Guide to CMS Requirements Specification

Choose the right CMS

M.Sc. Thesis by
Jan Keller Pedersen & Miriam Tang

jkp@cusix.dk miriamtang@itu.dk
http://www.cusix.dk/cmsguide

August 2007, IT University of Copenhagen

Table of Contents

1 How to assess and select a CMS

More and more organisations are facing the process of a CMS acquisition, and many organisations find themselves on shaky ground, because they do not know where to begin or what to expect of a CMS. Even people with knowledge of CMS's, might use a lot of resources on clarifying needs, requirements and not least on structuring them in a way, that makes the requirements manageable to the organisation as well as to consultants selling CMS solutions.

This document contains a CMS requirements specification template and a guide that will help identifying needs and requirements and convert them into a requirements specification that solution providers can use to respond with uniform replies, making the comparison of proposals easier. Thereby time and resources can be released for IT governance and communication strategies that are important factors as well, but are outside the scope of this CMS Requirements Template.

1.1 Why a CMS

Websites have gone from simple business card style static HTML to dynamic extensions of a company's image. A website is much more now than ever before. A website can be critical to attracting and keeping customers as well as it can *be* the business for a company.

The task of managing a website has grown in size as well, going from an IT supporter with flair for HTML managing a handful of pages to dozens or hundreds of non-IT practitioners adding, editing, deleting and arranging content on multiple domains simultaneously.

It is therefore no surprise that the demands for systems to manage the web content have grown, too. Today, content management systems (CMS's) can be as vital to a company's web strategy as the ERP system is to the internal running and management of the company itself.

1.1.1 CMS distinction

CMS's can be categorised on many parameters and groupings. We have chosen a distinction based on functionality and flexibility, visualised in Illustration 1, where a curve describes the typical range of CMS's.

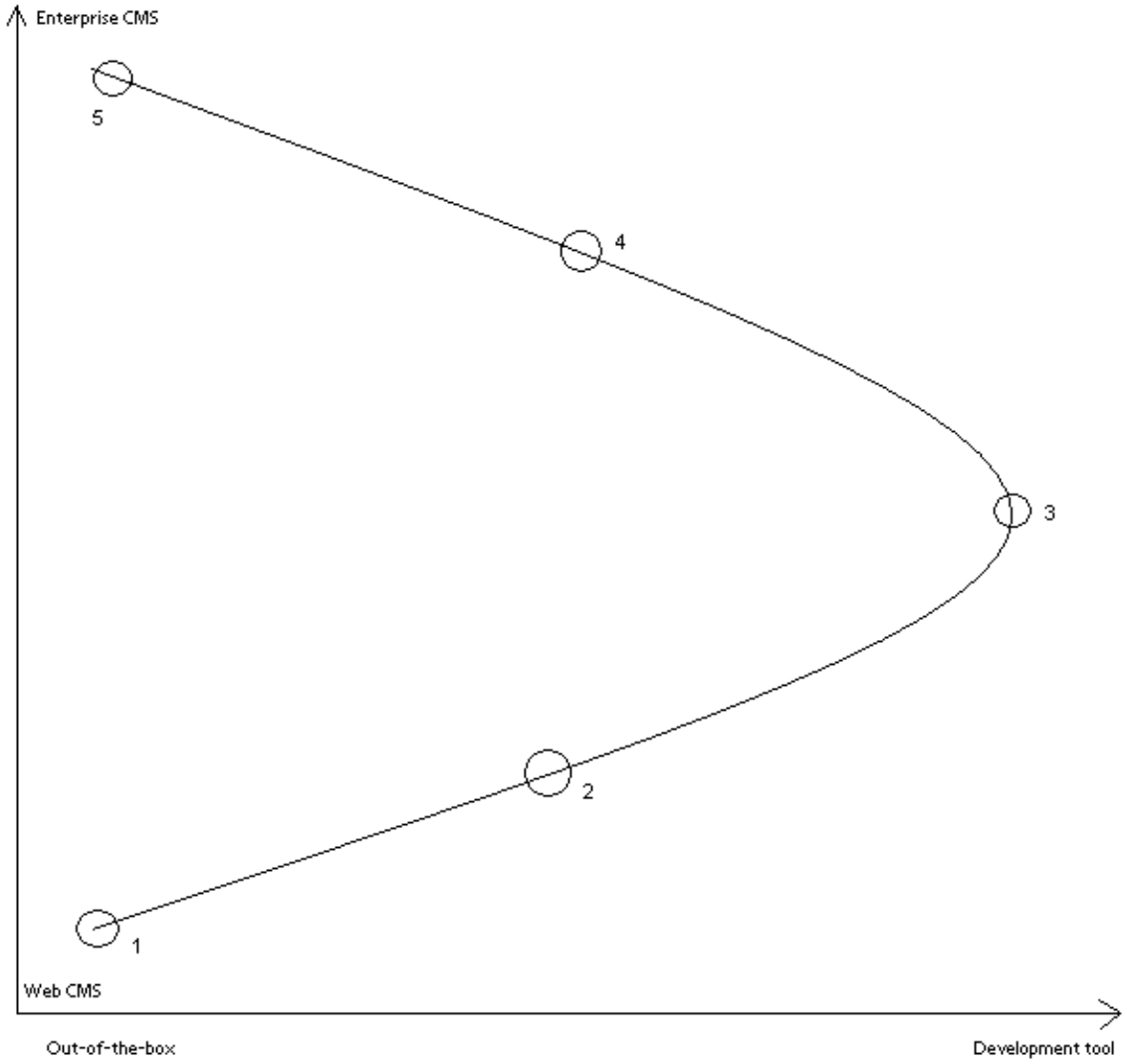

Illustration 1: Basic CMS categories

The illustration has five points of interest:

1. Out-of-the-box Web CMS – Able to manage web page content within certain limits imposed by the system. Few ways for the customer or a third party to customise the system beyond changing the frontend design.
2. Flexible Web CMS – Content is still web pages within certain limits, but the customisability and extendibility is increased. Customer or a third party can make extensive changes to the behaviour of the system.
3. Development tool – More of a platform to build on than a ready-to-use system. The customer or a third party will have few if any restrictions on how to use the system and instead the system will facilitate common functionality through standard libraries. However, practically nothing is working out-of-the-box and will have to be defined or developed by the CMS developers or an experienced third party, meaning that the customer relies heavily on their knowledge and skill with the CMS.

4. Flexible Enterprise CMS – Content is no longer web specific, but can be any content from any system, published through the CMS and made available to internal as well as external viewers based on the access rules defined in the CMS. As with the Flexible Web CMS, a wide range of possibilities exists for modifying and customising the system.
5. Out-of-the-box Enterprise CMS – More specific functionality than the Flexible Enterprise CMS and more functionality directly upon installation and configuration. This type of CMS aims at supplying everything needed by the customers so customisations are unnecessary.

One type of CMS is not better than the other. Which CMS is the right one for a customer depends on the customer's needs. Generally speaking, the closer the CMS is to a development tool, the more flexible is the system and the higher are the costs of customisations and modifications will be in the implementation process.

1.2 Basis of the report

The report is based on investigations on common needs and experiences collected from ten CMS costumer organisations, five CMS consultants and one CMS developers, with the aim of supporting the decision making in customer companies. The report helps defining customer needs and finding the right CMS to fulfil them in the process of (re-)defining their websites.

The result is a template for CMS requirements specifications that can be used by companies or organisations with a frequently updated website.

1.3 How to use this document

This document is a guide of how to customise the CMS Requirements Template into a specific requirements specification for a specific CMS acquisition project.

For each requirement in the template, it should be considered, whether it is a requirement that is relevant to the organisation or not. If relevant, the requirement can be used as-is or refined with further details or by filling in the solution example columns.

Once the requirements specification is finished, it can be sent to candidate suppliers to fill in the solution parts and their replies can be evaluated and compared.

1.4 Send requirements specification

With a requirements specification finished with all left-hand side columns filled and some right-hand side column solution examples, it is time to send it to a range of suppliers and evaluate the replies.

Screening candidate suppliers can be done in several ways:

Scenario	I. Start by selecting suppliers	II. Start by selecting systems	III. EU tender
Step 1	Find competent suppliers	Find interesting systems	Call for tenders
Step 2	Send requirements specification (letting the suppliers select the system, they will use)	Find competent suppliers	Send requirements specification to interested suppliers
Step 3	Get presentations in order to test pre-contract proof of concept (see chapter B2-1)	Get presentations in order to test pre-contract proof of concept (see chapter B2-1)	Receive replies
Step 4	Evaluate replies	Evaluate systems	Evaluate suppliers
Step 5		Select (several) suppliers	Evaluate replies
Step 6		Send requirements specification	
Step 7		Evaluate replies	

Table 1:Steps during selection

It is recommended that at least three suppliers receive the requirements specification, preferably more and definitely with competences within different systems. Especially in the first case the specification should be sent to a higher number of candidate suppliers, whereas case II, where the initial system evaluation has been performed beforehand, three can be a reasonable number. EU tenders have a minimum requirement of three candidates.

Whichever case is chosen, be sure to communicate what is expected of the supplier. It is also recommended to inform the potential suppliers that the same request is sent to other suppliers, so the suppliers know what and how many they are up against.

1.5 Evaluate replies

When the replies come from the suppliers it is time to evaluate the proposals and select the best one. Implementing a CMS is a partnership and the customer's work is not completed after the selection process. Therefore, it is very important that there is a good working relationship between the supplier and the customer, so take your time getting to know the suppliers before committing to one supplier.

Since the acquisition process is about a COTS (Commercial Off-The-Shelf) system, it is fair to assume that much of the required functionality will be available without any further development. This means that some, if not all, of the proof of concept requirements can be validated by performing usability tests according to the tasks in chapter C and receiving a hands-on demonstration from the supplier before making the final selection and signing any contracts.

This demonstration will

- give a preliminary indication of usability and of how much development is needed to fulfil the requirements.
- bring you face to face with the supplier.
- be a very good opportunity to make future daily users of the system perform as many of the tasks in chapter C as the system allows and evaluate the usability.

The next chapter 'Manual for CMS requirements template' consists of sub chapters A through L, explaining the different parts of the requirements template, and is followed by a glossary.

2 Manual for CMS requirements template

The manual for the CMS requirements template consists of the directions and advice on how to use the template. The chapters in the manual follow the chapters in the template.

The template and this guide can be downloaded from http://www.itu.dk/people/jankp/CMS.

Some requirements and solution examples are in *italic*, meaning that it is purely an example for illustrating purposes and that it is highly unlikely to be usable directly.

A Guidance to the supplier

This chapter gives information to the supplier on how to read the requirements and is copied in full from the corresponding chapter in Requirements Template SL-07 (© Søren Lauesen, 2007)

A Guidance to the supplier

This chapter explains the requirements format and how the supplier describes his solution.
The requirements are organised in chapters according to their kind, e.g. Chapter C about the user tasks to be supported. Within each chapter, the requirements are written in tables, e.g. a table with requirements relating to a specific task. The tables have three columns. The left column describes the customer's demand, the middle column an example of a solution or the proposed solution. The right column specifies when the solution is delivered and whether it will be part of a COTS system.

A.1 Functional requirements - constructed example

Here is a constructed example without relation to the present delivery. The purpose is only to illustrate the requirements format. The high-level requirement is that the system must support a number of tasks, including C5, and preferably eliminate the current problems.

C.5 Handle a request

. . .

Users: Call-center staff, often temporarily employed.
Environment: Open-plan, noisy office where users sit close to each other.
Frequency: In total around 500 calls per day, and a maximum of 100 calls per user.

. . .

Demand:	Solution:	Code
1. Receive the call through mail, phone or email. It may be a new request or information about an existing request.		
2. Find the request file . . .	In case of an email call, the system automatically transfers data from the email to the search screen.	
2a. Create a new request file.		
2p. Problem: The request file may be hard to locate, e.g. because the caller doesn't know the request ID or cannot remember his personal ID.	The system shows possible matches with the caller's name or parts of it.	
. . .		

The table lists the subtasks of task C5. There is thus a requirement to support each of the subtasks to some extent. The information about *Users*, *Environment* and *Frequency* is not in the table, so it is not requirements, but assumptions behind the requirements. It means that the task must be supported for this kind of users, environment, etc. (The real requirements about for instance response time, are stated in Chapter L).
The requirements are enumerated. Variants of a requirement are marked with letters a, b, etc. Problems relating to a requirement are marked with the letters p, q, etc. As a consequence, a cross reference to a requirement, a variant, or a problem will look like this:

> See problem C5-2p.

Demand. The left-hand column of the table specifies the customer's demand, e.g. a subtask the system must support, or some data the system must store.
Solution. The middle column specifies the system's support of the demand. In a request for tender, the column shows the customer's current imagination of a solution. This is *not a requirement* or a wish, but only a possible solution to help the supplier understand the demand. I many cases the field will be empty. In the reply, the supplier will fill in the solution he proposes (see section A.3).
Code. The supplier fills in the right-hand column with a code that specifies whether the proposed solution is part of a COTS system, an addition, etc. (see section A.3). Sometimes it is meaningless to specify a code, and then the customer has written N/A in the field.

A.2 Quality requirements - constructed example

Some requirements specify a quality where the customer doesn't want functionality but an amount, a time limit or the like. Here is a constructed example:
G1. Use of existing hardware and software
Currently the customer has the following IT equipment intended for operation of the system:
1. 20 servers of type . . .
2. . . .

In future peak load periods, the equipment will run other applications at the same time as follows:
3. Servers . . .
4. . . .

Platform requirements:	Example solutions:	Code
1. The proposed system must initially run on the existing equipment and meet the response time requirements in L1.	Under these conditions, the system can support ___ users. (The customer expects 20 users).	
2. . . .		

The customer still states his demand at the left, but now the middle column specifies how he wants to measure or structure the reply. In addition, he may state what he expects, i.e. when the solution is fully sufficient. However, the customer may accept something less than expected, but will then score the solution lower on this point.
The introduction to the table specifies the assumptions the supplier can make.

A.3 The supplier's reply

In the reply, the supplier fills in the middle and right-hand columns to specify the solution he proposes. He may show alternative solutions and specify deliveries in several stages or releases. Here is a possible reply to the first example above:

C5. Handle a request

. . .

Subtasks and variants:	Proposed solutions:	Code
1. Receive the call through mail, phone or email. It may be a new request or information about an existing request.	(The system doesn't catch email, etc. The user must initiate the registration.)	5
2. Find the request file . . .	See search screen 12 in App. x.	1
	~~In case of an email call, the system automatically transfers data from the email to the search screen.~~ Release 18 provides a semi-automatic transfer from email. See the description in App. x, page y.	4.18
2a. Create a new request.	See screen 13 in App. x.	1
2p. Problem: The request file may be hard to locate, e.g. because the caller doesn't know the request ID or cannot remember his personal ID.	The system shows possible matches with the caller's name or parts of it. The system provides phonetic search (see search screen 12).	2.1 (alt.A) 2.3 (alt.B)
. . .		

The supplier has changed the heading of the middle column to *proposed solution,* and he has crossed out the example solutions that are not relevant anymore.

Codes

The right-hand column uses the following codes:

1	Part of a COTS system.
2.x	An extension of a COTS system, but the extension is covered by the ordinary maintenance agreement. Will be available from delivery stage x.
3.x	Custom-made software or an extension of a COTS system that is *not* covered by the ordinary maintenance agreement. Will be available from delivery stage x.
4.y	Part of a future release that will be supplied under the ordinary maintenance agreement. Will be available from release y.
5	No solution is offered for this requirement.
alt.z	Alternative solutions are offered. This solution is part of alternative z.

In the example, the supplier has stated that subtask 1 is not supported by the system. The customer has to do it as today (code 5).

The solution for subtask 2 consists of two parts, and the supplier has split the reply accordingly. Part 1 is the COTS solution shown on screen 12 of Appendix x in the proposal (code 1). Part 2 is a feature that automatically analyses an email and transfers the data to the search screen. It is described in Appendix x of the proposal and will be delivered as part of release 18 (code 4). The customer's sample solution isn't relevant any more and has been crossed out.

Variant 2a is supported through screen 13 (code 1: part of COTS). In principle, the supplier doesn't need to describe a solution, but can do with specifying code 1. However, experience shows that this makes the customer feel uneasy. Does the supplier understand our demand at all? This kind of doubt significantly reduces the supplier's chance of winning.

Problem 2p illustrates one more complexity of the proposal. The supplier offers two alternatives that the customer can choose between. Alternative A has a traditional search feature that can be delivered in stage 1. Alternative B has advanced phonetic search that can be delivered in stage 3. Both features are covered by ordinary maintenance (code 2).

The customer may find it very hard to understand a proposal with several alternatives, stages and releases. It is strongly suggested that the supplier submits an overview of the alternatives, the stages involved for each, and the releases. It is convenient if the prices are shown in the same place.
Here is a possible reply to the second example above:
G1. Use of existing hardware and software
. . .

Platform requirements:	Proposed solutions:	Code
1. The proposed system must initially run on the existing equipment and meet the response time requirements in L1.	Under these conditions, the system can support 10 users. (The customer expects 20 users).	1
2. . . .		

The supplier has specified that his solution can support 10 users. Since this is fewer than the customer hoped for, he knows that he will score lower than a competitor who promises to support 20 users. On the other hand, if he had offered 40 rather than 10, this would give him no advantage to the competitor who promised 20.

B High level demands

B.1 Business goals

The business goals are important. Both as a communication tool towards the rest of the organisation (why are we doing this?), towards top management (what will be the benefits for the organisation?) and towards the evaluation process and towards the supplier (how do we know, if the project was a success?). The goals should be quantifiable, when possible, to enable measuring the success of the implementation, as long as the measurement is possible with a reasonable amount of effort.

A goal such as "increase sales by 10%" can be very difficult – if not impossible – to measure, since many factors come into play when sales increase. However, "increase sales *coming through the website* to 20% of total sales within 12 months" can be measured and used for evaluation.

When determining the business goals, requirements to accomplishing the goal will be identified along with requirements to measure the goal. E.g. to increase sales coming through the website to 20% of total sales, clients will have to be able to make purchases via the website, either directly or through contacting a salesperson. To know whether the sales coming through the website has indeed increased to 20% of total sales, a method of deciding whether a sale is coming through the website is required.

1. *Increase sales through the web by 20 % within 12 months*
A common use of a website is to allow customers from anywhere to place orders with the company or to place orders without employee interaction. Therefore, the goal of many CMS implementations is an increased use of the website as a sales channel, increasing revenue from the web.

2. *Increase page views / visits with 20 % within 12 months*
For most organisations, a website is a marketing channel intended for raising awareness of the organisation and its products. Page views and visits are in itself not a revenue-increasing goal. However, with increased page views and increased visits, awareness is raised and the chance of gaining new or retaining current customers increase.

To increase the page views and make visitors come back, the website should contain relevant, up-to-date content and respond quickly to the visitor's browsing. To get the visitors to come in the first place, optimising for search engines can make a large difference.

3. *Distribute website content updates across organisation*
More people updating the content increases the likelihood of the content being updated regularly, making the content more reliable and, since there are fewer steps from the people with know-how to the people updating the content, more correct.

4. *Knowledge sharing. 75% of all procedures available on the intranet within 12 months*
When employees can find and add needed information by themselves, less knowledge will be stored within departments in folders or inside the heads of department employees. This means less time spent on information search and more time spent on information use, increasing productivity.

5. *Reduce user management workload by 30%*
Integrated user management reduces the workload for IT administrative personnel, who can focus on other tasks.

B.1 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.

Purpose with the CMS	**Overall solutions**	**Related requirements**	**Deadline (if any)**
1. Increase sales through the web *by 20% within 12 months*	Automated sales process Consistent design	Sales/sign-up system with statistics (chapter D) Site templates (chapter C.1)	
2. Increase page views / visits by *20% within 12 months*	Relevant data on website Fast response times	Statistics functionality (chapter E.2) Short response times (chapter L.1) Search engine optimisation (chapter E.5)	
3. Distributed website content updates across organisation	Enable non-IT proficient users to add and update content	Usability, Education, Guides (chapters I and J)	
4. Knowledge sharing. *75% of all procedures available on the intranet within 12 months*	Make content easy to publish and easy to find	Easy content publishing (chapter C.4) Search functionality (chapter C.16)	
5. Reduce user management workload *by 30%*	Authentication Security	Directory service integration (chapter F.1)	
6. Reduce repetitive customer service calls *by 25% within 12 months*	Make content available and easy to find on the website	Easy content publishing (chapter C.4) Search functionality (chapter C.16)	
7. Move *35%* of the resources from technical development and operations to production of content	Streamline user and website management	Directory service integration (chapter F.1) Easy expansion of the system (chapter E.4)	
8. Accessibility standards compliance	Enforce accessibility standards	Chapter I.2	
9. Increase update frequency to *5 times a week* on 1st level, *5 times a month* on 2nd level, *6 times per year* on 3rd level from the front page	Easy to use and efficient CMS	High usability (chapter I) Reminders to page responsible, when page has not been updated recently (chapter E.6)	

6. *Reduce repetitive customer service calls by 25% within 12 months*
When customers can find the information they need on the website, fewer calls will be made to customer service, who can focus on providing a better service rather than answer the same questions again and again.
7. *Move 35% of the resources from technical development and operations to production of content*
Technical solutions by themselves do not increase revenues or give a better service. With less resources needed for maintaining the website, more resources can be spent on enhancing the website and its content, supporting the people, who *can* provide better services.

8. *Accessibility standards compliance*
More and more organisations – especially municipal authorities – face requirements from laws, stating that disabled people must be able to access content on their websites. When this is supported by the system, content providers can focus on what to publish instead of how the content should be structured.

9. *Increase update frequency to*
5 times per week on 1st level
5 times per month on 2nd level
6 times per year on 3rd level pages

Often the front page and pages placed high in the site structure, are apt to contain information which needs to be updated more often, than pages on a lower level, which are often more specialised and seldom needs to be changed. The easier it is to update a website, the more likely is it to be updated regularly.

B.2 Early proof of concept

This chapter describes the requirements that are considered to be the most risky, and why an early proof is needed. The goal of the supplier is to convince the customer that the supplier's system can meet the requirements within the allocated time frame.

***1**. After a short instruction by super users to the CMS backend, the ordinary users must be able to carry out all tasks in Chapter C within their own work areas without critical usability problems*
The easier the system is to use, the less resources will be spent on training, re-training and support.
Bringing super users into the evaluation and customisation process will also have the added benefit of involving the super users. Users, who are involved with a system implementation are usually much more inclined to use it properly and explore the possibilities than users, who get a system "pulled down on them" that they are forced to use.

***2**. Response times with the required number of users*
Implementing a system without knowing if it can deliver the required response times would rate high among the biggest errors that can be made. Promises from a supplier should be proven as soon as possible with expected data volumes, so the supplier will know whether additional measures, such as caching solutions or better hardware, are needed.

***3**. Integration with large external systems such as Navision, Siebel or SAP*
If the organisation relies on automatic order processing or lead generation, this integration will be vital. Knowing as early as possible that the connectivity and transfer of data from one system to the other is working, increases the likelihood of the full integration working at the specified time.

B.2 Early proof of concept

The customer wants to avoid the situation where the trivial parts of the system are developed early, while the hard parts are postponed and prove impossible to deliver. To prevent this, the customer requires an early proof of concept.

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

The following requirements are considered high-risk. Substantial deficiencies in these areas can hardly be corrected 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 work days after signing the contract.

Risky requirements where early proof is required	**Example of proof**	**Date possible**
1. After a short instruction by super users to the CMS backend, the ordinary users must be able to carry out all tasks in Chapter C within their own work areas without critical usability problems.	Task in chapter C should be tested as described in chapter I.1 (p. 64) Can be done before contract signing.	
2. Response times with the required number of users (all requirements in chapter L.1 for backend and frontend).	*A test setup with dummy frontend contend is used to simulate the required number of users. The response times are measured. Can be done within ___ working days.*	
3. Integration with large external systems such as Navision, Siebel or SAP	*Proof of connectivity with dummy data.* *Can be done within ___ working days.*	
4. Newsletter subscription process	*Proof of working prototype.* *Can be done within ___ working days.*	
5. Possibility for third-party expansion of the system.	*Documentation of parts of the system and the technical interfaces is studied by an independent software house in order to assess whether it is adequate for expanding the system. Can be done within __ working days.*	
6. LDAP integration with more than one domain	*Working prototype with LDAP authentication of users from 2 domains.* *Can be done within ___ working days.*	
7. Usability of supplier-developed pages, ie. user test of data view and functionality at the frontend of the CMS	*Test of a prototype of the pages data and functionality (maybe a paper mock up) is assessed by expert users.* *Can be done within ___ working days.*	
8. Understandable error messages	*Presentation or listing of error messages, asking future users whether they understand the meaning.* *Can be done within ___ working days.*	

***4**. Newsletter subscription process*
Newsletters can be very important to a company. It is an excellent way of communicating to many customers at once and it is extremely easy for the recipients to forward a newsletter to a friend. However, newsletter subscribers expect easy subscription and unsubscription. This is considered a high risk functionality, because in case the unsubscription process does not work properly, the newsletters sent to unsubscribed email addresses are effectively "spam" mails which can result in substantial fines. Additionally it is bad for the image of the business to send out newsletters to customers, who has stated, that they do not want to receive newsletters.

***5**. Possibility for third-party expansion of the system.*
A website is an organic entity where visitors expect changes and new initiatives on a regular basis. Having the possibility of hiring consultants to expand the system with custom components – or have an IT department do it – will allow you to react quickly to new ideas and lessen your dependency on the supplier.

***6**. LDAP integration with more than one domain*
LDAP integration – should it be a requirement – can make user administration much simpler for the IT department. However, experiences with multiple LDAP domains show that it can be a large task integrating them into one web application. Since the CMS is impossible to manage without the login procedure working, it is important to know with certainty that it will work.

***7**. Usability of supplier-developed pages, ie. user test of data view and functionality at the frontend of the CMS*
Similar to risk 1, usability is very important to test as early as possible. However, since these pages are developed by the supplier in the implementation process, it (usually) cannot be done before the contract is signed, as opposed to risk 1.

***8**. Understandable error messages*
Error messages are only useful, if they make sense to the user. Therefore the understandability of the error messages should be tested and the supplier instructed to change those that cannot be easily understood by the users.

B.3 Implementation time frame

The implementation schedule is important to consider as well. How early can you get a proof of concept? How early can the users get training? When can you expect your new website to be up and running?

Since you are looking at standard systems, it is fair to assume that much of the required functionality exists, although it will usually need adaptation to fit specific needs. Therefore it is also fair to assume that you can get a proof of concept before you have even selected the system and signed the contract. In the final CMS solution you should expect that standard modules can require some customisations to fit in with the customer's desires, even though the module doesn't have to be developed from scratch.

B.4 Selection criteria

To gain overview of the proposal each selection criteria from section B.4 can be given a weight that states the importance of the criteria. In the different proposals, the handling of each criteria should be given a score, stating how well the criteria is met by the suppliers.

B.3 Implementation time frame

The system delivery steps as expected by the customer. Missing steps can be added by the supplier. The various dates should be estimated by the supplier as if the decision to sign the contract is done *two weeks after* the suggested date for proof of concept – pre-contract proof.

Delivery	**Date**
1. Proof of concept – pre-contract proof	*2007-06-31*
2. Proof of concept – remaining proof	
3. Design applied	
4. Test set up – usability and functionality tests	
5. Data conversion and import	
6. Education material	
7. Education of users	
8. System documentation	
9. Production set up – acceptance tests	
10. Launch	*2007-12-31*

B.4 Selection criteria

The customer chooses the solution that is *best from an economic point of view*. Each criteria is given a weight between 1 and 10 to indicate the importance of this criteria. The supplier's solution will be given a score between 0 and 4 for each criteria and this score is multiplied by the weight. Adding the weighted scores for all criteria gives the total score for the solution. Additionally, each criteria has a minimum score assigned that will result in solution disqualification *regardless of the scores in other criteria*, should the solution score less.

Criteria	Weight	Min. score
1. Usability requirements in chapter I and C	*8*	*3*
2. Business value of the solution. Assessed as a number for each of the business goals (B.1).	*9*	*3*
3. The risk. Assessed from the time for the early proof (B.2) and the fraction of the solution that already exists in the CMS.	*4*	*2*
4. Delivery time B.3.	*3*	*1*
5. Price including the customer's costs for new equipment, data conversion, and user training B.5.	*4*	*2*
6. Market robustness, i.e. How safe the solution is in regards to longevity, financial backup, rate of development and similar.	*5*	*2*
7. Extendability. How easy will it be to extend the system by adding modules (E.4)or integrating with other systems (chapter F).	*10*	*3*
8. Viability of the community supporting the CMS	*2*	*1*

Weightings

It is not easy to set the weightings, and it can be done in different ways. The optimal way to set the weights would be on basis of a cost-benefit analysis. In some situations, like when a requirement specification is the basis of an EU tender, it actually is a requirement that the selection criteria have weightings, but this is a large area, on which information can be found at various websites[1]. The weightings are set in proportion to the estimated business value, so there are no upper limitation on the weightings to be set. The potential suppliers should be informed of the selection criteria and their weightings.

Score

When receiving proposals of suppliers on basis of the requirement specification, the proposals should be evaluated one by one. For each of the proposals, the criteria are given scores between 0 and 4, stating how well the proposal handles each criteria, as shown in the table below.

Score	Meaning
0	Not handled
1	Not as good as today / Less than expected
2	The same as today / Little less than expected
3	As expected
4	Better than expected

Table 2: Score

A criteria can also have set a minimum score to disqualify unacceptable proposals no matter their score in other criteria as shown in the table below in the columns with the header 'Min.'. The total score of each proposal can be calculated by summarising the criteria value multiplied with the weight.

		CMS:	proposal X		proposal Y		proposal Z		...	
Selection criteria	**Weight**	**Min.**	**Score**	**Weighted**	**Score**	**Weighted**	**Score**	**Weighted**	**...**	**...**
1. Usability	8	3	4	32	x	x	x	x		
2. Business value	10	4	4	40	x	x	x	x		
3. Risks	6	2	3	18	x	x	x	x		
4. Delivery time	2	1	1	2						
5. Price	3	2	3	9						
6. Market robustness	7	2	2	14						
7. Extendibility	6	2	3	18						
8. Community	1	0	0	0						
Total score:				133						...

Table 3: Evaluation of proposals

1 The EU directive 2004/18/EC is a good term in search engines when looking for information and can be found in full, legal text at http://europa.eu.int/eur-lex/pri/en/oj/dat/2004/l_134/l_13420040430en01140240.pdf

The scores should be used to indicate which proposals are simply not good enough compared to the others. The proposal with the highest total score is not always the best proposal, but it is recommended to select a proposal with a high total score.

Usability

Usability can be measured in the pre-contract signing proof-of-concept phase as described in 1.4 assigning scores depending on the number of issues and their severity.

Business Value

The business value is an estimation of how well the proposal will fulfil the business goals.

Risks

The risk is a measurement of how convinced the customer is on whether the supplier delivers the expected CMS. The earlier the proof-of-concept risks can be proven, the better. Other indicators of low risks could be, that the right side of the tables are filled in with good solutions, it could also be, that the supplier has solved similar problems before as well, as it can be experiences from previous deliveries from the supplier. The more indicators of low risk, the better should the score of the proposal be in this criteria.

Delivery time

The rule of thumb is naturally "the sooner, the better". However, be careful not to apply too high weightings to this criteria, ending up with poor system functionality delivered quickly. One way of evaluating delivery times is to set an expected date for the system launch. Offers delivering around this date gets a score of three – as expected – with earlier offers achieving a score of four and later offers two, one or zero, depending on the difference to the expected date.

Price

If the price of the CMS acquisition is above a certain amount, there must be an EU tender, an open proposal evaluation to avoid corruption and nepotism. This requires objective selection criteria. There are two different ways to choose a proposal in an EU tender.:

1. The 'cheapest proposal' is simply the proposal with the lowest price. If the requirements are met, it will not be taken into account that one of the proposals might have a higher business value.
2. The 'economically most attractive proposal' is the proposal that gives most value per price unit, i.e. the highest coefficient. There might be cheaper proposals, but the focus is the value compared to the amount of money spent, as shown in the coefficient calculation example:

$$\frac{\text{Business value} - \text{Risk}}{\text{Price}} \quad <=> \quad \frac{400{,}000 - 100{,}000}{100{,}000} \quad = \quad \frac{3}{}$$

The risk is estimated as a quantification of the risks involved with the supplier's bid, the amount of functionality that will need to be developed and the amount of functionality that is available for a proof-of-concept session. If the cheapest proposal has to be chosen, the supplier's reservation about requirements, which are described in the requirements specification, can be estimated as a quantification of the risks and added to the price, when finding the cheapest proposal.

EU tendering is a large area, and if the CMS acquisition has to call for a tender, it is important to look further into the area, since the selection criteria must be substantiated in the requirements specification and the type of EU tender chosen beforehand.

Market robustness

When evaluating a candidate system, it must be considered, how stable the organisation behind the system is. This includes both the supplier, the system developer (if the CMS is developed by someone else than the supplier) and the system developer's other partners. Will the system be maintained and updated in the future? Will consultants be available? Is there a customer base large enough, so that the developers most likely have made improvements to the system, based on feedback from suppliers and their customers?

Extendibility

Customer needs do change and it can be extremely difficult to predict, what will be needed of the CMS in the future. Ensuring that the CMS can be extended with new content types, new functionality and new integrations increases the probability that the CMS will fulfil future needs.

Community

Sharing experiences and getting advice from people in a similar situation is a good way to improve the use of a CMS in an organisation. Additionally, if the customer would like to be able to make development on the system themselves, it is important that there is a viable community supporting the product. Does the community have independent contributors or is it only the CMS supplier that makes contributions? How active are mailing lists and web sites? Is there a friendly culture in the community, and are they welcoming newcomers?

B.5 Price

How much you are willing to spend on the website is an important consideration from a quite early point in the process. Integration with other systems as well as development of custom content and design will be consultancy projects that can weigh heavily on the budget after the system acquisition and license payments.

It is always a good idea to get a solution built from modules, where each module can work independently on a basic module. In case the price exceeds the budget at the time, it will still be possible to launch a smaller solution that can be extended later, without having to rebuild the existing solution.

This would also be a good place to ask questions of the supplier regarding what is included in the agreed price.

Can the customer claim that the supplier corrects errors for free, if they are caused by the supplier's code? And if so, for how long?

Can the customer expect that the supplier corrects errors for free, the CMS has them from the beginning?

Who pays in case the supplier with a correction introduces new errors?

Getting the prices specified in as much detail as possible will help you determine the total price and running costs better and lead to fewer costly surprises.

B.5 Price and deliverables

In this list, the supplier shall input the prices for the various deliverables in as much detail as possible as well as he is encouraged to add missing items on the list. The list is divided into one-time costs in the implementation process and running costs after implementation.

One-time costs	**Price**	**Specifications**
1. Start licence		
2. Implementation Documentation User and developer guides to the CMS		
2a. Variant: Hourly rate during implementation		
3. Hardware		
4. Design		

Running costs	**Price**	**Specifications**
5. Maintenance licence		
6. Education		
7. User support		
8. Hourly rate for post-implementation developments		
9. Correction of errors caused by developer updates to the CMS	*0*	*Included in maintenance license*
10. Correction of errors caused by the supplier	*0*	*Included in maintenance license*
11. Correction of errors in the CMS, present at the time of implementation	*0*	*Included in implementation costs*

C Work areas and tasks

The work areas of this chapter contain tasks that are used in the CMS on a regular basis. Most of the functional requirements will be in this chapter. When defining the tasks, you should consider the following:

Chapter C contains CMS requirements which are related to procedures and tasks, that the system must support. The requirements have their starting point in user needs, and the requirements are formulated in a way, that leaves it to the supplier to find out, how to satisfy them. In this way the specification will not restrict the developers or the range of potential systems unnecessarily.

In the following, some of the requirements are explained further, in order to help the customer decide, if the requirements and any related problems are relevant in their situation.

C.1 Manage page templates

Templates are usually used to maintain a certain layout on pages of a site. Templates can be on site-level, where they determine the general layout of items, headers, navigational links etc. or they can be on page-level, where they determine the layout of the page itself.

Most CMS's employ this concept in some form.

C Work areas and tasks

The system must support all user tasks in this chapter, including all subtasks and variants, and mitigate the problems. The subtasks are numbered for reference purposes. They don't have to be carried out in this sequence, and many of them are optional. A subtask may also be repeated during the same task.

C.1 Manage templates

Using templates is a way to control the layout and a tool to create a uniform appearance of all the pages that make up the site, no matter who updates them.
A CMS is often set up in a way, that forces the users to use the templates, and therefore it is important, that there are enough templates to provide flexible presentation of content while keeping the number low enough to retain the overview of available templates and when to use which template.

Start: A page template needs addition, updating or deletion

End: Page template is saved or deleted

Frequency: Rarely once set up

Users: Website administrators, graphical designers

Sub-tasks and variations	Solution example	Code
1. List templates		
1p. It can be difficult to distinguish between the layout of templates, if they only are identified by names	Instead of choosing among the templates from their names only, it might be easier to overview and choose the right template, if they are represented by small rough models of their layout for instance.	
2. Add template		
3. Edit template		
4. Undo changes		
5. Save template		
5a. Variant: Save template with another name/copy template		
6. Delete template		
6p. Problem: Template is in use. If a template is being used by pages, the system should prevent that it is deleted, and if the template is to be replaced by another one, it is useful to be able to see which pages are using the template.	Block deletion and list pages using this template Option to select a replacement template to be applied to all pages using this template	
6q. Problem: Users cannot see which templates are in use before trying to delete them. There may exist numerous templates in the CMS, and some of them may even be exact copies of others, meant to be changed in some way. To be able to clean up among the templates, it has to be clear which are in use and which are not.	List templates with indication of whether it is in use	

C.1 Manage templates (continued)

Sub-tasks and variations	Solution example	Code
7. When new design templates are implemented or new page layout templates are created to replace existing templates, a method to replace the old templates with the new is needed.		
7p. Problem: User can only replace the template one page at the time. Sites can have thousands of pages, and changes to the general layout on only parts of the site do occur. Hence, it should be possible to replace the template of many pages all at once.	Replace template of several pages all at once. For instance by replacing the template from a certain site-node and below or by marking the pages, in a model of the site tree structure, and replace with another template all at once	
8. Replace template for a certain page node and for all pages below. Pages below another page are often related and so is the layout. Therefore, it makes sense to optimise the layout change for a page node and all the pages below it.		
8p. Problem: Pages below a certain page node may have been specially developed or they may use another template that should not be replaced	Replace pages with a specific template only. Pages using other templates will stay unchanged.	

C.2 Add/edit/delete content item

Add, edit and delete content are basic functionalities in every CMS, and every CMS will contain the functionality. The interesting things are therefore the problems that users have experienced when adding, editing or deleting content. These problems are explained below because these are problems that are handled differently, if at all, in different systems.

Most pages will include media files, most often images, to liven up the page and aid the readability.

C.2-*4p Problem: Copy-pasting to the CMS editor does not automatically upload picture to media library*

The fact that media files have to be uploaded to the server to be visible to the public on a website, is an abstraction that some new or rare users of CMS's forget. A user writes some content in a Word document, inserts some pictures and tries to copy and paste text as well as pictures into the CMS editor. This is a intuitive thing to do, but usually CMS's cannot handle the pictures this way. The ideal solution would be a CMS that could handle this, and automatically upload the picture to the media library and change the reference in the text from the local pc to the media library on the server. Another way of handling the problem could be a message telling the user to upload the picture to the media library, if a picture is pasted into the editor.

C.2 Add/edit/delete content item

Work area: CMS backend

Start: Content has to be edited, feedback received, resumption of parked content

End: Content sent for approval, published, or parked for later resumption

Frequency: Daily

Users: Administrators, editors and authors

Sub-tasks and variations	Solution example	Code
1. Find content item. When a user edits or creates a page, there might be a break in the task, and the user will have to save the unpublished content item in order to resume the task the next day, or hand over the task to a colleague. Therefore the users will need an easy way of locating content to resume editing.	List of content items Directory structure with content items	
1p. Problem:A parked content item can be difficult to find	Option to list only unpublished pages Mark unpublished pages with appropriate icon or colour	
2. Edit page/content		
2a. Variant: Create page based on template		
2q. Problem: CMS editor only shows a little part of the page, so the overview of the page in its context is missing	CMS editor should be able to show the entire page and not only the box that is being edited	
2r. Problem: Someone edits a page he should not edit	Permissions should be set up so that only relevant pages are visible to and editable by a user	

C.2 Add/edit/delete content item (continued)

Sub-tasks and variations	Solution example	Code
3. Search among media files in the media library	Meta data for search purposes: Name, text, photographer, date	
4. Insert a media file on the page		
4p. Problem: Copy-pasting (from Word for instance) to the CMS editor does not automatically upload picture to media library	CMS editor, that automatically uploads media file, when copied into the editor and prompts the user for meta data Message, that tells the user to remember to upload a picture	
4a. Variant: Insert other content (see E.3 Dynamic content components)		
5. Upload a media file and set meta data		
5p. Problem: User overwrites file used on pages with an unsuitable image, thinking that it is not in use.	Give a warning that the file is in use and block the overwrite	
6. Add/change meta data to a content item such as: responsible person, creation date, valid from, valid to, keywords	Some of the values could have default values set automatically e.g. creation date and responsible person defaulting to the current user	
7. Delete content		
7p. Problem: Deleted page has sub-pages in the tree, resulting in “orphan” pages without parent pages	Warning about consequences when trying to delete the page, block deletion	
7q. Problem: Deleted page is linked to from other pages. Referencing pages will have a dead link to the deleted page.	Block deletion and show list of referencing pages	
8. Park/Save content for later resumption, i.e. it has to be saved without being published or sent for approval		
8p. Problem: The tree structure in the backend can become pretty deep, and the window that contains the tree structure is refreshed every time the user presses the save button and he is redirected to the top of the tree structure instead of to the folder where he is currently working. Then he has to scroll down and maybe even expand the tree structure again to be able to continue where he left.		
9 Send for approval	Manually by mail, phone or verbally or by an automated workflow functionality	
10. Receive feedback from approver/editor	Manually by mail, phone or verbally or by an automated workflow functionality	

C.2 Add/edit/delete content item (continued)

Sub-tasks and variations	Solution example	Code
11. User should be able to see how the edited content will look on the page without publishing it		
11p. Problem: Preview doesn't look exactly as the page does, when saved. Users do a lot of editing and will have to publish every time they want to see what it really looks like in the page's context.	Publish to staging server with browsing as if it was published Preview only selected page in frontend design as if it was published	
11q. Problem: It is difficult to se how the edited content item is going to look on the page, if the preview does not contain the rest of the page		
12. Undo changes	Roll back to earlier versions Undo in editor	
13. Publish (See long subtask C.4)		

C.3 Review and approve content

Publish content has been separated as a long subtask to C.2 to give a better overview of the process. The subtasks can be called a workflow, which supports and enforces publishing restrictions.

A workflow is a chain of tasks that are carried out by different user roles. An example could be that an author writes an article and an editor approves the article before it gets published. A workflow system can enforce and support these organisational procedures directly in the CMS, thus not allowing for content providers to skip this step.

If the customer wants to enforce content review prior to publishing and/or enforce that only a few people can publish content, this long subtask is relevant. In our experience, though, it is often thought of as needed in the requirements, but rarely used in the finished solution, due to the increase in bureaucracy and the fact that the process could just as easily be manual with content providers communicating directly, asking for a review, when in doubt.

C.3 Review and approve content

Start: Content is ready for review, notification received by reviewer

End: Content reviewed and feedback given to content provider or content approved and published

Frequency: Varies between many times a day to weekly, monthly or yearly

Users: Content editors

Sub-tasks and variations	Solution example	Code
1. Receive notification about content ready for publishing	Manually by mail, phone or verbally or by an automated workflow functionality	
2. Locate content		
2p. Content created or last edited by another editor can be difficult to find	Link in notification email Search functionality List of unpublished content items Well-arranged directory structure, where unpublished content items can be identified easily	
3. Correct content		
4. Publish (See long subtask C.4)		
4p. Problem: It is only possible to publish either one single content item or all unpublished content items at the same time. Sometimes, editors work on several items and publishing content items one by one is too slow, while publishing all items in the queue will publish items not ready for publishing.	Possibility to pick out a number of content items to publish in one go	
4q. Problem: It is time consuming that the CMS is locked with an hourglass while a page is being published. It should be possible to continue the work while a page is being published		
5. Reject content	Manually by mail, phone or verbally or by an automated workflow functionality	
6. Notify user about rejection or publishing of the content	Manually by mail, phone or verbally or by an automated workflow functionality	
6a. Variant: Add a message to the notification		

C.4 Publish content

Long sub-task to C.2 and C.3.

C.5 Manage media library

All websites need at least a few hand-typed pages with text, images and links to other pages or websites. Most CMS's have a repository for media files – images, video files, Word and PDF documents etc. - that can be inserted on a page. This is a basic requirement that must be present, although the specific functionality can vary.

The requirements to media library varies, but if the site contains many pages referring to media files, or if the media library contains a lot of media files, it is recommended to look for a CMS with a good media library.

The media library can resemble a file system with folders and files as in the Composite CMS, but can also contain meta data that will aid when searching for relevant files. Some organisations – mainly those with many content providers – will need the searchable media library whereas others will 'just' need to set up a relevant category/folder structure so users can browse for the relevant files.

C.4 Publish content

Frequency: Varies between many times a day to weekly, monthly or yearly

Sub-tasks and variations	Solution example	Code
1. Set publish date and time		
2. Set un-publish/expiry date and time		
3. Save published content item		
4. Publish to staging server		
5. Publish to production server		

C.5 Manage media library

Work area: CMS backend

Start: User receives/collects one ore more media files such as PDF documents, pictures and videos, to be available on the website

End: Files are uploaded

Users: Administrators, editors and authors

Frequency: Varies between many times a day to weekly, monthly or yearly

Sub-tasks and variations	Solution example	Code
1. Locate media file on local pc	Browse functionality	
2. Upload media file to media library		
2p. Problem: Uploading many files to the media library one by one is cumbersome, since they have to be uploaded individually to set meta data for each file	Multi-file-upload form with fields for meta data	
3. Set meta data	Fields for meta data in file upload form	
3a. Variant: Edit meta data of a media file which is in the media library already		
4. Search among media files in the media library		
5. Show media files in the library		
6. Show which pages refer to a media file		
7. Delete media file		
7p. Problem: There are references to the media file from pages that will either link to an non-existing file (a dead link) or may refer in the page contents to a file that is no longer displayed.	Block deletion and list pages using the file Allow user to replace the file with another file that is subsequently used on the pages instead	

C.6 Link checking

Links referring to pages on external sites, uncontrolled by the organisation, can be outdated if the external page is removed. A periodical link check that checks whether links are still valid, is often used to avoid dead links on the site. Usually a CMS has some functionality, that prevents the site from containing internal dead links (see C1-7q), but if this is not the case, the link check should also include checking of links to internal pages.

C.6 Link checking

Internal links to and from a page are usually checked when the page is saved, but external resources can change independently. Therefore, it can be necessary to periodically check whether these links are still valid.

Work area: CMS backend

Start: User wants to check for dead links or retrieve an earlier link check result

End: Site or sub-site without dead link or list of pages with dead links parked for later resumption

Frequency: Varies between many times a day to weekly, monthly or yearly

Difficult: Many pages with dead links require editing

Users: Editors and authors

Sub-tasks and variations	Solution example	Code
1. Find saved link check result list		
2. Start link check		
2p. Problem: User might only be interested in doing a link check of a particular part of the site. A site can contain several thousands of pages, and a link check of the whole site is very time consuming	Link check on page level or sub tree level of the site structure	
2q. Problem: Link check is too sensitive, and considers links to large files as dead links		
3. It should be easy to find the pages with dead links	Each item in the list has a link to edit the page with a dead link.	
3p. Problem: Link check result list disappears every time the user goes to one of the pages to correct dead links.	Open in new window “Remember” link check result list for when the user returns	
4. Save link check result list for later resumption		

C.7 Manage site structure

Site structures change with time and one or more pages will be moved from one place in the site tree to another. Although it is not a daily recurring task, it can be time consuming if each page has to be moved individually, if references are not updated automatically or if user permissions are not handled automatically.

C.7 Manage site structure

Work area: CMS Backend

Start: Administrator receives information about organisational changes/ major changes to the website.

End: Site structure reorganised

Frequency: Daily/Weekly/yearly depends of the number of sites

Users: Administrator

Sub-tasks and variations	**Solution example**	Code
1. Create sub-site	Sub-sites are first level pages in the page tree below the root	
1p. Problem: It is difficult to get site overview	List sites and sub-pages in a tree hierarchy	
2. Move sub-site	Drag and drop pages and sub-pages from one place in the tree to another	
2p. Problem: References to and from other pages have to be changed to avoid dead links pointing to the sub-site that is now deleted.		
2q. Problem: User permissions might be wrong after a sub-site has been moved to another place in the site structure		
3. Maintain user rights and responsibilities to sub-site (see H.2 Security management)		
4. Notify responsible for pages on a sub-site about the change in site structure		
5. Publish changes		
5p. Problem: User cannot see changes from the visitor's point of view before publishing. With major changes like the movement of a sub-site, navigation menus and other elements affecting the layout may change, too.	Publish to staging server, where changes can be seen in frontend view before publishing to public server or a graphic model of the changed page structure	
6. Edit page responsible/contact information		
6p It can be difficult to make changes in responsible contact information if it has to be done on each page	Contact information stored centrally – perhaps in a separate module – and referenced from the page.	
6q. If a page responsible needs to hand over the responsibilities to another employee it is unproductive if it is done page by page	Update references centrally, replacing with the new employee	
7. Remove sub-site consisting of one or more pages		
7p. Problem: Pages are referenced from other pages	Block deletion and list referencing pages	

C.8 Manage newsletter types

An organisation might have several newsletters that need to be managed. Users should have the possibility of creating new newsletter types as well as changing the layout and name of an existing newsletter.

C.9 Manage newsletter subscribers

Managing subscribers from the CMS backend can be very useful. For instance when a marketing campaign at an exposition yields a (possibly handwritten) list of email addresses of people interested in the organisation.

C.8 Manage newsletter types

A newsletter type is a series of newsletters – e.g. the monthly promotional newsletter or the yearly Christmas newsletter – that will have a number of subscribers and information common to all newsletters sent with this type, such as subject and sender address.

Work area: CMS Backend

Start: Newsletter type is to be added, edited or deleted

End: Newsletter types are up to date

Frequency: 1-2 times a year

Sub-tasks and variations	Solution example	Code
1. List newsletter types		
2. Create newsletter type		
3. Edit newsletter type		
4. Delete newsletter type		

C.9 Manage newsletter subscriptions

Subscribers to a newsletter can come from many sources. Many can subscribe directly from the website, but at trade fairs, through customer studies or similar, a list of new recipients can be gathered, which has to be manually imported.

Work area: CMS backend

Start: One or more email addresses are to be added to a newsletter type

End: Email addresses added to the newsletter type

Frequency:Periodically

Users: Editors, administrators

Sub-tasks and variations	Solution example	Code
1. Subscribe recipient to newsletter type		
2. Delete recipient from a certain newsletter		
3. Change email address of recipient		

C.10 Manage newsletter subscriptions from front end

Subscribing and unsubscribing to a newsletter should be easy to do for a website visitor. However, the subscription and unsubscription processes should be handled with care and security so the customer can be safe in the knowledge that he is only sending out newsletters to people, who have opted for it. Sending newsletters to people, who never signed up or who have followed the unsubscription process is classified as "spam" and it can be a costly affair, if the organisation is fined for it.

C.10 Manage newsletter subscriptions from frontend

Work area: Frontend

Start: User wants to subscribe or unsubscribe to a newsletter

End: User subscribed/unsubscribed

Frequency: User: Once a year, System: Dailyb

Users: Visitors on the website

Sub-tasks and variations	Solution example	Code
1. Subscribe to a newsletter		
1p. Problem: Visitor might subscribe another person's email address or be unsure whether the subscription is confirmed	Send subscription confirmation email to user	
2. Confirm subscription	Subscription email contains link that the user must follow to confirm the subscription	
3. Delete subscription to a certain newsletter		
3p. Problem: Visitor can unsubscribe another person's email address or be unsure whether the unsubscription is confirmed	Send unsubscription confirmation email to user	
4. Confirm unsubscription	Unsubscription email contains link that the user must follow to confirm the unsubscription	
5. List newsletter types with indication of whether the user is subscribed or not		

C.11 Send newsletter

Be sure to confirm before sending a newsletter. Newsletters are sometimes sent out to a huge amount of subscribers, and there are often advertisers who pay to have their products in the newsletters. If the newsletter is not entirely ready, it gives the receivers an unfavourable impression of the company and/or the advertisers.

C.12 Manage news

The ability to display news is desirable by the majority of organisational websites. It can be news about the company, the market, products available, new employees and much more. To really benefit from the versatility of "news", articles should assigned to categories so it is possible to pull a "latest news"-component with only press releases or only new employees, relating the content to the context in which it is used.

Most CMS suppliers will be able to fulfil this task with a standard component.

C.13 Browse news

Listing too many news articles on a "Latest news" page will only annoy the visitors, but manually transferring articles to an archive is equally impractical to maintain. The CMS should in this standard component be able to only list a limited number of news in the various categories and have a common interface with numerous search and filter options in a news archive.

C. 11 Send newsletter

Work area: CMS backend

Start: Usually sent periodically, for instance twice a week

End: Newsletter sent or parked for later resumption

Frequency: Once a day/week/month

Users: Editor, author, Administrator

Sub-tasks and variations	Solution example	Code
1. Create newsletter		
1a. Variant: Edit newsletter		
2. Insert text		
3. Insert media files in newsletter		
4. Undo changes		
5. Send newsletter to recipients		
5p Newsletter can be sent out before it is ready		
6. Newsletter should be approved by somebody else in the organisation before is can be sent		
7. Send test, which sends the newsletter to one email address only		
8. Park newsletter for later resumption		
9. Find parked newsletter for resumption		

C.12 Manage news

Work area: Backend

Start: A press release, new employee or similar news is received by the editor

End: News story published

Frequency: System: Daily, User: Weekly

Users: Editors

Content	Solution example	Code
1. Create news story		
2. Edit news story		
3. Delete news story		
4. Insert media file		
5. Undo changes	Undo in the editor or roll back to earlier changes	
6. Save and publish news story		

C.13 Browse news

Work area: Frontend

Frequency: System: Daily, User: Weekly

Users: Frontend visitors

Content	Solution example	Code
1. View news overview	Page with latest x news listed	
2. Read news story		
3. View news archive		
3p. Problem: News accumulate to the thousands, too many to display on one page		

C.14 & C.15 Manage course enrolment and enrol to course

These two tasks are example tasks, connected to the data example in chapter D.

C.14 Manage course evaluation form

This task is part of a custom integration with an existing course registration application. See Chapter D for descriptions of data.

Start: Questions need to be added or removed from a course's evaluation form

End: Changes saved

Frequency: once a week

Users: Editors

Sub-tasks and variations	***Solution example***	***Code***
1. Create new question		
1a. Variant: Locate and edit existing question		
2. Set question type and answer options		
3. Save question		
4. Add question to a specific course		
5. Remove question from a specific course		
6. Undo changes	*Undo in the editor or roll back to earlier version*	
7. Publish evaluation form		
8. Export replies to Microsoft Excel format		

C.15 Reply to course evaluation form

This task is part of a custom integration with an existing course registration application. See Chapter D for descriptions of data.

Work area: Frontend

Start: Course participant wants to evaluate a course

End: Evaluation finished

Frequency: System: 20-30 times a week, User: 1-2 times per year

Users: Website visitors

Sub-tasks and variations	***Solution example***	***Code***
1. View a list of courses		
2. View course evaluation form		
3. Reply to evaluation form		
3p. Problem: Visitor evaluates a course he did not participate in	*Look up participant with username and password in course enrolment system*	
3q. Problem: Visitor evaluates a course multiple times	*Register that the user has evaluated the course.*	
3r. Problem: Users with database access can see, who has given which reply	*System does not allow anyone to connect a specific evaluation to a specific person*	

C.16 Search content

If visitors cannot find the desired information, they go elsewhere. Having an easy-to-use search functionality can be vital to a website.

Most CMS's have basic search components, but real search engines are usually not a kernel functionality in CMS's. Search engines are a rather large area with specialised search engine suppliers, and performance and features vary from simple database searches to search index engines with phonetic and synonym recognition that record searches without results. The latter type can come with a high price tag, but is very useful especially on legal and public authority websites where the language used in the content can differ from the language used by the visitors. However, a cheap solution will often suffice to begin with, while the customer discovers the specific user requirements to a search engine. The search engine is usually easy to replace, so the choice of search engine is not tied to the choice of CMS. Nevertheless, a good search engine is often a prerequisite for the website to become a success – especially if the CMS is used for an intranet.

C.16 Search content

Work area: Frontend

Start: User is looking for content

End: Content found

Frequency: System: Very frequently, User: Weekly

Users: Frontend visitors

Content	Solution example	Code
1. Search for content with a search term		
1p. Problem: Search does not find terms inside documents	Search index reads PDF, Microsoft Word and ODF documents	
1q. Problem: Search term is misspelled	Phonetic searches Search includes a list of common misspellings and their correct term	
1r. Problem: Search term has a synonym that is used in content	Search for synonyms to the search term	
2. View search results overview		

D Data to present

This chapter is about structured business data that should be presented on website pages. The data will be used by the CMS either by storing the data inside the CMS or by integrating with other systems that contain the data already. The tasks that are carried out in connection with the structured data – from the frontend as well as from the backend should be described as tasks in chapter C in the requirements specification.

To manage these data, the developers need to know the data structure and how it should be used and presented. This is done in the tables with descriptions of necessary data and information about where the data are currently stored. For these data, the customer is the one with the domain knowledge, as opposed to other parts of the CMS, where the suppliers are the experts. It is usually a good idea to consult the users, who are going to use the pages, about the procedures. Be aware of the fact that it is very likely that users have exceptions from official procedures, and this is impossible to handle by a system, unless the exceptions are analysed thoroughly and accounted for in the system. To avoid this kind of problems, be careful that the system doesn't restrict the users unnecessarily, and be sure to ask the users what they need for the system to become a success.

The data should be presented on specially developed pages. In order to ensure that the supplier can deliver an appropriate solution, the customer should give as much information as possible in data models. Alternatively, the customer can provide mock-ups of pages containing the data and let the implementation details be up to the supplier.

If the same kind of data is used by other systems as well, it might be a better solution to integrate with relevant databases or systems outside the CMS, since redundant data are hard to keep consistent and lacks scalability. Integration with external systems is covered in chapter F.

Data examples

Organisations often have data collected without really thinking of it as data. The example in the requirements template is an existing course enrolment application that should be extended with a course evaluation functionality, accessible on the website.

Getting this information into structured form inside the CMS can make updating and looking up information easier for everyone in the organisation as well as be of use to website visitors.

It is important to describe the existing data in as much detail as possible since the supplier will either build a new extension from scratch or try to apply the data to an existing extension with or without modifications. If the data are not thoroughly described, the supplier may think that an existing extension can be applied and only late in the process discover that a new extension or large modifications are needed that he has not included in the estimates.

The data model in shape of E/R diagrams presents:

1. Relevant parts of the existing data, that are used in connection with an existing course enrolment system
2. A course evaluation system that should be developed and presented on the site. This diagram is an example of how the business data could be structured.

If the data volume is very large, the supplier would have had to take it into consideration when making time and price estimates, so the customer should give an estimate on current and prospective volumes.

D Data to present

The customer needs the system to present data described in this chapter. The table below shows the data, that should be presented on different pages on the website. Besides presenting the data, the CMS should provide functionality to search, create, delete and edit the data as described in chapter C, tasks C.14 Manage course evaluation form, and C.15 Reply to course evaluation form.

The CMS shall extend an existing course enrolment application with course evaluations performed through the website. The course enrolment application is accessible through web services supplied by the customer.

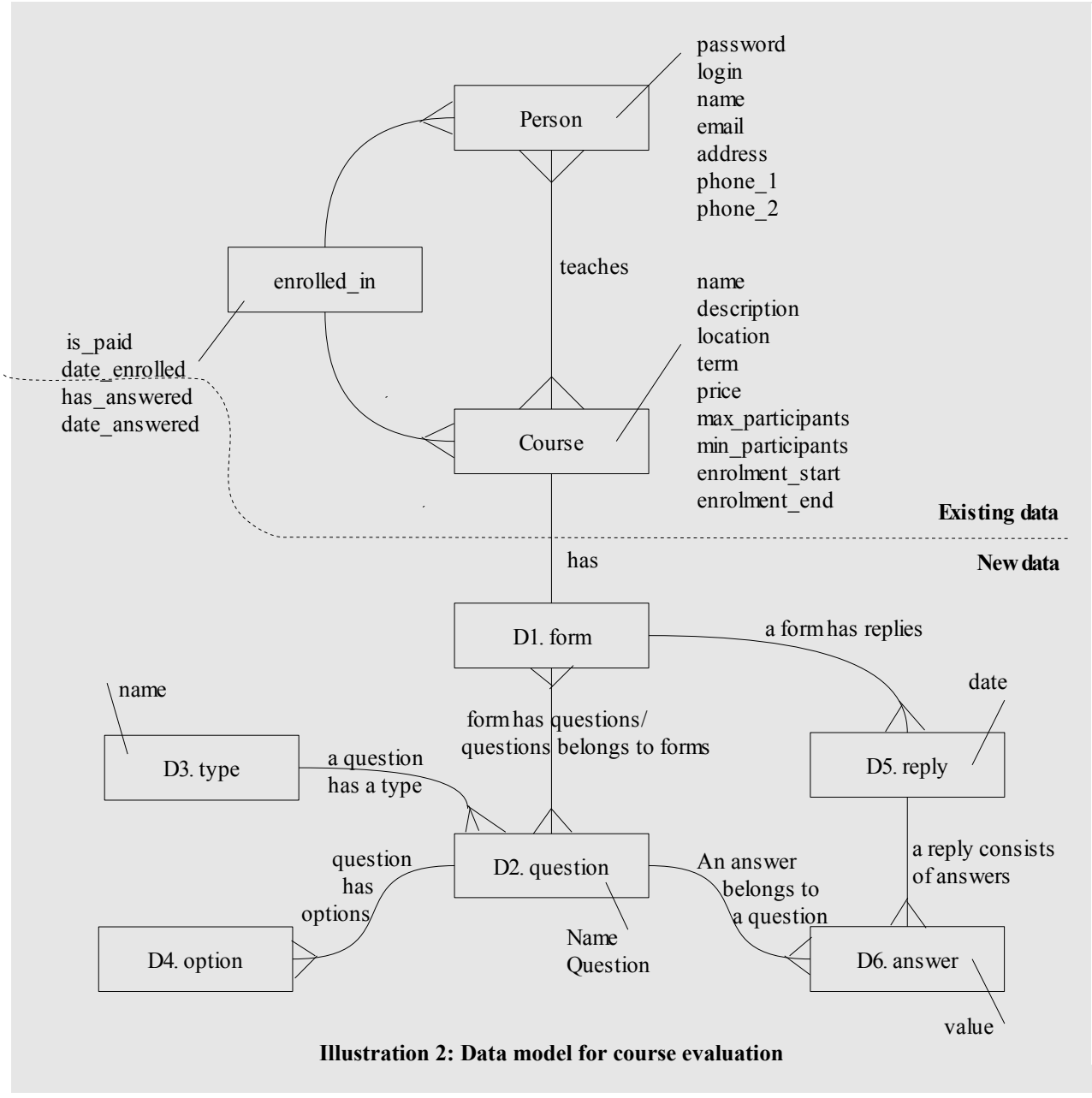

Illustration 2: Data model for course evaluation

D.1 Form

Connects questions to courses, specifying which questions should be asked in the evaluation of which courses. Note that a question can be re-used for several courses.

Data volume: *600, increasing by 200 per year*

No fields, only foreign keys. Table is here to create one single point of integration with existing data.

D.2 Question

Data volume: *400, increasing by 100 per year*

Fields and relations	Solution example	Code
1. QuestionID		
2. Name, used for short identification e.g. In look-up lists when adding questions to questionnaires		
3. Question in full length		
4. Question type, reference to types		

D.3 Type

Data volume*: 10-20*

Fields and relations	Solution example	Code
1. TypeID		
2. Name, used for short identification e.g. In look-up lists		
3. Question type, whether it is a group of checkboxes or radio buttons, text fields or text areas		

D.4 Option

Some question types have fixed options, e.g. Checkboxes, radio buttons and drop-down selection lists

Data volume*: 1000, increasing by 200 per year*

Fields and relations	Solution example	Code
1. QuestionID, reference to questions		
2. Value, i.e. the option text		

D.5 Reply

An answer is the reply to one course evaluation

***Data volume**: 50 thousand, increasing by 4 thousand per year*

Fields and relations	Solution example	Code
1. FormID, reference to forms		
2. Reply date		

D.6 Answer

An answer of one question to one reply

***Data volume**: 200 thousand, increasing by 60 thousand per year*

Fields and relations	Solution example	Code
1. QuestionID, reference to questions		
2. ReplyID, reference to reply		
3. Value, answer to the question		

E Other functional requirements

E.1 Complex calculations and rules

Processing data in chapter D will often be straight forward relationships between data, but there may also be calculations and rules that the system must obey, which only the customer can supply. These calculations and rules could be the logic behind determining who are eligible for discounts, which only the customer can decide.

E.2 Print-outs, statistics and reports

Do not underestimate the use of prints. Even though we are in the age of digitalisation, prints are often used at meetings, as reminders or to show friends and colleagues. Making sure that the product descriptions, service offerings or similar can be printed on standard paper directly from the browser will make the pages readable on printouts.

When it comes to visitor statistics, the customer will have to define what kind of statistics reports are needed. This is especially important if the organisation's business goals is to increase traffic; if the customer cannot get a report of weekly and monthly visits, he cannot determine if the goal is reached. Analyse the needs and make report mock-ups – this will be needed by the supplier as well as it gives the people, who are going to use the reports, an idea of what to expect.

E.3 Dynamic content components

In this table, you should input the components, you want to use on a page. The content will be managed centrally and "pulled" into the page based on the configuration on the page.

For example, news articles can be managed in a news module that allows content providers to write articles in different categories. Then, they can add a component with a list of the 5 latest news articles – possibly confined to one category – and place it on any page in the system. When a new article is added, the component is automatically updated on all pages, where it is used, instead of content providers having to edit each page individually.

In a customer organisation interview, we learned of a CMS implementation that allows for the content provider to select a contact person for each page. This person's contact details is visible when viewing the page, letting visitors know how to contact the person responsible for the page content. When a person leaves the organisation or responsibilities shift, all pages connected to one person can be transferred to another contact person in one go, instead of on each individual page.

E Other functional requirements

E.1 Complex calculations and rules

Function	Solution example	Code
1. A course evaluation cannot be viewed or submitted before the end of the course term and not more than 3 months after the term ends		

E.2 Print-outs, statistics and reports

Requirements	Solution example	Code
1. All pages displayed at the system front-end must be printable on standard A4 paper		
2. The system shall register page views and visitor statistics in a way so that the customer can pull reports on: a) Monthly visitors b) Monthly page views c) Sales/sign-up transactions started d) Sales/sign-up transactions completed e) Visitor paths through the website		
3. All reports must be printable on standard A4 paper		
4. All course evaluation replies for a course shall be possible to export to Microsoft Excel format		
5. Statistics on user activity in regards to page updates of these types: *a) Page-centric; when were the pages updated and by whom?* *b) User-centric; Which pages have the users updated in a specified period?*		

E.3 Dynamic content components

The customer wants to be able to add the following content components on any page in the CMS, either as part of the page content or in predefined areas of the page.

Components	Solution example	Code
1. User input forms with configurable input fields and post-submit actions		
2. News article lists, listing titles of articles and linking to the full text.		
2a. Variant: List x latest news items		
2b. Variant: List x most read items		

E.4 Expansion of the system

CMS's are made to be expanded – or they should be. Websites on the Internet have changed enormously over just a single decade and website owners are expected to explore new possibilities and expand the use of their websites continuously. Therefore, being able to expand the CMS with new content types, modules and dynamic components is a must-have requirement for any CMS and a customer should be able to change certain things very quickly or hire a third party to do so.

The CMS itself can be closed source, but the source code of custom built extensions paid for by the customer, should be available to the customer. It should be possible to have other developers extend the system, and therefore the customer needs access to a well-documented API for the CMS. Otherwise the customer will in practice be dependent on the initial developer, and that is usually not a good way of securing quality and reasonable prices.

E.5 Search engine optimisation

The optimisation of a website to be found high in the lists of popular search engines is a science of its own. The CMS can be an obstruction to this process or it can be a facilitator, but it will not be the sole answer to search engine optimisation. Many factors come into play, but the most important ones relate to the actual content written by the content providers – i.e. People outside the control of the CMS supplier. The design can be structured properly, the CMS can allow e.g. Titles on hyperlinks, set meta tags on the page and provide a page title, but if the users do not set a proper hyperlink title, relevant meta tags and good page titles, all the CMS' offerings will bring few results. However, the CMS should not *prevent* users from setting these and therefore, the requirements in this chapter are likely used to disqualify the poor CMS's rather than to select the right one.

E.4 Expansion of the system

The customer or a third party hired by the customer shall be able to expand the system without involving the supplier. In this section, "customer" means his own IT staff or a third party authorized by him.

Requirements to the extension possible	Solution example	Code
1. The customer can add new content types with corresponding management and output	Add components, e.g. generation or consumption of RSS feeds	
2. The customer can change behaviour of components and modules, *e.g. Add functionality to send an email to a teacher, when a course evaluation is submitted*	Edit/replace components, e.g. Replace the newsletter module with another	
3. The customer can change layout of existing content types	Content types use templates that can be edited through the template management, similar to page templates	
4. The customer can use a well-defined API to communicate with the core system. Preferably, the CMS source code is available. Well-defined means that everything that can be done by an administrator can be replicated using the API.		
5. The source code of all components constructed specifically for this implementation shall be available to the customer along with developer documentation. The customer shall have the rights to use and modify both source code and documentation.		

E.5 Search engine optimisation

Requirements	Example solutions:	Code
1. The system should generate human readable URL's to website pages and internal links in content		
2. Search engines must be able to browse the site	Site navigation is functional without JavaScript, Flash or other non-HTML technologies	
3. Page code structure is targeting search engines	HTML code is structured with navigation links first, then the page's unique content, then content common for several or all pages	

E.6 Other functional requirements

Functional requirements that are not directly involved with use tasks can be specified here.

E.6-1 Visitor accounts

User management through a directory service is an excellent tool to keep the IT department's workload at a minimum, but when the site also allows visitors to register accounts, certain considerations must be taken. Do we want just anybody into our directory service or should internal users be managed separately from external users? How will the system distinguish between them? Should internal users create external user accounts to participate in discussions and use the internal account purely for other content management?

There are many questions, so the requirements should specify in as much detail as possible, what would be acceptable and what not.

E.6-2 User action logging

Logging the actions of users can be helpful to determine who has done what and when. It can be used for keeping an eye on users – are they updating the pages, they are responsible for? - and it can be useful when investigating a breakdown or just determining, who has added a specific piece of content.

However, just as with statistics gathering, the level of user action logging should be specified as well as the filtering of data. Knowing exactly which part of the CMS backend is accessed by which user can be useful at times, but usually only one user's actions or one type of actions is necessary and filtering for this will reduce the noise of, in this case, unnecessary information.

E.6 Other functional requirements

Requirements	Example solutions:	Code
1. Visitors can create accounts and participate in discussions, manage newsletter subscriptions etc.		
1p. Problem: Backend users authenticate through LDAP, visitors should not – but they should be able to participate in the same discussions		
2. Log user actions with timestamp, user name and a description of the action when - a user is created, updated or deleted - a page is created, edited, deleted, viewed or moved - user subscribes/unsubscribes from newsletter - newsletter sent		
3. Log server errors, eg., - 404 Page cannot be displayed - 500 Server error		
4. The system shall support content in multiple languages, allowing easy translation of page content into different languages and linking to "this page in [other language]" when displaying the page.		
5. Backend users can choose backend interface language		
6. Notify person set as responsible for the page, if a page has not been updated within *14 days*		

F System integration with external systems

F.1 Directory service

If there are many users of the CMS, it might be a good idea to consider integration with a directory service.

A directory service is basically a network based application, that contains information about users and resources on a network. Examples of directory services are Active Directory, Novell and eDirectory, and the information in these directory services is accessed with a protocol called LDAP[2], and therefore integration with a directory service is also referred to as 'LDAP integration'.

The directory service is used to centrally manage users, resources, services and users' access to resources and services such as the CMS. This means, that the user can use the same user name and password to the CMS as to other systems that also use the directory service.

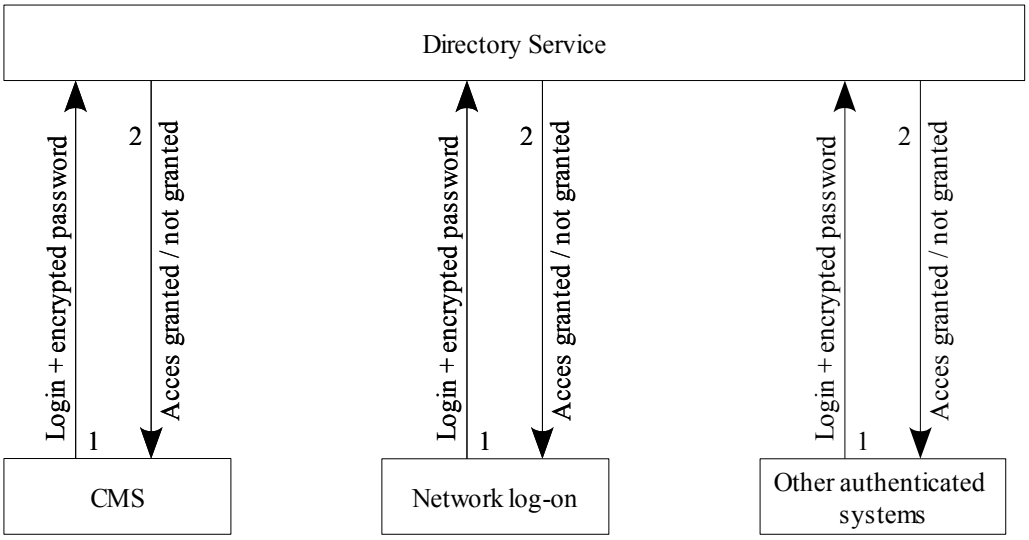

Illustration 3: LDAP integration - same authentication to different systems

The main benefit with LDAP integration is, that users have only one user name and password combination to remember and can change it once for all the systems authenticating with the directory service. It is often used in connection with single-sign-on where a user is granted access to several systems as soon as he has locked on to the domain administered by the directory service.

A directory service can have several domains, that reflect the geographical or organisational structure of the organization. An example could be a multinational cooperation, that has affiliates around the world on different domains. In this case each affiliate might have a sub-site with locally relevant information communicated in different languages. It might have its advantages to let the website reflect the structure of the organization, and grant some users from each domain access to edit their sub-site.

2 Lightweight Directory Access Protocol

F System integration with external systems

The CMS shall integrate to the following external systems

1. LDAP Directory service
2. *Google Analytics, website statistics gathering solution*
3. *Course enrolment system*

F.1 Directory service

The directory service is used to authenticate internal users with the same username and password as in other systems

Integration requirements	Solution example	Code
1. The CMS shall authenticate users through LDAP in real-time		
1p. If a user is deleted in the directory, the CMS can suddenly have pages with a page responsible that no longer exists	*Invoke code in the CMS when a user is deleted in the directory, notifying relevant people about the deletion.*	

F.2 Analytics software - Google Analytics

Google Analytics gather statistics on page views and visitors

Integration requirements	Solution example	Code
1. Register page statistics in Google Analytics	*All pages must include the Google Analytics JavaScript code*	

F.3 Course enrolment system

The course enrolment system can communicate through web services or, if the CMS is hosted at the customer, direct database access to certain views in the Oracle database running the enrolment system. If the CMS is externally hosted, only web services are available. Web services or database views will be created by the customer to fulfil the supplier's requests.

If the supplier provides a solution with a higher degree of integration (1 is lower than 2 is lower than 3), he does not need to provide a solution to a lower degree.

Integration requirements	Solution example	Code
1. The CMS periodically replicates course enrolment data		
2. Users can initiate manual replication, when needed		
3. The CMS pulls data about courses, its teachers and students from the course enrolment system in real-time.		

F.2 Analytics software

Statistics gathering is important, as can be read from the business goals in chapter B. Additionally, website statistics can give valuable information on which content is read by visitors, which pages are linked to from other websites and on which pages the visitors leave the website. This knowledge can be used to improve the website structure and content.

F.3 Database systems

Integration with systems such as Navision, Siebel or SAP, can be of great value to the automation of a website. It is always beneficial to consider direct integration with other systems rather than inputting data into several systems. If the website is asynchronous with the other systems, data will have to be updated in both systems, thus doubling the workload and making data less reliable.

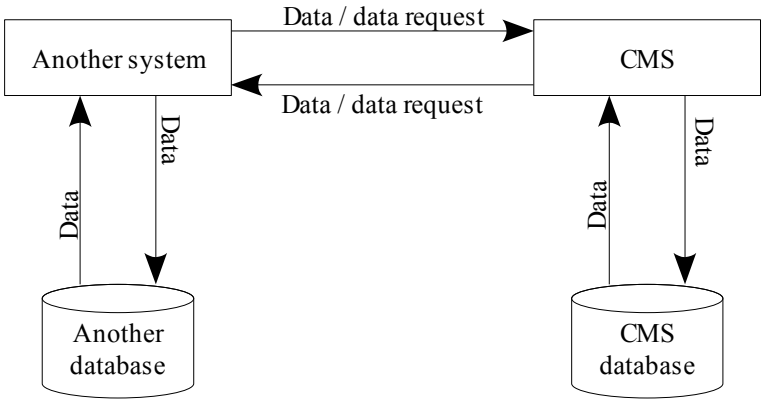

Illustration 4:Database system integration

F.4 Integration with new external systems

When integrating with other systems, the CMS can have the role as either the server where other systems initiate functionality or the CMS exports data, or as the client where the CMS imports data from other systems or initiates functionality in other systems.

To facilitate the integration, it is also important to evaluate the developer documentation and rights to data, ensuring that you – or someone you hire – can integrate other external systems with the CMS and the data within the CMS without involving the supplier

F.4-3 Documentation and rights

Being "locked" to only one vendor after the CMS acquisition is a risk that should, and usually can, be avoided. Complete portability between different CMS systems – where content from one can be automatically transferred loss-less to another – is unrealistic without customised migration scripts or programs, but data should be stored in a standard database and the database schema available, so data are always reachable with other tools than the CMS and a skilled developer will be able to ascertain which information is stored in which tables.

We recommend that you ensure that there are multiple consultants available for the CMS so you have alternatives, should the first consultant suddenly increase prices, drop in quality, become hard to reach or go out of business. Healthy competition in the area of consultancy services will only be of benefit to you as a customer.

F.4 Integration with new external systems

The customer expects to integrate the CMS with other systems, possibly using a third party consultant. The CMS shall be prepared for this integration through the following requirements.

F.4.1 Server role

System interface in server role	Solution example	Code
1. The system shall be able to export data in a format, readable to other systems	*Standardised export format* *Plug-in architecture for sending data*	
2. Functionality in the CMS can be triggered by other systems. It shall be possible to use an API accessible from other systems to - Create a new or modify an existing user - Create a new page with content - Perform a search in the CMS' search engine	*Webservice API, XML-RPC etc.*	
3. The database is available for use by other systems		

F.4.2 Client role

System interface in client role	Solution example	Code
1. The system shall be able to import data from other systems and automatically create new pages in the page hierarchy.	*Standardised import format*	
2. Functionality in other systems can be triggered by the CMS, e.g. Calling a webservice and processing a reply	*Webservices, XML-RPC etc.*	

F.4.3 Documentation and rights

Documentation and rights	Solution example	Code
1. API documentation should document all areas of the API available to extensions and components in a way so it is understandable to a third party		
2. The customer or a third party appointed by him has all rights to use the documentation and API		
3. The customer holds all rights to all data stored in the CMS		
4. Content data are stored in a standard database, with connectivity through standard database tools, making access to data available without using the CMS		

G Technical IT architecture

G.1 Hardware requirements

G.1-1 Existing hardware

If the customer has hardware that he expects will be sufficient for running the CMS, he should list the specifications and the supplier can reply with expected capacity of such a system.

G.1.-2 Hosting

The customer hosting the CMS will ensure physical access to the servers and complete control of what is running on which servers. However, it also requires IT personnel with web server management competencies to keep the servers secure and running.

Hiring a data centre to host the CMS can outsource the need for web server managers in exchange for less control with the physical servers – and perhaps a higher price.

Often, CMS suppliers offer to host the CMS for customers. The advantage of this solution is that it will be hosted with someone, who knows the CMS very well and who can include CMS security updates in the normal server maintenance procedures.

G.1-4 Test and development environment

Adding new features, updating the CMS or changing the set up can be an intimidating task on a production website with many risks ranging from misplaced images to server errors. Therefore it can be useful or even necessary to have a range of servers for developing, testing and staging these changes

Development server
The server used by developers (in-house or external) to develop new features, modules, templates etc. Content is test content to test whether a feature works or a template looks correct. Not everything is finished and ready for deployment to the test server.

Test server
Server used by developers for testing whether developments work together with other developments and to show changes to interested parties before deployment on the staging and production servers.

Staging server
This server is used by content providers to make changes to the content or site structure without affecting the website visitors, until the changes are published to the production server. Content should always be synchronised with the production server, barring the unpublished changes.

Production server
Server (or several servers in a load balancing solution) accessed by the website visitors.

For organisations with large websites containing critical data or functionality vital to the company, all of the servers mentioned in the requirements template are likely to be necessary. For others, only the production server. For most, a development and test server can be combined into one and a good versioning and preview function can combine staging and production servers.

G Technical IT architecture

G.1 Hardware and software requirements

The customer currently has the following equipment, to be used in the CMS operations:

1. *2 servers with these specifications ....*
2. *100 Mbps connection to the Internet*
3. *4 Oracle database licenses*

Requirements to hardware and software of the customer	**Offered solution**	**Code**
1. The system is expected to run on existing hardware, complying with the requirements in L.1 and L.2	Under these circumstances, the system can handle ___ hourly page loads. (The customer expects ___ page loads)	
2. With increasing traffic and use, the hardware will need to increase, too, to comply with requirements in L.1 and L.2	For each __ hourly page loads, the hardware requirements increase __ to comply with L.1 Response times	
3. The solution shall be hosted by the *supplier*		
4. A testing and development environment where new and edited components can be tested as well as new designs and other templates can be showcased to interested parties.	*A test server is required with these specifications:* .... *A development server is required with these specifications:* ...	
5. Operating system requirements		
6. Other software requirements		

H Security

Something that is taken for granted by many CMS customers is the aspect of security. If considered at all, emphasis is usually on accidental adding/editing/deleting of content by users or unauthorised access to the server itself. However, other areas should be formalised and considered properly.

H.1 Access rights for users

A basic feature of any CMS is the ability to block off areas of the website from the eye of the general public. Most notably, the administration backend should never be accessible without proper user authentication.

H.2 Security management

Managing users and their access rights should be targeting the IT department and accommodate their wishes of easy user administration – possibly centralised through a directory service as covered earlier.

Managing the permissions in the directory service is rarely available, but some CMS's have synchronisation of user groups and user group memberships between the directory service and the CMS, so defining the permissions on a user group basis allows for moving users in and out of groups directly in the directory service management application, saving the IT department from keeping user group memberships synchronised manually.

Should the requirements not include authentication through a directory service, it is important to ensure that the user and permissions management is suited towards the number of users. With many user accounts, it becomes a very large task to assign permissions to each and every one of them or assign users to user groups individually.

In chapter E.5, we suggested that the system should log user actions. However, for the logging to hold real value for the users, there should be a way of reviewing and filter this log so only data relevant to the reviewer is presented.

H Security

H.1 Access rights for users

Manage users, rights, roles, and responsibilities

Requirements to security	Offered solution	Code
1. Access security, ensuring that only authorised users access regulated areas		
1.p Problem: Users use several systems and forget user names and passwords if there are too many.	*Single-sign-on through directory service*	

Access rights:

1. Right to view pages
2. Right to add/edit pages from a certain level and down in the document tree
3. Right to delete pages below a certain level in the document tree
4. Right to upload files to the media library
5. Right to edit or delete files in the media library
6. Right to use HTML and JavaScript in page content editing

An editor might have rights 2,3,4, while a content provider may have only 2,3 and an administrator has all rights from 2 through 6.

H.2 Security management

The work in security management includes the following subtasks.
Difficult: *When 120 new people have been employed and need access rights from day one.*

Subtasks for security management:	Example solutions:	Code
1. Assign or remove access rights for a user.		
1a. Variant: The user must be created.		
1p. Problem: A lot of users have to be created, e.g. when they start the first day in the month.	*Automatic transfer of data from the personnel system.*	
2. Create new user roles		
3. Assign roles to a user		
3.a Variant: Assign users to a role		
4. Assign permissions to role		
5. Get an overview of who has which rights and whether some rights have not been assigned to anyone.		
6. Review user action log (see E.5)		
6p. Problem: User action log is filled with data about normal or unimportant actions	Ability to filter log on user, action, time period or similar parameters	
7. Delete user		
7p. Problem: User is responsible for one or more pages	Ask for replacement user to take over responsibilities	

H.3 Protection against data loss

Measures protecting against loss of data should cover three situations:

1. Transfer of data between servers or systems – ensuring that data are received correctly before being purged from the sending system.
2. Concurrent use of the system by multiple users – protecting against corrupted data as a result of several concurrent writes of the same item.
3. Physical damage to or theft of servers – protecting the servers from physical access by unauthorised personnel, proper fire extinguishing measures and backup power supplies.

Whatever measures are taken, always be sure to test the backup and restore procedures regularly – at least once a year – both to be certain that the backup and restore procedures are working *and* to ensure that the responsible persons are familiar with them, so they know what to do and how to do it properly.

H.3-1 Transfer of data

Transferring data from one system to another is always a situation to be cautious. Many factors influence a proper data transfer, ranging from the physical condition of wires to the other processes performed by the servers running the systems. Therefore, it can be resources well spent, ensuring that data are transferred correctly, thus not ending in a situation where data have been deleted in one system without being properly received at the other end.

H.3-2 Concurrency issues

When two users open the same page for editing at the same time and then save it, the CMS can have difficulties determining which edition is the right one to use. Should it overwrite the first with the second? Disallow saving the second since the original has been changed? Disallow opening the second for editing while it is open for editing by someone else? Save the second as a new version so the first is still available? Should some users have privileges that overrule the restrictions?

There are advantages and drawbacks with all the mentioned solutions – and for most organisations it will be a non-existing problem. Usually, site content responsibilities are divided among users, who each have their area. If the responsibility for a page's content is shared among several users, they will usually work closely together and communicate about changes to pages without the need for CMS enforcement of edit rules. However, an indication or warning that a page is already open for editing, would facilitate the communication.

H.3-3 Physical damage or theft

Web servers should be kept protected from physical damage or theft just like any other asset, vital to an organisation. This could be an argument in favour of using a hosting company to host the physical servers, if adequate protection is not available at the customer's location.

When the customer is not hosting the CMS it is particularly relevant to ensure that the data on the servers belong to the customer and not the hosting company and that the customer frequently receives back-ups of all data required to reproduce the website(s) on another platform.

H.4 Protection against unintended user actions

No matter if a user is a long-time IT professional or has only just found the power button of his workstation, CMS users can perform actions that are not intended to be performed. Ranging from answering "yes" instead of "no" in a confirmation dialogue to editing the wrong page, users can do the wrong thing and will need to undo it. How much can be undone and how it can be undone is very different from CMS to CMS, but at no point should the system shut down or otherwise malfunction from a regular user's wrong click on a button. If the users can

also be certain that they cannot make functional mistakes like writing a date in a text field, a number in a date field and text in a number field or that they can revert to a previous version of a page, they will feel more secure with the system and less nervous of making mistakes.

H.3 Protection against data loss

Data may unintentionally be lost or misinterpreted.

The system must protect against:	Example solutions:	Code
1. Loss or replication of data transferred between two systems, e.g. because one or both systems close down.		
2. Concurrency issues, ex. two users editing the same page at the same time should not result in data loss.		
3. Loss of data, because of fire, theft or hardware failure	Periodical backups	
3p. Problem: The backups can be lost if it is kept in the same location as production data, should the location suffer theft, fire or similar physical dangers	Store backups at different geographic locations	

H.4 Protection against unintended user actions

Unintended user actions mean that the user happens to do something he didn't intend to do, e.g. hit the wrong key or use 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.		
2. All data entered must be checked for format, consistency and validity. In case of doubt, the user must be warned and asked what to do.		
3. Versioning should be available, so users can roll back to a previous version		

H.5 Protection against threats

Not many consider – or have the knowledge – to ask questions to suppliers about the security level of the CMS when it comes to SQL or code injections, allowing malicious users to make changes to the database or run unauthorised code on the system. Therefore, it can be beneficial to consult with a security expert to evaluate the security of prospective systems. However, some aspects can be covered beforehand, by asking the supplier to specify certain security measures.

H.5-3 SQL Injections

Pages in a CMS are for the most part dynamically generated by looking at the user's input (accessed URL, parameters in the request) and looking up the corresponding page's content in the database. If the input from URL and parameters is not handled correctly before being used in a database query, malicious visitors may be able to include SQL syntax that will be executed. This could lead to discovery of passwords (another good reason to use a directory service for authentication), modification of content or change of user permissions, giving the visitor administrator rights in the CMS.

A good CMS will ensure that all user parameters are handled as not-to-be-trusted and filtered before being used in database queries.

H.5-4 Cross Site Scripting (XSS)

Even if the CMS filters the user parameters to avoid SQL injections, the user submitted input can still be of danger to the website and its visitors. Cross Site Scripting (abbreviated XSS to avoid confusion with CSS – Cascading Style Sheets) involves adding JavaScript code to the user input, which will be executed by the visitor's browser, when he visits the page with the code. XSS attacks can result in cookies being stolen or redirecting the visitor to a malicious site with even more malicious code to attack the visitor's computer.

This is mainly an issue for websites where visitors can contribute content, e.g. write comments in a discussion forum.

To avoid XSS problems, the CMS should escape the input so that it is not executed when displayed on a web page. Additional measures could be logging XSS attempts and notifying server administrators of the possible attack.

H.5-5 Cross Site Request Forging (CSRF)

The HTTP protocol used by web browsers to communicate with web servers is a state-less protocol, meaning that each request sent by the browser will be handled by the server as if there have been no previous requests. Therefore each request must contain all the information needed by the web server to serve the correct page. This also means that anybody can build an HTTP request and send to a web server, which will handle the request based on the information.

CSRF attackers exploit this, by forging a request to look like it is coming from an authenticated user, who is allowed by the server to perform actions, such as modifying content or granting CMS access rights.

The usual way to determine whether a request is genuine and part of a session for a logged-in user, is to set a cookie on the user's computer that will be sent along with the request. Provided that the malicious person has not stolen cookies through XSS or physical access to the user's computer, it is very difficult to know the cookie content to send with the request. However, if the malicious person can somehow get the user to *send the request himself*, the cookie will be included and the attack may succeed.

H.5 Protection against threats

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

The system must protect against:	Example solutions:	Code
1. Unauthorised persons obtaining administrator rights trough the Internet (hacking).	*Administration can only be accessed on the internal network OR* *Administration can only be accessed from a whitelist of IP addresses*	
2. Wire-tapping of passwords.	*Password encryption* *SSL encryption on login page*	
3. SQL Injection protection		
4. Cross-site-scripting protection		
5. Cross-site-request-forging protection		
6. Session hijacking	*Privileged users cannot change IP address within a session*	
7. Visitors subscribing or unsubscribing other people's email addresses to/from newsletters (see chapter C.10-1p and C.10-3p		

There are several methods for a malicious person to trick the user into sending the forged request:

1. If the web server performs the desired actions with GET parameters (parameters sent through the URL), the attacker can merely add a link in a comment, hoping that a CMS administrator will click on it or – if the comment system allows this – add an image with the attack link as its source. When a CMS administrator clicks on the link or views the page with the image, the URL will be invoked with a request from the administrator and the web server will perform the desired actions.
2. If the web server allows HTML in the discussion comments, a malicious person can build a web form, e.g. With an image as the "submit button" and writing something along the lines of "click on the image to see an enlarged version", hoping that a CMS administrator will do so, thus submitting the form.
3. If neither option is available, the malicious person can build the attack form on another website where an option is available – or his own, where he can add content without restrictions – and then trick the CMS administrator into visiting the website and clicking the link. This is of course a very unreliable method for the attacker, but it happens nevertheless.

Protection against CSRF attacks can consist of

- Never accepting database changes from GET requests – except logging and similar actions.
- Disallowing HTML for website visitors or have a small white-list of accepted HTML that the visitors can use, filtering or escaping all other HTML. Many websites allow no HTML but instead allows the use of so-called "bbcode" where pseudo-code is transformed into HTML within very restricted limits.
- Checking the HTTP_REFERER property of the request. This will contain the URL the user is coming from. If it does not come from the same web domain as the website, it can be discarded. However, many software firewalls block this information to protect users' privacy, since it allows websites to register, where the user is coming from.
- Ensuring that submitted forms include keys that a malicious user cannot know about and incorporate in his attack forms. This could be a randomly generated key saved in the user's session data (which is stored on the web server) that is only valid for a period of time and added to the form dynamically. When the form is submitted, the web server can check to see whether the key is there and whether it corresponds to a key in the user's session.

I Usability and design

Usability of the CMS is a decisive success criteria. The usability has often been ignored, because the focus has been on the usability of the CMS frontend, i.e. the website, or because the customer does not know, what to look for when the CMS is chosen.

When choosing a CMS it is decisive to test the usability of the backend before contract signing as described in chapter B, while the usability of the frontend design can be tested after contract signing. As the design of the CMS backend is a basic part of the CMS, it is usually not easily changed, and therefore the usability has to be evaluated before a CMS is chosen and a contract signed.

A way of measuring usability, is to let a user carry out relevant tasks (chapter C) and count the number of critical usability problems and demand, that there may only be a certain number of critical usability problems. The definition of a critical usability problem is a problem, where the user:

1. cannot carry out the task on his own
2. or believes that the task is carried out even though it is not
3. or expresses that this is difficult, really!
4. or the test sees, that the user does not use the system in an efficiently manner

Table 4: Critical usability problems

To perform a usability test, require that the suppliers show the backend from an existing CMS solution, and perform think-aloud-tests with future users to find potential problems. The think-aloud-test should contain as many tasks from the requirements specification chapter C as possible, as we have also described in chapter B about proof of concept.

I.1 Learning and efficiency in daily use

Usability is primarily about how convenient it is to use the system. The typical users of the backend are editors, administrators and authors. To frequent users of the CMS, efficiency is the most important usability factor, while ease-of-learning is the most important usability factor to users, who seldom use the CMS. However, both usability factors are important, because the bigger a site is, the more likely it is that the responsibility of different pages and sub-areas are taken care of by many not-so-frequent users in an organisation, while the few people overall responsible of the site use the CMS very frequently.

If the backend is not logical or intuitive, it will be a problem especially to infrequent users. The result of poor usability can be that organisations give up distributed update, using only one or two people to do all the editing. If the CMS is not efficient, it is very inconvenient to frequent users, who use the CMS as their primary work application. Either way, it will make the users less motivated to use the CMS and thus less motivated to update the content frequently and fulfil the B.1-3 and B.1-9 business goals. The ideal situation is that the CMS supports the organisation rather than the organisation having to fit to the CMS, so it is a good idea to spend time and resources to check the usability of different systems thoroughly.

Often it is much easier to find "ease-of-learning"-usability problems than it is to find "efficiency"-usability problems, because the "efficiency"-usability problems become visible only to users, who have to do the same inefficient task 50 times a day.

I.1 Learning and efficiency in daily use

It is important that the system obtains adequate usability. This is best done through early usability tests. After the early tests, customer and supplier jointly decide the detailed requirements to be verified at the time of system delivery. It may for instance be a detailed specification of the tasks to be tested, and the acceptance criteria to write in the middle column.

Usability test of proof of concept	Example solutions:	Code
1. The supplier must test the user interface for usability before signing the contract.	Usability testing (thinking-aloud testing) is carried out for existing parts of the system in a suitable setup.	
2. The most serious usability problems must be corrected until usability testing gives acceptable results. In addition the parties must agree on the detailed usability requirements.	For parts that don't exist yet, thinking-aloud testing is done with paper mockups. Three new users participate in each round of testing.	
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 without critical usability problems.	Within each work area, thinking-aloud testing is done with ___ ordinary users. A maximum of ___ critical usability problems may be observed.	
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. Super users must be able to learn the system quickly so they can train other users.	Training of a super user takes ___ days. (The customer expects _ days).	
6. A user who has used the system for a week, must be able to quickly create a new page and add content	A typical user is able to create a page with text and images in __ minutes.	
7. Usability of consultant-developed interaction, ie. user test of data view and functionality at the frontend of the CMS	Within all consultant developed interaction modules, thinking-aloud testing is done with ___ ordinary frontend users. A maximum of ___ critical usability problems may be observed on each module.	
8. Easy access to CMS-backend	Single sign-on – If the CMS is integrated with a directory service, there will be access to the CMS as soon as the user has logged on to the domain according to their permissions This can be done within ___days.	
9. Easy access to intranet	Single sign-on. “Remember me” functionality. If the CMS is integrated with a directory service, there should be access to the intranet as soon as the user has logged on to the domain. This can be done within ___days.	
10. Consistent design of functionality. For instance, a 'save' or an 'edit' button or menu should be alike, no matter if it is a template or a content item that is being saved or edited.		

For instance, we have interviewed a user who frequently publishes a lot of content items on a large website. Every time he presses 'publish', he has to wait about 10 seconds while the pages are being published, without being able to do anything in the meantime. Apart from that the system can either publish a single content item or publish all unpublished content-items. When the user publishes e.g. 44 out of 65 unpublished content items, he can only do it one by one and each time wait at least 10 seconds. This is inconvenient as well as time consuming. An additional problem is the preview functionality that does not look exactly like the published content item. Therefore the user actually has to publish every time he makes small changes, if he needs to see how it really looks. This is especially relevant on the frontpage of the website, where line breaks have to be controlled and text has to fit into different boxes on the page.

These are typical efficiency usability problems that are often not noticed in time and reduce efficiency and flow in daily work tremendously. Therefore it is recommended to pay special attention to the problems, that are connected to the different tasks in chapter C when carrying out usability tests as well as to specify usability tests with enough volume to test the efficiency.

I.2 Accessibility and Look-and-feel

The CMS backend is usually not optimised towards visually or otherwise impaired people, while the accessibility to the frontend to some extent can be supported or restrained by the choice of CMS. This is a large subject, with many different approaches to linking, titling, content organization and rapid navigation, as well as generation of multiple sites optimised towards the special browsers used by people with different kind of impairments. Therefore the accessibility requirements might not be satisfactory, and if accessibility is a critical issue, where it isn't enough to follow an accessibility standard such as WAI, it is necessary to consult an expert in this area.

Many CMS's attempt to exploit the general user's familiarity with applications such as Microsoft Word by creating page text editors with a similar toolbar layout as Microsoft Word. Familiarity with known programs, usually makes them easier to learn and remember, but as such a familiar look is not always followed by familiar functionality, and if that is the case, the user is better off without the familiarity.

I.2 Accessibility and Look-and-feel

Accessibility requirements	Solution example	Code
1. Web pages must be suited for screen readers, scaling to visually-impaired users, and utilising the full screen size on small as well as large screens.	The pages follow the HTML guidelines for Accessibility (WCAG10 from W3C).	
1a. Variant: Alternative text in connection with pictures, so that the screen reader has something meaningful to read	CMS supports or even forces the user to type in an alternative text to the picture	
1b. Variant: Check for complexity of link names for the sake of disabled who use screen readers	Give a warning if for instance a link is called 'here', like in 'Read more about SOA here' which makes it difficult for a visually impaired to jump from link to link, and get any meaning out of it. Instead it should say 'Read more about the subject in the article about SOA'	
1c. Variant: Easily adjust font size for better readability		
1d. Variant: Input fields should be connected to their titles out of consideration for screen readers	Title label for input fields	
2. User interface should be familiar to most users		
2p. Familiar look is not always followed by familiar functionality and the familiarity is confusing rather than an advantage.	Only use familiar icons when they perform comparable actions as in the program or operating system they are based on.	
3. It should be easy to navigate in the CMS backend	In a usability test, users shall be able to locate all functionality relevant to their tasks without any critical problems.	
4. The interface shall provide the user with enough information so that he can determine how a page will look when published	WYSIWYG (What You See Is What You Get) editor, where content is displayed with the formatting as exactly as it will look on the front end . Editor surrounded by content as it is on the front end There could be an easy and fast preview function, where the front end page is loaded with the edited content.	
5. Moving content in page hierarchy should be easy and intuitive	*Pages should be movable by dragging and dropping a page with all sub-pages or mark pages to move and selecting the destination*	

J Other requirements

J.1 Standards

Concerning requirements to follow standards, it is important to be specific and have some background knowledge. Some CMS's use accessibility standards compliance (WAI-compliant) as a sales parameter, even though many of the requirements in this standard are virtually impossible to enforce by the CMS. The CMS can enforce an alternative text to images and title properties on hyperlinks, but the WAI standard says that these texts and titles must be *meaningful*, which – so far – no computer can determine reliably. Therefore, the requirements specification should include specific requirements dealing with these standards rather than just state that they must be followed (See I.2 in the requirements specification template for examples of requirements related to the WAI accessibility standard)

J.2 Training

The easier the CMS is to use for a common user, the easier it will be to return to the CMS after a break or to include many users in different areas of the organisation, so pay attention to the Usability parameters (see chapter I Usability and design)

With increasing complexity of the CMS, the need for training the users and administrators also increases. The availability of courses from the CMS supplier or a third party can be an important factor in choosing the right CMS.

J.3 User Documentation

Documentation is equally important for the user training. Good documentation of an easy-to-use CMS may even eliminate the need for formal training, although it should not be expected. The documentation should be sufficient so that:

1. Superusers can train other users.
2. Superusers have a reference guide for all functionality available to them.
3. Any competent third party can read documentation and learn enough to extend the system.

Furthermore, documentation should be machine readable so it can be electronically adapted and redistributed. That way, the customer can make his own documentation that for instance not only explains how to perform actions, but also related company policies or common mistakes.

J.4 Data conversion and import

Often, the organisation will have a website already, and desire to quickly import the contents of this into the CMS. Requirements to import functionality should be listed here.

Importing data can be a large implementation task and therefore it should be very well established, how this import can occur. The customer should decide on using formats that explain the data thoroughly, using standardised formats wherever possible.

Remember – importing takes time. If 6000 pages are to be imported from an old website, these pages will need to be validated to ensure that they are correctly imported. If they also need some manual processing, 15 minutes per page amounts to 1500 hours that should be included in the manpower calculations.

J.5 Installation

This chapter establishes the formalities surrounding who installs which parts of the solution.

J Other requirements and deliverables

J.1 Other standards to be followed

Requirement	Solution example	Code
1. Page output shall comply with the standards in (X)HTML *version 1.0 strict*		
2. CSS files shall comply with the standards in W3C's CSS standards		

J.2 Training

Requirement	Solution example	Code
1. Training needed to become superuser of the system		
2. Training needed to become regular backend user of the system		
3. Training needed to use the system after a period of absence – training performed by a superuser		
4. Training needed to administrate the system		
5. Training needed to extend the system		

J.3 User Documentation

Requirement	Solution example	Code
1. Course material shall be available for superusers for training other users *at least one month before training begins*		
2. Documentation of all user functionality shall be available to superusers		
3. Custom developed code must be documented sufficiently for a third party to develop it further		
4. Documentation must be machine readable and the customer must have permission to use and modify it, enabling the customer to improve the documentation or modify it to reflect the customer's specific rules and processes.		

J.4 Data conversion and import

The supplier shall perform the import of data from the following systems:

Requirement	Solution example	Code
1. Import data from old course evaluation application. Data will be available in an XML format, described in Appendix...		

J.5 Installation

This chapter specifies, who installs which components – hardware as well as software – and who imports data from other sources.

Requirement	Solution example	Code
1. New hardware shall be installed at the hosting location	*Supplier installs new hardware*	
2. Operating system and software shall be installed	*Customer sets up server operating systems, web servers, FTP servers and database servers*	
3. CMS shall be installed and configured	*Supplier installs and configures the CMS to production setup*	
4. Data shall be imported as described in J.4	*Supplier imports data*	

K Customer deliverables

The implementation project is not a one-sided affair. In order for the suppliers to deliver a good solution, certain practicalities must be handled by the customer organisation.

1. Hardware – physically procuring the required hardware, allowing time for delivery delays, is the customer's responsibility, unless the desired solution involves hosting by the supplier or another third party.
2. Office space – even consultants have to sit somewhere, preferably in a close proximity of the customer organisation's project team members.
3. Production data samples – Demonstrations and dummy data can aid in development, but production data ensure greater resemblance to the end solution.
4. Test cases – Supplying test cases so the supplier can test the solution.
5. Test subjects for usability tests – Future users of the system must be allocated time to perform usability tests, even if it interferes with their daily tasks.
6. Project team members – To ensure good communications between customer and supplier, at least one project leader from the customer organisation should be available to the supplier. Depending on the project size, needs can range from a part-time contact person to several full-time positions, working with the supplier to evaluate development, answer questions about processes etc.
7. Superusers – Superusers must be trained in the CMS sufficiently to pass their knowledge on to other users.
8. Validation of converted data – The supplier may not be able to properly validate that converted data have been converted *correctly* and resources from the customer organisation should be allocated to ensure correct conversion.
9. Contributions to course material – Giving feedback on course material can help the supplier improve the quality, benefiting both the customer and other users of the CMS.

K Customer's deliverables

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

The customer delivers:	Example solutions:	Code
1. Hardware, software, and external systems that the CMS system requires (see the details in Chapter G). The equipment must be available when the installation test starts.		N/A
2. Office with three IT work places from one month before the planned installation test to one month after system delivery.		N/A
3. Samples of production data for testing purposes and the full data set for conversion.		N/A
4. Test cases for deployment testing.		N/A
5. Test subjects for usability tests.		N/A
6. A half-time project manager and a half-time secretary.		N/A
7. Super users/instructors who learn the system in order to train ordinary users.		N/A
8. Expertise for validation of converted data.		N/A
9. Contribution to the course material on future work processes (cf. J3-1).		

L Operations, support and maintenance

L.1 Response times

Defining fair requirements in the area of response times is a difficult task since many factors come into play with web servers. The number of visitors and subsequent strain on web server and database server as well as latency and physical distance between content providers and the web server can all have negative effects on the response times of the system. However, the CMS should provide measures to ensure that the most used tasks are performed as quickly as possible without too much going on in the background, affecting the response times.

A possible solution to lowering the response times is a caching solution. When the content on a web page is the same for everyone visiting at all times, pre-publishing, where a page is saved in a generated form and subsequently delivered directly and only updated when the page content is changed, should be possible. For dynamic content that pulls information from other systems, a time-based caching will most certainly help cope with peak situations with many concurrent visitors.

Most CMS's include some kind of internal caching mechanism, generating a page once and keeping the generated output for a certain time period to serve it to new visitor requests instead of regenerating it on every request. However, there are also other solutions available, called reverse proxy caching solutions, where the web server (or an extension hereof) does the caching and serving without executing CMS code at all.

Naturally, serving a complete page to several visitors will only work well, if the page for visitor 1 is identical with the page for visitor 2. Hence, if the website has visitor login and visitor specific information available, caching the entire page and serving it to other visitors will not work. In those cases, the CMS will be required to cache the common parts and leave the visitor specific parts uncached.

L Operations, support and maintenance

L.1 Response times

The system load varies through the day and week. Response time is particularly important during the busiest hours, the *peak load* periods.

Peak load

1. *100 users work with the content management backend.*
2. *1000 visitors browse the frontend.*

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 submitting a form or following a hyperlink.

Measuring generation time

The generation time is the period from the server receives a request to the processing is finished and the response sent to the user.

In the backend, measurements will be response times, whereas for frontend pages, the generation time is measured, since the supplier is not responsible for the amount of content and graphics on frontend pages. All measurements are made during peak load. The specified times must apply for 95% of the response times in these periods.

Production work through local area network: Measurements are made with a setup corresponding to the daily operation.

The public web part: Measurements are made on a PC connected to the Internet through a 56 KB modem with low traffic on the route to the servers, but peak load of the servers themselves.

It is important that response is so fast that users are not delayed. The following requirements aim at that.

Response time requirements:	**Example solutions:**	**Code**
1. Page response and generation time measurements must be performed regularly.	*The system measures page generation times continuously.* Frequent response time measurements are performed with a stopwatch	
2. Generation time for a frontend page	Generating a frontend page takes __ seconds	
3. Loading time for a backend page	Loading a backend page completely from within the local network takes __ seconds	
4. Time for saving a page	Saving a page takes __ seconds	
5. Time for publishing a page to production server	Publishing a page takes __ seconds	

L.2 Availability

To avoid different opinions on how to define availability, the customer can write his definition and expectations to availability values. It should be considered, how a breakdown is defined, based on the consequences of a breakdown of various lengths. For a very well visited website with many updates every day, a 20 minute breakdown can result in bad publicity, missed sales and employees without access to their everyday work tool. For a small website with weekly updates, a 2 hour breakdown can be a nuisance without further effects on the revenue.

L.3 Support

User support can benefit from standardised systems in the way that some CMS developers have user support hotlines for any user in any customer organisation with a support agreement. This means that much of the load is taken off the shoulders of the consultants, who specialise in developing and implementing the system and usually are not interested in end-user support, and off the shoulders of super users within the customer organisation. Naturally, some support must be given by the consultants, especially when it comes to customised or custom developed functionality, but the "how do I insert a hyperlink/an image on a page?"-type of questions can be handled by the centralised phone support.

L.2 Availability

The system is out of operation when it doesn't support some of the users as usual. The cause of the break down 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 a wrong configuration.
4. Planned maintenance.
5. Insufficient hardware capacity.

Measuring availability

A break down 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 break down period.

Only break downs caused by the supplier's issues are included in the availability statements.

The **operational time** in a period is calculated as the total length of the period minus the total length of the break downs 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 break down, 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 stated regularly. The statement may compensate for the number of users experiencing breakdowns.		
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%).	
3. In other periods the availability may be lower.	In these periods the total availability is at least ___%. (The customer expects 95%).	

L.3 Support

Support comprises help to users, monitoring the operation, and configuration management. The specified times must apply for 95% of the cases. Super users are the ordinary user's first point of contact. This means that the supplier only has to help when the super users cannot remedy the problem.

Support requirements:	**Example solutions:**	**Code**
1. The supplier must help the super users when they cannot remedy the problem themselves. Help covers all equipment and software provided under this contract.		
1p. Problem: Super users cannot 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.	
2. In the period from 8:00 to 18:00 on weekdays, super users can quickly get phone contact with the supplier's support organisation.	*In this period, contact is available within __ minutes. (The customer expects 10 minutes).*	
3. Whether the problem is received by phone or email, the supplier will reply quickly. The reply is help to the super user until the problem is remedied, or a decision that it is a system defect.	*The supplier replies within __ hours in the period from 8:00 to 18:00 on weekdays. (The customer expects __ hours).*	
4. In case of a system defect, the supplier will as far as possible help the super user circumvent the problem, and in addition report the defect to the maintenance organisation.		
5. The supplier sends a support person when this is necessary to remedy the problem.		
6. The supplier can perform remote diagnostics to remedy the problem.		
7. The supplier records data that allows computation of support response time, as well as data for identification and prevention of frequent problems.	The supplier keeps a log of all requests from super users with indication of the problem cause and the time when the problem had been remedied.	
8. The supplier monitors the operation in order to foresee availability problems, and takes steps to change the technical configuration so that availability is maintained.		
9. Supplier provides a single contact for the customer.		
10. Customer is informed regularly about the status of pre-paid support or the outstanding balance		

L.4 Maintenance

When consultants customise the CMS to the organisation, it is essential to have clear established agreements that say that the consultants have to see the project out and correct errors that comes with it.
Additional modules or modifications naturally have to be paid for, and when new modules or functionality are bought, it is important to have a developer that you have confidence in, when it comes to development time, costs and quality. Using a standard method for estimating a project's size, and arranging a fixed price per estimation unit, can give the customer a way to include new developments in the yearly budgets. The requirements template uses function points, because it is based on transactions and user interaction, but there are several estimation methods available for the customer to choose from.

L.4 Maintenance

Maintenance includes defect removal and system changes. The specified times must apply for 95% of the cases.

Requirements for defect removal:	Example solutions:	Code
1. The supplier keeps a log of reported defects as well as change requests.		
2. For all reported defects, the supplier quickly assesses whether the defect is business critical, possible to circumvent temporarily, or possible to circumvent permanently (i.e. reject).	*In the period from 8:00 to 18:00 on weekdays, the supplier completes the assessment within __ hours. (The customer expects __ hours).*	
3. Business-critical defects are removed quickly.	Business-critical defects are removed within __hours. (The customer expects __ hours).	
4. Customer and supplier meet regularly to check the defect assessments, and to decide what to repair or change, and what it will cost.	The parties meet every ____. (The customer expects __ly meetings).	

Requirements for system improvement:	Example solutions:	Code
5. The supplier installs new versions and releases of the delivered software without undue delay.	Installation takes place within ___ days after release of the new version or release. (The customer expects __ days).	
6. Within a period of __ years, the supplier must offer changes at a fixed cost per function point.	*The price per function point is ____ US$.*	
7. Disagreement on the function point calculation must be resolved by . . .		

3 Glossary

CMS	Content management system.
Consultants	Company implementing a CMS for a customer or aiding a customer selecting the right CMS.
Content item	In some CMS's it is possible to publish just a small part of a page, in other CMS's it is the whole page that is a content item.
COTS	Commercial Off-The-Shelf, a standardised software product that can be (imagined) packaged in a box and put on a shelf in a computer shop.
Customers	CMS buying organisation
Developers	Company developing a CMS
LDAP	Lightweight Directory Access Protocol. Protocol used for authenticating users through a centralised directory service.
Supplier	Company supplying a CMS to the customer. Receives the completed requirement specification from the customer and fills out the right side of the tables with their solutions.