explanations must be acceptable. |  |
| 5. It must be possible to operate the system with keyboard only. Users must be able to learn it on their own. | Late in the usability test, the user is asked to use keyboard only. \_\_% of the users must be able to do so. |  |
| 6. Super users must be able to learn the system quickly so they can train other users (cf. J2-1). | Training of a super user takes \_\_\_ days. (The customer expects 3 days). |  |
| 7. A user who has used the system for a week, must be able to quickly order 5 services for a patient, e.g. lab test, scanning … | A typical user is able to order these 5 services in \_\_ minutes. |  |

**Requirement note: Test tasks**

A good test task corresponds to something a real user would do. It must be presented in such a way that it doesn't guide the user. Here is a good and a bad example:

**Test task 1 (good): Order medicine:** The patient complains about pain. Use the system to treat the problem.

(When the user carries out the task, notice whether he checks the existing medication situation before he orders something.)

**Test task 2 (bad - guides the user): Order medicine:** The patient complains about pain. Enter the patient ID and choose the medication screen. Look at the other medications and decide what to order. Close the medication screen and select the order screen …

## Accessibility and Look-and-Feel

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The user interface must follow the MS-Windows guidelines, which most users are familiar with. |  |  |
| 2. Web pages must be suited for screen readers, scaling for visually impaired users, and utilizing the full screen size on small as well as large screens. | The pages follow the HTML guidelines for Accessibility (WCAG 2.1 from W3C). |  |
| 3. The user interface must be in Danish. The pages with opening hours, phone numbers, and addresses must be available in Danish, English, Turkish, and Urdu. |  |  |
| 4. The customer is able to define help texts – also after delivery. | Simple popup texts that the customer can create and change. |  |

# Other requirements and deliverables

## Other standards to obey

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The system must follow the law on accounting 2017. | The supplier obtains the necessary auditor approval or certification. |  |
| 2. … |  |  |

## User training

The customer wants to deliver a large part of the training himself. The idea is to train super users first and then let them train others.

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier must train 50 super users, making them able to train other users. The training must enable the super users to carry out all tasks in Chapter C, including variants, within their own work areas. | Training of a super user takes \_\_ days. (The customer expects 3 days). |  |
| 2. The supplier must train 10 IT staff, making them able to handle the customer's part of system operation and support. | Training of IT staff takes \_\_\_ days. (The customer expects 10 days). |  |
| 3. The training must be carried out within the last month before system delivery in order that users and IT staff can use the system immediately and haven't forgotten what they learned. If necessary, the training must be repeated and the delivery delayed. |  |  |
| 4. … |  |  |

## Documentation

The customer expects that only super users, IT support staff, and systems developers will read the documentation. Thus there is no need for beginner's documentation, except for course material.

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Before system delivery, course material must be available for super users to use when teaching other users. (The customer contributes with documentation of the future work processes, see K4-12.) |  |  |
| 2. A month after system delivery, user-oriented documentation of all system functions must be available. The documentation must be suited for super users. |  |  |
| 3. Before system delivery, sufficient documentation must be available for the customer to handle his part of IT operation and support. |  |  |
| 4. For specially developed software and technical interfaces for third-party development, sufficient documentation for further development must be available two months after system delivery. |  |  |
| 5. Documentation of all tables and fields in a way that third-party developers understand, must be available two months after system delivery. |  |  |
| 6. All documentation must be delivered in electronic form. The customer may freely modify it and copy it for his own use. |  |  |

## Data conversion

| The supplier must convert the following data from the existing systems: | Example solutions: | Code: |
| --- | --- | --- |
| 1. Those data from the patient management system that the EHR system will handle in the future. The format is described in … |  |  |
| 2. Those data from the old EHR system that the EHR system will handle in the future. Data must be transferred through IBM 3270 emulation. See the screen format in … |  |  |
| 3. All converted data must be validated. | The supplier is asked to describe how. |  |
| 4. … |  |  |

## Installation

| Requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The supplier must install all parts of the delivery, hardware as well as software. |  |  |
| 2. The supplier must install all converted data. |  |  |
| 3. … |  |  |

## Testing the system

| Requirements for the supplier's test: | Example solutions: | Code: |
| --- | --- | --- |
| 1. The customer wants to audit which tests the supplier makes and how well they cover. | The supplier makes his test cases and test methods available to the customer. |  |
| 2. There is a need for repeating large parts of the tests after changes. | The supplier uses regression testing. |  |

| Requirements for the customer's own testing: | Example solutions: | Code: |
| --- | --- | --- |
| 3. The customer needs to test the system before accepting the delivery. | The supplier makes a test version available to the customer. |  |
| 4. Special situations must be tested. | The customer can insert special test data. |  |
| 5. 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. |  |

## 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. J3-5. |  |  |
| 4. The supplier must loyally assist with phasing out the system and transferring it to another supplier. |  |  |
| 5. The supplier must carry out the work at a fair price that covers time and material. |  |  |

# The acquisition process

## 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. They are a solution example. Many activities may be done concurrently, e.g. numbers 7 to 11. The plan contains many tests, but the experience is that even with great care, there are surprises at deployment. For this reason it is important for the customer to deploy gradually, e.g. one specialty at a time, learn from experience and gradually deploy more specialties.

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 dialog. | v | v | v |  | N/A |
| 2. Revise requirements and send for tender. |  | v |  |  | -80 |
| 3. Write and send proposal. |  | (v) | v |  | -40 |
| 4. Supplier selection. | v | v | v |  | -20 |
| 5. Signing the contract. |  | v | v |  | 0 |
| 6. POC (Proof-Of-Concept, see B3). | v | v | v |  |  |
| 7. Design and usability-test of user interface (see req. note). | v | v | v |  |  |
| 8. Development. |  | v | v |  |  |
| 9. Data conversion (see J4). |  | v | v |  |  |
| 10. Integration with other systems (see Chapter F). |  | v | v | v |  |
| 11. Documentation (see J3). | v | v | v |  |  |
| 12. Installation test (see req. note below). |  | v | v |  |  |
| 13. System test (see req. note below). |  | v | v |  |  |
| 14. Deployment test (see req. note below). | v | v | v |  |  |
| 15. Training super-users who can train others (see J2). | v | v | v |  |  |
| 16. Training the customer's supporters. |  | v | v |  |  |
| 17. Pilot test for one specialty. Observe actual usage. | v | v | v |  |  |
| 18. Gradual adaptation and deployment of the other specialties. | v | v | v |  |  |
| 19. Operational test (see Chapter L and req. note below). |  | v | v |  |  |
| 20. Evaluate the business results. | v | v |  |  |  |
| 21. Warranty period (one year, see req. note below) |  | v | v |  |  |
| 22. Operation and maintenance (see Chapter L). |  | v | v |  |  |

**Requirement notes**

**Market dialog:** As part of the dialog, 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 just presents his own solution, 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.

**Design of user interface:** Experience shows that usability problems must be detected and corrected before programming. Later it is too expensive to correct problems that need program changes. The remedy is to make early screen prototypes, usability-test them and repeat until the result is acceptable. Some of this can be done at POC, the rest is made early during development. Experience shows that it speeds up development significantly. See also requirement I1.

**Installation test:** The purpose of the installation test is to ensure that the system has basic functionality for carrying out the following tests.

**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).

**Deployment test**: The purpose of the deployment test is to check that the product can work satisfactorily in daily operation with production data and real users. At this point you will normally not carry out real work with the system.

**Pilot test**: A pilot test is real work being done, but with a small number of users and/or limited functionality. You can better overcome helping a few users and then improve the process for the next users. The customer wants the supplier's experts to observe the users and see whether they use the system as planned. The parties assess how much support the users need, and scale up to full operation.

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

**Operational test**: Starts after acceptance test 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).

## 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 can 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. In particular, it is important to make sub-activities during the main development visible. |  |
| 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. Maybe update the issue list. |  |  |
| 7. Can we approve the tests? Can we deploy the system, maybe as a partial delivery? (See the requirement note below). |  |  |
| 8. In case of serious troubles: What can we do? Have we found the root cause? Replan? Change staffing? Change organization and decision authority? Get help from outside? Increase funding? Change scope? Close the project? |  |  |
| 9. Is the business case still valid? |  |  |
| 10. 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 delivery payment is reduced accordingly.

## Update issue list

Large projects may have more than thousand issues, requests for change as well as defects, i.e. something 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 and decide what to do.

**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. Analyse 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 request for change (the customer has to pay)? This may be hard to determine (see the requirement note below). For minor changes, you can make the change and postpone the decision on payment 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?**

1. 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.
2. When the system doesn't meet the requirements or the proposed solution description, it is also a defect and the supplier covers the costs.
3. 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.
4. 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.
5. In other cases it is a request for change and the customer has to pay.

## Workplace and the customer's deliverables

| Workplace: | 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 Chapters F and G). The equipment must be available when the installation test starts. |  | N/A |
| 3. Office with three IT work places from one month before the planned installation test to one month after system delivery. |  | N/A |
| 4. Samples of production data for testing purposes and the full data set for conversion. |  | N/A |
| 5. Test cases for deployment 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 and a half-time secretary. |  | N/A |
| 9. Super users/instructors who learn the system in order to train ordinary users. |  | N/A |
| 10. IT staff who will support the system at the customer site. |  |  |
| 11. Expertise for validation of converted data. |  | N/A |
| 12. Contribution to the course material on future work processes (cf. J3-1). |  | N/A |
| 13. Rights to integrate with the systems mentioned in Chapter F and get the support mentioned. |  |  |

# 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 deployment test. The full verification takes place later, at the operational test. Some of the requirements are only relevant when the supplier is operating the system, others only when he has support responsibility, etc. See the guide booklet.

## Response times

It is important that response is so fast that users are not delayed. Response time is particularly important during the busiest hours, the **peak load** periods, which are morning 8-11 and …

The system must be able handle the load specified below, with the response time specified in the table. The figures are the **nominal** load, i.e. the supplier is not responsible for response time if the actual load exceeds the nominal load.

**Nominal load**

Alternative 1: Suitable when the supplier is familiar with the work area.

1. 2000 users work with clinical session (C10)

2. 1000 users work with administrative tasks (C1 to C4)

3. 100 external users look at opening hours, maps, etc.

Alternative 2: Suitable when the supplier is not familiar with the work area.

The system must be able to handle this concurrently:

1. Simple queries in clinical sessions (C10): 10 per second.

2. Updates in clinical sessions (C10): 2 per second.

3. Public web access: 5 pages loaded per second.

4. …

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 or a mouse click. All measurements are made in a **sample period** of 60 seconds. If for instance, the number of simple queries in the sample period exceeds 10 \* 60 or the number of updates exceeds 2 \*60, the nominal load is exceeded and the sample period is discarded. The transactions must be randomly distributed in the sample period. E.g., the 600 queries of the period may not start within the same second.

**Production work:** Measurements are made with a setup according to Chapter G.

**The public web part:** Measurements are made on a PC connected to the Internet through a 20 Mbit down/2 Mbit up connection with low traffic on the route to the servers.

| Response time requirements: | Example solutions: | Code: |
| --- | --- | --- |
| 1. **Fractile.** The times specified below must apply in almost all cases. | In any sample period, \_\_% of the response times must be within the limits. (The customer expects 98%.) |  |
| 2. Response time measurements must be made regularly in the peak load periods. | Measurements are made once a week with a stop watch.  Or: The system measures all the time. |  |
| 3. 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 0.2 s.) |  |
| 4. 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.) |  |
| 5. 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.) |  |
| 6. Reports used frequently must be visible within the mental switching time. | The report must be visible within \_\_\_ s. (The customer expects 1.3 s.) |  |
| 7. Reports used occasionally must be visible before the user loses patience. | The report must be visible within \_\_\_ s. (The customer expects 20 s.) |  |
| 8. 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 or better.) |  |
| 9. 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.) |  |

## 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.

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.

When the supplier is not responsible for operations, only breakdowns with cause 3 are included in the availability statements. When the supplier is responsible for operations too, he is also responsible for causes 2, 4, and 5.

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%) |  |

## 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 with the response times specified in L1. |  |  |
| 2. MR scans and … are kept for 60 days only. |  |  |
| 3. The system must give access to archived data for the last 20 years with response times as for occasionally used reports (L1-7). |  |  |

## Support

Support comprises help to users, configuration changes, and monitoring of the operation. In this chapter, "supplier" means the supplier's operational organization. A "supporter" means a qualified supplier employee. The support covers all hardware and software delivered under this contract.

Super users are the ordinary user's first point of contact. The supplier has to help only when the super users cannot remedy the problem.

| 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: Even super 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 in person. | 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 sends a supporter when this is necessary to remedy the problem. |  |  |
| 7. The supplier can perform remote diagnostics to remedy the problem. |  |  |
| 8. The supplier monitors request handling to see that requests are closed and response times met. |  |  |
| 9. The supplier records data for computation of support response time, and identification and prevention of frequent problems. | The supplier keeps a log of all steps in the request handling and the cause of the problem. |  |
| 10. The supplier monitors the operation in order to foresee availability problems, and changes the technical configuration so that availability is maintained. |  |  |
| 11. Customer and supplier meet regularly to review response times and discuss prevention of problems. | The parties meet every \_\_ month.  (The customer expects monthly meetings.) |  |

**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).

1. 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.
2. Change configuration: E.g. start servers, change settings, replace printer cartridges, install software. Reply to the user when it has been done.
3. Order equipment or help from another organization: Reply to the user about the expected delay.
4. Defect: The support organization cannot solve the problem. Report it to the maintenance organization. Reply to the user that it has been done.
5. 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.
6. 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.

## Maintenance

Maintenance includes defect removal, system updates and system changes.

| 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 as well as change requests. |  |  |
| 3. For all reported defects, the supplier quickly decides whether the defect is business critical, possible to circumvent temporarily, or possible to circumvent permanently (i.e. reject).  Alternative 1: The local super user decides.  Alternative 2: The customer's IT department decides. | 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.) |  |
| 5. Customer and supplier meet regularly to check the defect assessments, and to decide what to rectify or change, and what it will cost. | The parties meet every \_\_ months.  (The customer expects monthly meetings.) |  |

| Requirements for system improvement: | 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.) |  |
| 7. Within the duration of the contract, the supplier shall on request give quotations for changes  Alternative 1: based on estimated work hours.  Alternative 2: based on a fixed price per Function Point. Disagreement on the Function Point calculation must be resolved by …  Alternative 3: Third-party authorized by the customer makes the changes. |  |  |