8) Requirements specification

General remarks

The requirements specification is a **key reference document**. It **describes** what the project owner/sponsor expects **the outcome of the project** to be (generally a product, as far as we are concerned here). It should express **user needs and other requirements** in such a way that potential project implementation entities can easily understand the **scope and constraints** of the project and respond to the requirements with a detailed plan, including a cost estimate, and, at a later stage, with detailed design specifications of the target product.

As mentioned in the previous chapter, a summary requirements specification is usually provided as part of, or appended to, the project charter, but a more elaborate and complete requirements specification needs to be written before detailed project planning can be undertaken.

If project execution is outsourced, the requirements specification is usually appended to the contract between the project owner and the entity in charge of project implementation (sometimes called "main contractor" or "prime contractor").

Subsets of this reference document may be used, possibly with some rewriting, for particular requirements specifications intended for (sub)contractors to which parts of the project work may be given.

The requirements specification document intended for contractors is sometimes referred to as the "**Scope statement**" or "**Statement of Work (SOW)**".

There are several methods for **collecting requirements** from future users and other project stakeholders: one-to-one or group interviews, focus groups, workshops, brainstorming sessions, questionnaires, etc.

The PM (or whoever is assigned to writing the requirements document) **manages the process of collecting "input" and consolidating it**.
Marketing & Sales ("M&S") should be involved in the requirements specification process, and sometimes provides the PM with a preliminary expression of needs.
(See chapter 18 for more information on cooperation between the PM and M&S.)

In theory, according to some product development life cycle models, the requirements specification should not be modified during the project's execution phase. This recommendation is however difficult to follow in "real life". More pragmatic methodologies, such as Agile Software Development, provide more **flexibility** and even encourage the revision of requirements as implementation moves forward.
A number of development life cycle models are presented in chapter 12 of this guide.

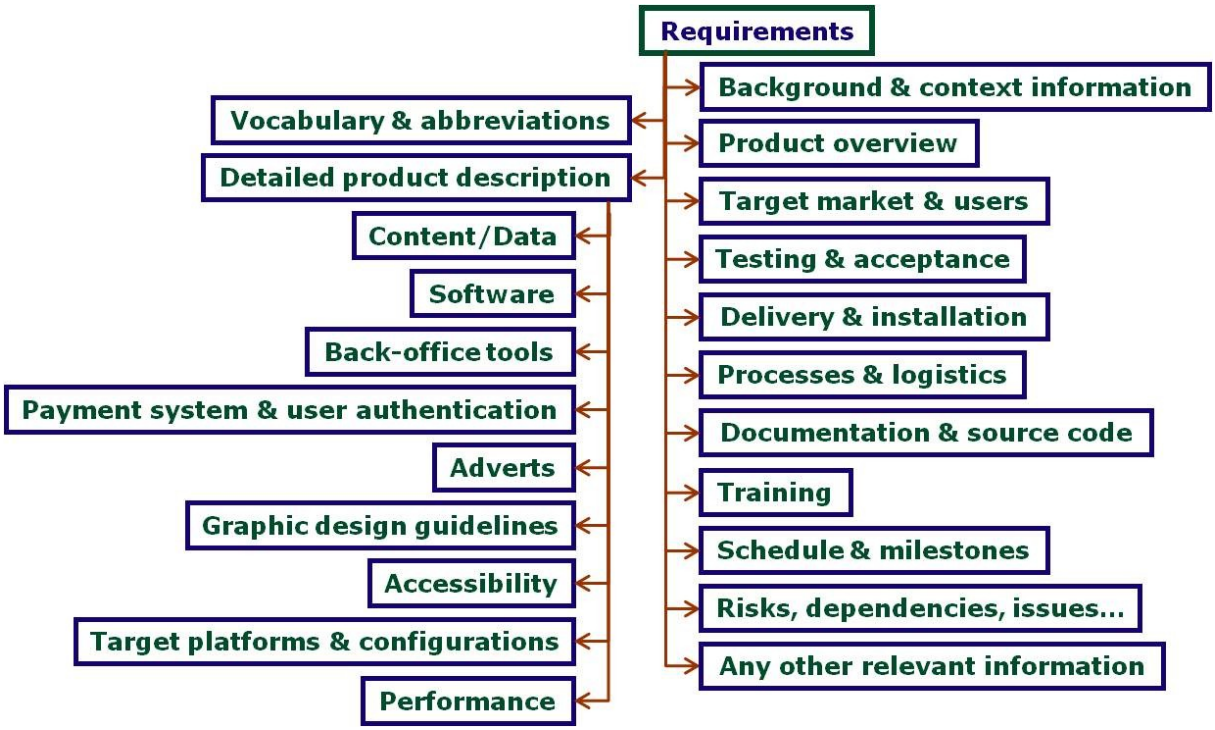
As an example of "built-in flexibility", which is generally desirable, here is an excerpt from the introduction chapter of a requirements specification written by a British publishing company (in 2007):

> *"We have provided as full and comprehensive a description of our requirements as we are currently able, but we do not consider this specification to be complete and final. We expect to work collaboratively with [the Developer] during the design phase for the products to refine and finalize these requirements, where necessary."*

The initial requirements specification and its various revisions must be approved by all project stakeholders concerned.

The **final version of the requirements specification** is a **baseline document** that can be used as a **benchmark for the acceptance testing** of the finished product.

A summary (non-exhaustive) description of the **main components of a requirements specification**, from which subsets may be derived as needed, is given hereafter, assuming that the target product includes content (data), to be provided by the project owner (or an editorial subproject implementation entity), and software (which may be a website), as well as back-office (editing and administration) tools, to be developed by the technical subproject implementation entity.
The list of components may obviously vary according to the specific nature of a project.



Background information

If the requirements specification is to be used as a stand-alone document, for example for discussion and contractual agreement between the project owner and an external project implementation entity, it should include appropriate excerpts from the project charter document, in order to clearly explain the **overall context of the project**.

Such excerpts may relate to some of the following topics:

- context and objectives of the project,
- presentation of the project owner and other stakeholders,
- business model (insofar as it has an influence on the product requirements).

Vocabulary/abbreviations

It is useful to provide, at the beginning of the requirements specification or as an appendix, a list of specific vocabulary (with precise definitions) and abbreviations used in the document, in order to avoid any ambiguity and misinterpretation.

Product overview

A **summary description of the product**, preferably with a **graphical representation** if possible, is useful as an introduction to the more detailed information that will be provided in subsequent sections of the specification.
This overview should address the "problem" that the product is supposed to solve, the proposed solution and the benefits it is expected to provide to its users.

Target market and users

This section should describe the target market and users of the product. The typology of users may indeed have an influence on the product's functionality and design.
Some of the functions and features of a product may be available to all users, while others may be reserved for certain categories of user, for example administrators, webmasters or editors.

The design of a product may need to take into account the age range of users (children, adults, senior citizens…) or the market segment to which they belong (general public, business, education, government…).

The requirements document may include, in this section or in the "Software" section, so-called "**user stories**" or "**use cases**", which are short descriptions of product functions and features from the viewpoint of a user (or category of user).

Detailed product description

Content / Data

A product's content (or data) may consist of **text and multimedia assets**, the latter being still images, animations, videos and sound. If the product is a business application, its **data** may be restricted to **text and numbers** (amounts, dates, etc.), and possibly **pictures, diagrams and charts**.

Content needs to be described at the level of detail required for product development. A section or an appendix may actually be dedicated to the specification of the "**data delivery format**", ie the format in which data will be delivered by the content provider to the software developer.

If content is expected to be **multilingual**, the languages to be supported and possible character set constraints (fonts, encoding) must be clearly specified. More generally, clear and comprehensive **guidelines for localization** should be provided.

It may be useful to group content elements by **type**, each type possibly treated as a specific editorial subproject.

> *The EHM's content was broken down into the following "data sets":*
>
> - *encyclopedic articles (the fundamental content of the encyclopedia),*
> - *definitions (French-language dictionary),*
> - *summary articles (for schoolchildren),*
> - *document excerpts (literature, speeches, etc.),*
> - *atlas (maps and other geography-related media, including data sheets for all countries),*
> - *photographs and drawings,*
> - *animations (some interactive),*
> - *videos,*
> - *sound documents,*
> - *timeline (a graphical panorama and related notes),*
> - *facts & figures tables,*
> - *website URLs and descriptions (linked to encyclopedic articles),*
> - *media captions,*
> - *quiz.*

For each data type, the (approximate) **number** of items, their estimated **size** ("footprint") and **format** should be documented.

The content of the last edition of the EHM on DVD-ROM included approximately 108,000 texts (42,000 articles and 66,000 definitions), "weighing" 100 million or so characters, the texts being illustrated by close to 17,000 multimedia assets on the DVD-ROM (roughly half that amount in the online EHM), some of which, in particular 3D rendering of historic sites, with a footprint of several megabytes each.

Formats were: JPEG for still images, QuickTime for videos and non-interactive animations, Flash for interactive animations and interactive maps, QuickTime (DVD-ROM version) and MP3 (online version) for sound.

Content that is visible to the end user is generally only part of the full set of data in a product. **Tags** and, more generally, **metadata** are usually required, for example for searching (indexes), for content display (style sheets, etc.), as well as for links between elements of content.
Note that the input of metadata and the setting of links, which are often manual operations, are editorial tasks that can be very heavy and time-consuming.

For the EHM, structured text was marked up in XML (initially in SGML), using a very complex "DTD" (Document Type Definition), which was provided as an appendix to the requirements specification.
Furthermore, there were metadata associated to each "document" (the generic name use to refer to an item of content: article, definition, media asset…), and, in the case of long articles or definitions of words with multiple senses, to document sections or subsections. The metadata included one or several topics (subjects), chosen from a "topic tree" with close to 3,200 "leaves", and, if applicable, one or several geographical locations (regions, countries…) and periods (dates), such metadata being used for a number of functions, in particular advanced search.
Finally, each multimedia asset was manually linked to one or several texts.

There is also content that may be called "**secondary**" or "**ancillary**" but which is **nevertheless important** and has to be taken into consideration for both editorial and design/development/integration purposes.
Such content may be for example: help text, credits, licencing and other legal information, sales terms & conditions, contact information, etc.

For example, the online EHM's ancillary content included:

- *home page text and illustrations,*
- *conjugation tables,*
- *a pedagogical guide,*
- *"Help", integrated in the software application,*
- *"About…" (credits, etc.),*
- *"User licence" and other legal information,*
- *commercial pages (presentation of the product offering and fees, subscription process…).*

Note that **elements of the user interface** of a product (eg menus, menu items, dialog boxes, text buttons, tooltips, error/warning messages, etc.) and **metadata** should also be considered as **content** since they need to be created by people with editorial skills.

In the case of a "static" application (eg a non-dynamic website), information regarding content may be complemented with an indication of the number and size of "pages".

If applicable, the specification should also address **content updates and additions**: which areas/items of content are liable to be modified or expanded, to what extent and how frequently.

For a website, more or less simple "**holding pages**" may be required, their content being intended to "whet the appetite" of visitors before the final product actually goes online. Such pages also need to be mentioned in the requirements specification.

Software

The software (in the broadest sense of the word, which includes websites and web-based applications or "cloud" applications) exploits content and makes it available to users by means of a **user interface** and a set of **functions and related features**.

The **description** of the software in the requirements specification may be given at a **high level** (ie not at a very detailed level) but it should nevertheless be **exhaustive**, in that it should encompass **all of its functions and features**, as well as the **main characteristics of its user interface**. The objective is to enable readers to clearly understand what the software should achieve. In this respect, a **graphical representation** of the software's functionality (and mock-up user interface), as well as evocative **metaphors**, may help to make a long narrative text easier to understand.

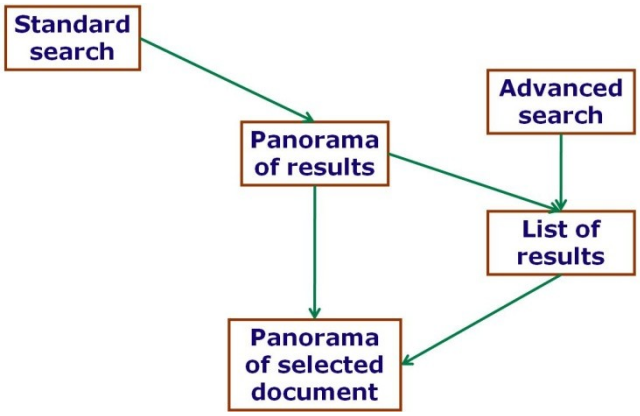
It is sometimes possible to rank functions and features by **priority**, for example in two groups: "**must-have**" and "**nice-to-have**" (ie "desirable but not absolutely necessary"). Doing this allows the implementation entity to divide the response to the requirements into two distinct sets of functions and features, each with a specific price attached.

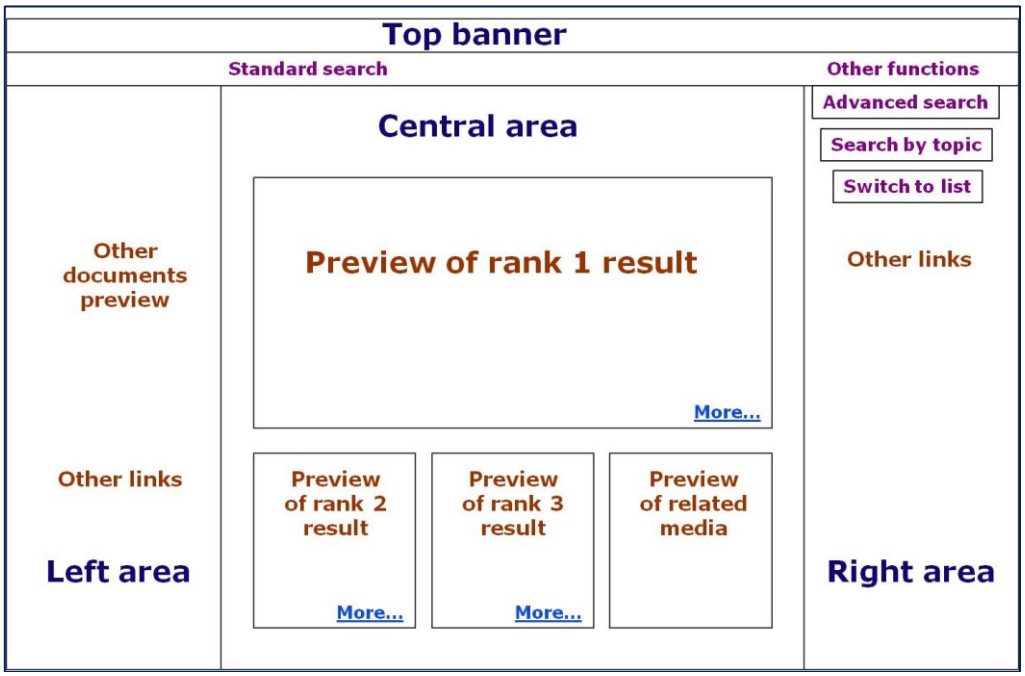
Any foreseeable **evolution** of the software's functions and features, as well as any requirement for **subsets** of the product or **by-products**, should also be documented.

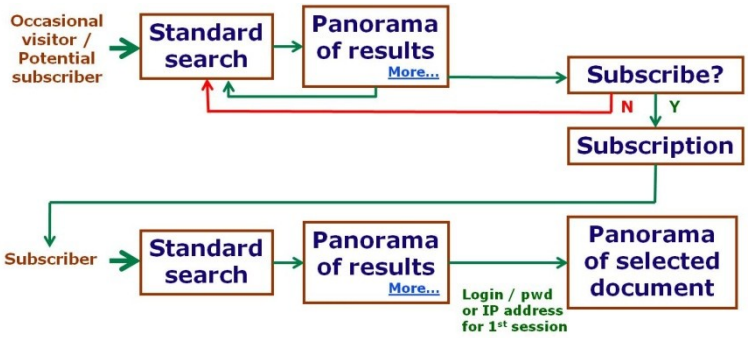
> *For the most recent version of the online EHM, the software's main functions and features could be described as follows:*
>
> - *online multimedia encyclopedia dynamic website, exploiting a content database, with frequent updates and additions,*
> - *search box always present in the upper part of the interface,*
> - *standard search (full text, multiple words, with "wildcard" representing any number of characters),*
> - *presentation of standard search results in "panorama" mode, comparable to the front page of a newspaper, in three areas, the central area occupying half of the total width of the page and dedicated to the preview of most "relevant" results and related media, the left and right columns being reserved for previews and links to related documents,*
> - *presentation of standard search results in "list" mode upon user request,*
> - *presentation of a document, selected from the panorama or from the list, in the central column, the left and right columns being reserved for previews and links to related documents,*
> - *long articles on a single page with table of contents in the left column,*
> - *full-width display of documents as a user-triggered option,*
> - *advanced (multicriteria) search (by topic, location, period, words, with AND, OR, NOT boolean operators),*
> - *presentation of advanced search results in "list" mode (the panorama mode being irrelevant for this type of search),*
> - *hypertext via double click on any word (launches a standard search),*
> - *access by single click to "Help" and "About…",*
> - *access by single click to the conjugation tables and pedagogical guide,*
> - *compliance with the W3C/WAI recommendations concerning accessibility for visually-impaired users,*

- *subset of the encyclopedia (home page, "help", "about…", standard search and panorama of results) available free of charge,*
- *full content and functionality available by subscription,*
- *payment by "Audiotel" (toll phone) or by credit card,*
- *authentication by login/password or by IP address.*

Below are examples of block diagrams and graphical representations of some of the above functions and features of the EHM, including an illustration of the "panorama" metaphor (reminding of the front page of a newspaper), as featured in the requirements specification, followed by screenshots showing how the functions and features were implemented.

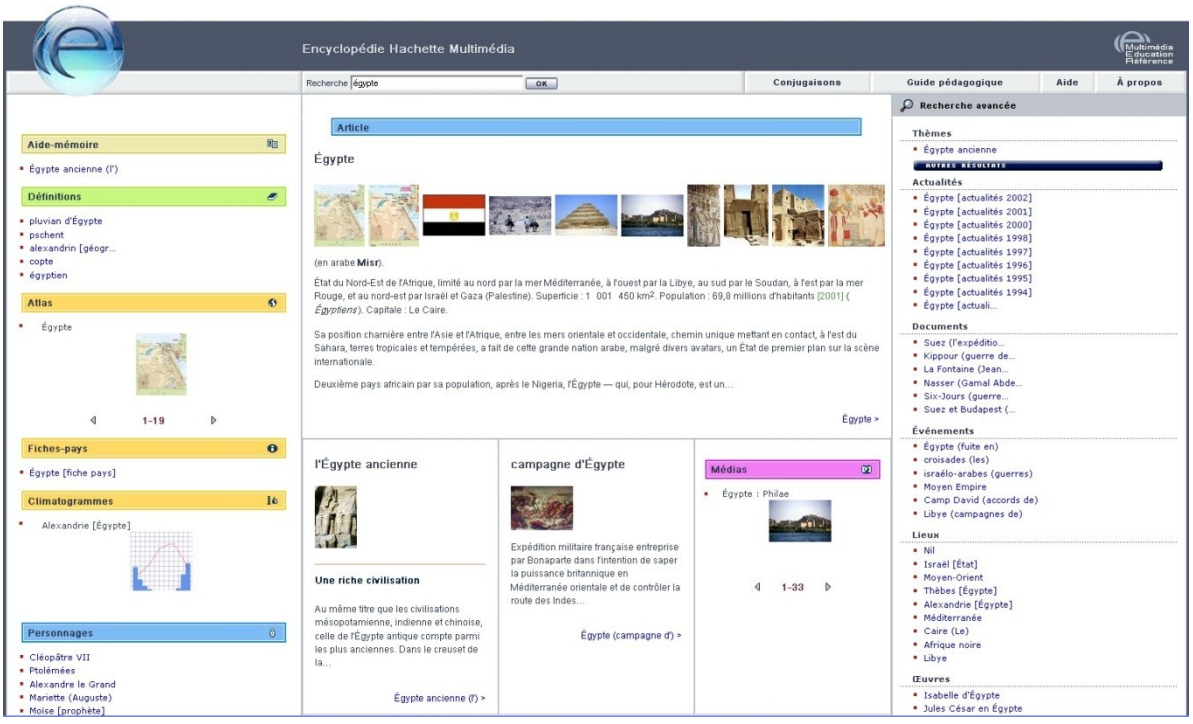




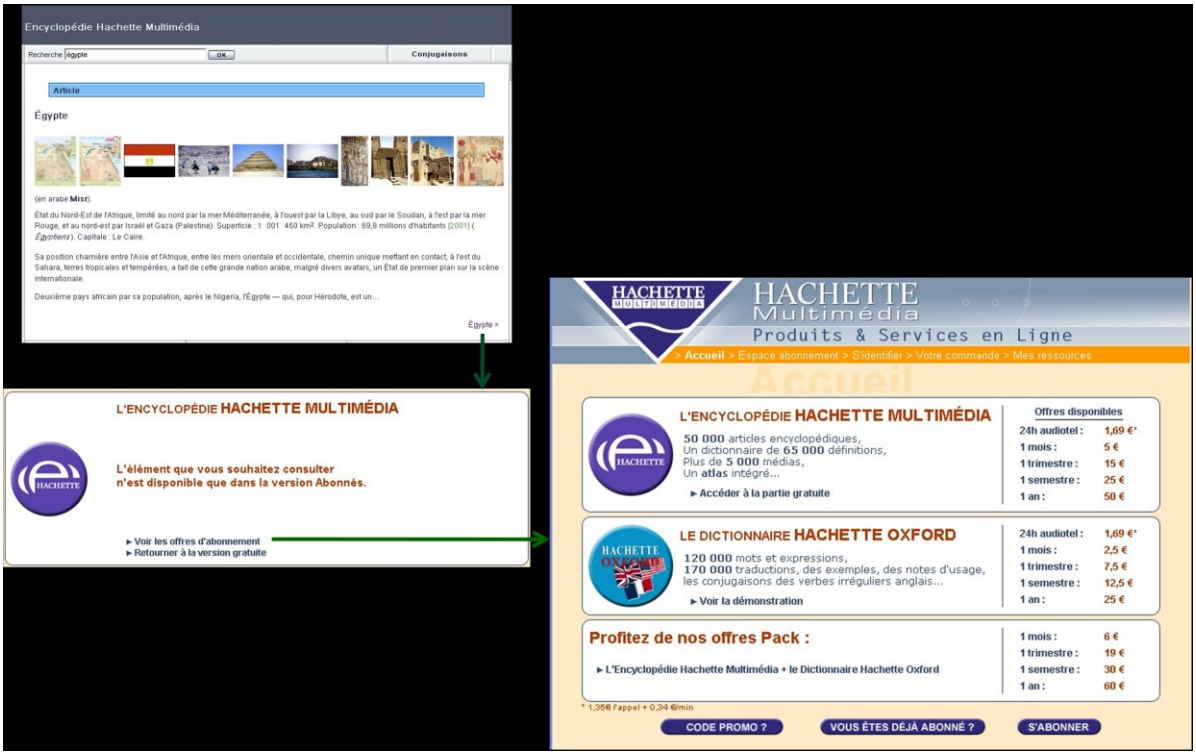


Search the EHM (free of charge):

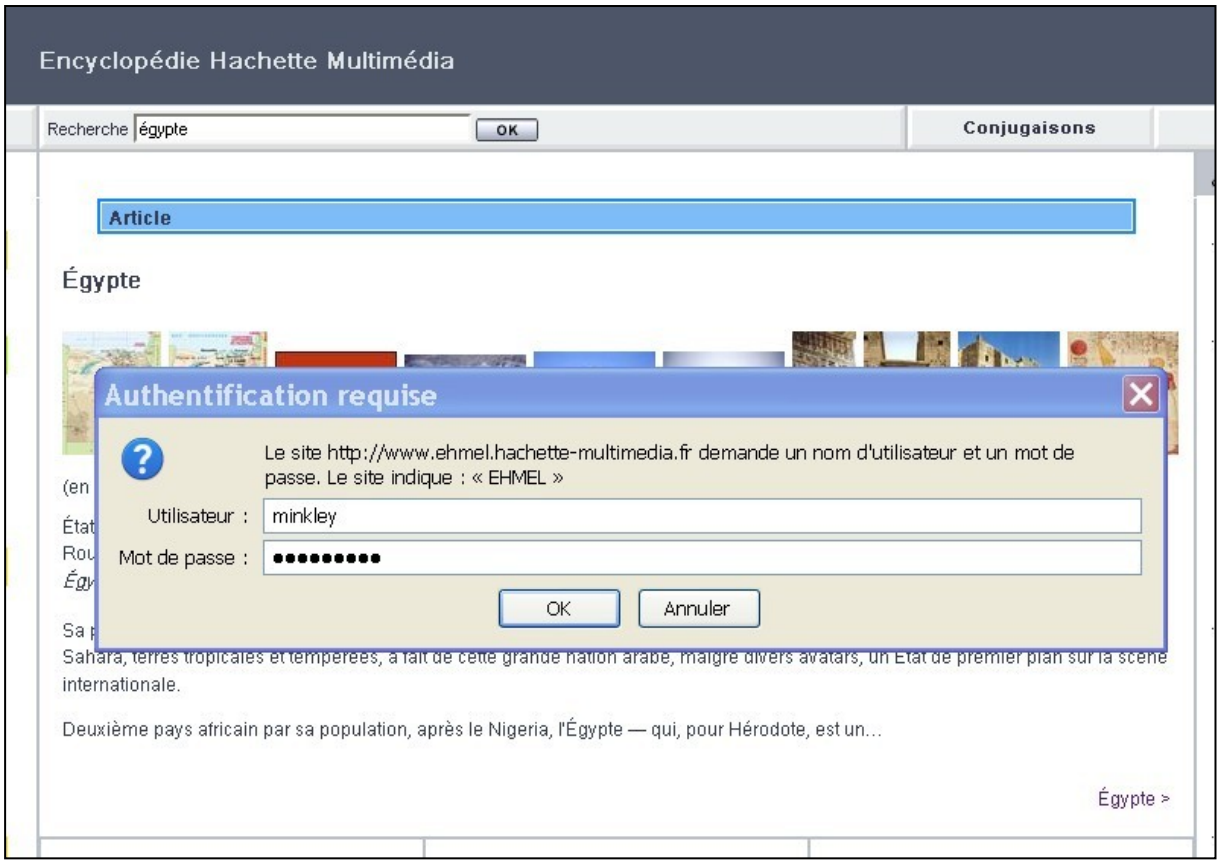
Panorama of results (provided free of charge):



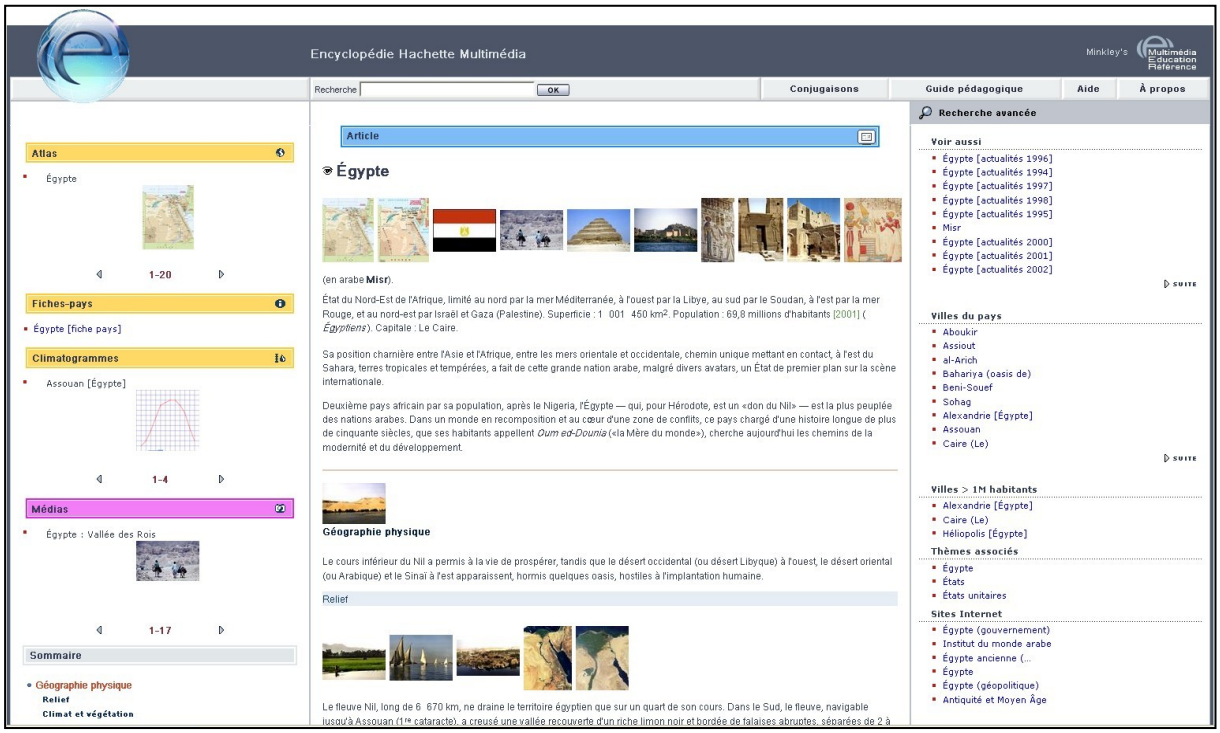
Click on link to display the full content of the main article: if you are not a subscriber, you are given information about the various offers and the option to subscribe or return to the part of the EHM which is free of charge:



If you have already subscribed to the EHM, you need to go through an authentication process (only once per session):



You now have access to the full content of the encyclopedia (starting with the article of which only a preview was displayed as a result of the initial search):



Back-office (editing and administration) tools

For products that include content which is liable to evolve (through updates and additions) on a regular basis, it is necessary to provide **editors**, who are generally **not IT specialists**, with tools (eg a "content management system") enabling them to **manage content** (data and metadata input, modification, publication, etc.) in a simple and efficient manner, **without** any need for help from **developers**.

Furthermore, products such as websites generally require some form of **administration**, usually performed by an administrator or webmaster.

The editing and administration tools form what may be called a "**back office**" (or "back end"). These tool sets, the creation of which may actually constitute a **subproject**, must be specified in the requirements.

The editors of the latest versions of the EHM used a back-office system that was custom-made by the company that developed the EHM software. The system included an Oracle database and content management/editing tools that were accessible via an intranet, using a Mozilla-based custom browser.

Another tool was used after each "data freeze" to extract the content of a new edition of the EHM from the Oracle database, to create all of the indexes required for the various types of search, to format the data for the production of a CD/DVD-ROM or a new online version, and, for the latter, to actually publish the content on the EHM website.

The database and editing tools resided on two servers (one mirroring the other as a back-up). The extraction, indexing, formatting and publishing tools resided on a third, high-performance server.

All necessary functions of this content management system had been specified in a comprehensive and detailed requirements document which was the consolidated result of interviews of editors conducted by the developers, and of workshops, brainstorming sessions and other meetings involving all persons concerned.

The requirements document was actually written by a PM working for the development company, then submitted for review and approval to Hachette (the editors and myself, as PM for the project owner). Needless to say, there were many revisions of the requirements specification before the final version was agreed upon by both parties.

The functions provided by the online EHM administration tool set included user account management (creation, modification, deletion) and usage statistics.

Payment system and user authentication

A payment system may be required as part of the application to be developed. The project owner may have already chosen such a system (PayPal, SWREG, etc.) or may leave the issue open for discussion. However that may be, the payment process and required user authentication mechanism should be documented in the requirements specification, along with the specification of information items that need to be entered by users and stored in some form of database (eg customer information such as name, address, e-mail, login, password, etc.).

A "**single sign-on**" (SSO) mechanism may need to be implemented so that users do not have to identify themselves several times, for example in a situation where a web application requiring authentication is accessed via a website which also requires authentication.

The online EHM was made available to schools via a platform (the "Kiosque numérique de l'éducation": www.kiosque-edu.com) to which users (teachers and students) could log in for access to all of the resources to which their school had subscribed, including the EHM, without any further authentication required.

Some applications do not involve any payment but nevertheless require user authentication (eg an intranet or extranet application, restricted to certain categories of users).
Conversely, there are applications that involve payment but which do not require user authentication for initial access to the website (eg an online store).

Adverts

If adverts are to appear in the product (eg on a website), their desired dimensions and position in the interface should be specified, as well as any requirement or suggestion regarding the advert feed mechanism (eg the use of an "ad server").

Graphic design guidelines

If user-interface graphic design guidelines or recommendations are available, they should be included in the requirements specification.
The distribution of roles in this area should also be clearly indicated. Responsibility for graphic design may be given to the developer or may be held by the project owner, who may choose to subcontract the work to a specialized agency.
Any requirement or preference for fonts and font sizes should be documented.

Mention should be made of logos or other identity-related items (usually supplied by the project owner) which are to be displayed in the user interface in specific locations.

Accessibility

Requirements for compliance of the product with accessibility standards (eg W3C/WAI) need to be clearly specified.

Target platforms and configurations

The hardware and software environments in which the product is expected to function, as well as minimum configurations required for the product to work properly, must be clearly documented.

Such requirements may result from a market study and competitive analysis. They, as well as performance requirements, may of course be challenged by the developers, but once the requirements document has finally been agreed upon by both parties, these requirements become part of the project execution contract that developers must comply with.

> *The latest version of the EHM on DVD-ROM was required to run on configurations specified as follows:*
>
> - *PCs with Windows 98SE, Me, 2000, NT, XP or Vista; PCs with Linux;*
> - *Macintoshes with Mac OS X;*
> - *600-MHz Pentium 3 for PCs (should also work on Pentium 4 and Intel Core Duo PCs);*
> - *500-MHz G3 for Macintoshes (should also work on G4, G5 and Intel Core Duo Macs);*
> - *512 Megabytes of available RAM;*
> - *DVD-ROM drive;*
> - *1 or 2 Gigabytes of available hard disk space (for partial or full installation);*
> - *1024x768 screen resolution.*
>
> *The compatibility requirements for the online EHM were the following:*
>
> - *Windows: Microsoft Internet Explorer 6 (or later), Mozilla Firefox;*
> - *Linux: Mozilla Firefox;*
> - *Mac OS 9: Netscape 7;*
> - *Mac OS X: Safari, Mozilla Firefox;*
> - *Display optimized for a 1024x768 screen resolution.*

The hardware and software infrastructure required for the operation of an information system or product may be unknown to the project owner, for lack of technical knowledge on his part. In such a situation, the project owner usually relies on the project implementation entity to make appropriate recommendations (network architecture and components, server hardware and software configurations, etc.), as part of the response to the requirements specification.

> *The subcontractor used by Hachette for the development of the online version of the EHM determined the type and configuration of the servers required for the product to run properly and with an acceptable level of performance. The recommended servers were procured by Hachette and installed in its hosting service provider's server centre.*

Performance

It is advisable for certain categories of product to specify the minimum speed, throughput or, more generally, the performance of the product to be developed, at least as far as major functions are concerned. Whenever possible, **detailed benchmarks** should be provided in order to measure the degree of achievement of the performance objectives.

The following indications (among others) were given for the first version of the EHM (which was released in March 1998):

- *the scrolling of a long list (including the full list of entries) should be fast and fluid (these "subjective" criteria were not ideal…);*
- *the display of an article should be completed in less than one second after a click on its title in a list;*
- *the display of an article should be completed in less than one second after a click on a word in any text (full hypertext function);*
- *the display of a multimedia asset should be completed in less than two seconds after a click on its title in a list or within an article;*
- *the results of a full text search should be displayed in less than two seconds after the search has been triggered;*
- *the results of a multi-criteria search, including a period search (defined by beginning and end dates), should be displayed in less than three seconds.*

Testing and acceptance

The requirements document should specify how the product will be tested at the various stages of development. The **final acceptance** tests generally use the requirements specification (in its final approved state) as the "ultimate" benchmark. Intermediate and final testing processes need to be specified in terms of the nature of the tests and the product acceptance criteria.

For business applications and, particularly, for complex information system (IS) projects, specific testing phases may be required and therefore specified by the project owner, such as:

- an "**operational acceptance testing (OAT)**" phase, sometimes called "**operational readiness testing**", intended to determine whether the product meets requirements in a real-life situation, but on a relatively small scale, before it is fully deployed and put into service for all of its target users;

- an "**operational health check (OHC)**", intended to verify that the product meets requirements after being fully deployed and put into unrestricted and regular service.

Delivery medium and installation

The requirements for the medium (or media) on which the product is expected to be delivered, as well as installation options, should be clearly specified.

A software product may be delivered for example on a CD-ROM, a DVD-ROM, as a file (eg a disk image) uploaded to a website, on an SD card, etc.
A website may be delivered as such by the developer or may be provided on a portable medium (such as a CD/DVD-ROM), the content of which will then need to be uploaded to an actual site.

Installation options for a CD/DVD-ROM for example may be: no installation at all (the application runs from the disk); partial installation (only part of the product is installed on the user's hard disk, the remaining part being on the CD/DVD-ROM); full installation.

If a CD/DVD-ROM is to be copy-protected, this requirement should be specified, as well as the copy-protection mechanism if a choice has already been made (which may have an impact on development).

Processes and logistics

The operation of certain applications requires the execution of processes (eg a particular workflow for the completion of specific tasks) and the availability of adequate logistics (eg the storage and shipment of goods ordered from a website). Insofar as they are part of a project, processes and logistics need to be covered in the requirements specification.

> *One of the projects I coordinated as a freelance was the design and development of a start-up company's website, which included an online shop for the purchase of goods (eg leather covers for e-readers). The initial version of the shop was limited to the catalog of goods along with a notice stating that ordering of goods would be made available soon...*
> *Version 2 was supposed to integrate all the functions expected of an online shop: catalog, shopping cart, payment (via PayPal), etc.*
>
> *The development work for the online shop was done by a small firm based in Shanghai, China and it was completed with a relatively small number of development-testing iterations, in full compliance with the requirements.*
> *The shop could however not be opened to the public before the logistics and administrative processes had been put in place. The issues that remained to be resolved were the provision of "Sales Terms & Conditions", the process for sourcing goods, the choice of a logistics partner for the storage and shipment of goods, the processes for order management and customer services.*
> *Those issues had not been addressed by the project owner by the time my contract expired (in Nov. 2007) and were still unresolved the last time I checked (years later): the fully-featured online shop had remained in the state of "vapourware"!*

Documentation and source code

In many cases, a project owner will require software that is developed within the scope of the project to be adequately (more or less extensively) documented. Furthermore, the developer may be required to **transfer ownership** of the source code (and the source code itself) to the project owner upon completion of the project. Such requirements need to be clearly specified.

Those requirements may extend to documentation for administrators, webmasters, technical support personnel and end users.

Training

Any requirement for training, as well as training material, to be delivered by the project implementation entity to the project owner and intended for administrators, support people, users, etc., also needs to be properly specified.

Schedule and milestones

Major milestones relating to the project's **deliverables** should be documented in the requirements specification, possibly as an appendix (which may provide a preliminary project schedule). For example, milestone dates may be given for the following events:

- response (from the project implementation entity) to the requirements specification;
- agreement on requirements by project owner and project implementation entity;
- start of implementation (which requires confirmation of the project implementation entity's availability as of the specified date);
- completion of product design specifications;

- agreement on functional design and user-interface specifications by project owner and project implementation entity;
- start of content creation;
- start of software development (usually done in parallel with content creation);
- delivery of sample data (or data subset) to the software developer;
- first "alpha" version of software (integrating subsets of data and functions);
- acceptance of the alpha version by project owner;
- delivery of full data set to the software developer;
- first "beta" version of software (integrating full data set and all functions);
- acceptance of the beta version by project owner;
- first "release-candidate" (or "master-candidate") version of software;
- results of the testing and acceptance of the final release-candidate version;
- delivery of the final "master" to project owner;
- acceptance of the product by the project owner;
- delivery of software documentation and source code to project owner;
- delivery of training to specified target audiences.

Risks, dependencies and other issues

Any relevant risks, dependencies and other issues that may have been identified at this stage of project preparation should be clearly mentioned in the requirements document, insofar as they need to be taken into account by the project implementation entities, who may actually be encouraged to recommend actions or alternatives in order to reduce the extent and impact of such risks, dependencies and issues.

As an example, here is an excerpt from the "Risks and dependencies" chapter of a requirements specification written by a British publishing company (in 2007):

- *"Web platform flexibility to accommodate and provide required functionality for a wide range of products in the future.*
- *Meeting accessibility requirements for all features of the website.*
- *Conflict with other projects: it will be important to assure that this substantial development does not conflict with any other projects or put undue strain on the resources of [the Developer]."*

Any other relevant information

The above-described components of the requirements specification may of course be complemented with any other information that is judged to be relevant and useful in the context of the particular project concerned, for example environmental considerations.

> See the following site for **information on writing a Software Requirements Specification (SRS)**:

>> techwhirl.com/articles/writingsoftwarerequirementsspecs/