Deliver Now! Methodology

For Implementing Websites Using

Kentico CMS for ASP .NET

Kentico
unlimited website possibilities

www.kentico.com

Deliver Now! Methodology

For Implementing Websites Using

Kentico CMS for ASP .NET

Authors: **Miro Remias**
miro@kentico.com
Consulting Services Manager
Kentico Software

Date: October, 2014

Release: Revision 3

Kentico CMS
Right-first-time technology

www.kentico.com
consulting@kentico.com

Kentico Software US
5 Commerce Park North
Suite 202
Bedford, NH 03110
USA
Phone: +1-866-328-8998

Table of Contents

Introduction

If you have ever been involved in web development project management or have implemented a Kentico CMS project, you probably know how important it is to keep in mind all those little details that draw the line between a successful project and an implementation bad egg. To help you avoid pitfalls and uncover potentially weak places in the project during the initial stage of the development process, we have prepared the Kentico Deliver Now! Methodology.

The main goal of the methodology is to provide you—Project Managers, System Architects, Integration Specialists, etc. with an easy-to-follow guide that describes the Kentico CMS project development process step-by-step before the actual implementation takes place. It should give you all the important information you need in order to finish your projects faster, using less resources, matching higher quality products while keeping your budget reasonable.

The methodology is neither attempting to serve as new documentation material nor as the Developer's guide. It therefore does not include the same technical information you can find in other Kentico CMS documentation. Anytime a link to the documentation (available at the time of releasing this document) could be beneficial, it is included, so you can easily navigate to information about the more technical aspects of the particular subject.

Each chapter is introduced by a set of simple questions related to the particular subject. These questions help you think about areas you might forget about. Moreover, based on your answers, you should be able to select only information which is essential for you. In addition, chapters dealing with specific subjects also include some extra materials like checklists or sheet templates.

We hope you will find the methodology a great help, especially when going for your first project, but particular parts should also come in handy to experienced teams.

Miro Remias
Kentico Consulting Group

Process Overview

The life-cycle of a Kentico CMS Project could be described by a simple scheme as pictured below.

Figure 1 Kentico CMS Project Life-cycle

Each phase expects some information, data, and actions as input, while providing a kind of product as the output. It is important to think about this concept as a complex element. If you do not invest enough effort to cover a particular phase, it might influence all the work done during the later stages of the project life-cycle.

Looking at the project from a time-line perspective, it might be interesting to consider how much overall time is spent in particular phases of the cycle. There are many elements included within each phase that influence the total time devoted to that phase to some extent; however, if we consider a standard-sized project, the graph could look like the following.

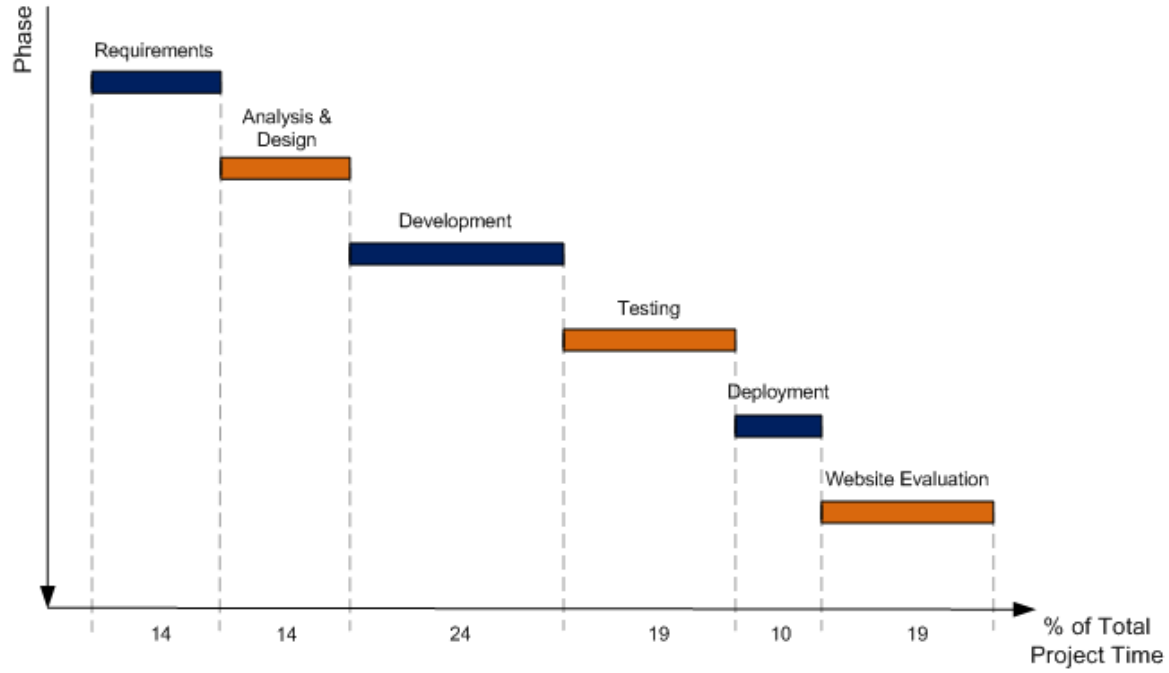

Figure 2 Portion of the total project time reserved by particular phases

You may not completely agree with what is displayed above. You are right; you can't create hundred percent accurate project time-line diagrams that would fit in all cases. Instead, consider it to be the fundamental draft of what you can expect. It isn't really important whether the collecting of requirements would last for 15 or 17 days here. More relevant information stems from the time ratio of phases. For example, looking at the graph you can see yourself devoting more time to the Testing phase or the Website Evaluation phase than you would probably anticipate at the beginning. It is what you should look for at this time.

You may also consider, with respect to the above time-line diagram, how release of new Kentico version and possible upgrade during the development would influence initial project estimates.

1 Requirements

The Requirements phase is where all the necessary information should be gathered. It usually follows an initial stage where the project proposal is provided by the customer. This phase should not be rushed in any way. Try to get as much information as you can from the customer on what the goal of the project is and what functionality is required. All the input data gained at this stage tends to be used during the phases that follow - you will refer to this piece of information over and over again. You should be really thorough during the Requirements phase.

A. What is the goal of the project?

Try to categorize your project based on the size of the website, its purpose, target audience, position on the market, etc. To understand whether it is going to be a simple web presentation, an e-commerce solution, web application working with sensitive data, a complex portal, extranet or intranet solution or something like a community portal is crucial for getting familiar with the customer's needs.

One of the most important discussions you should have within your team, and/or customer, is around balancing requirements on security, performance and usability of the system. It obviously depends on type of project, although, actual goal is common for all types. You do not want to spend too much time dealing with security and/or performance concerns if the web site does not really host and manage sensitive data and/or you expect rather low traffic with not a whole lot of concurrent users accessing the site at same time.

Also, get the information on the motivation behind the project.

- What are the main reasons the customer decided for a new website?
- What problems, weaknesses or blind spots should the new website banish?
- Who is going to use the new website?

B. What is the expected number of users/ page views during the peak load, what is the expected number of pages on the website?

It is important to know the approximate number of users you can expect to visit your website per day as well as the number of page views your website is supposed to handle. Moreover, try to agree with the customer on the number of users able to

login to the member areas of your website. It does not really matter if those users should be end users accessing a secured area on the live site or editors/designers with the access to the administration UI of Kentico CMS. In other words, you want to get information about the load your website would face during the peak-load period.

Ask the customer about the website content:

- Is it going to be static or dynamic content?
- How many pages do they plan to include within the website (hundreds, thousands, etc.)?
    - How many of those are going to serve static content?
    - How many of those are going to display personalized content based on current context?
- Where core data are stored right now, where new data will be stored, and what is migration plan for existing data?
- Does your client plan to create multiple websites sharing the same (re-usable) content?

As you will see later, you are going to build the whole website structure around the information gained in this phase. In addition, plenty of the system settings are more or less dependent on the conclusions that you come to here.

C. What is the structure of the website and what are different types of content being published?

As already mentioned above, you need to gain more details on what type of content will eventually be placed on the website and displayed to the visitors. The goal is to find out whether to use any of the built-in page types or if you need to create custom ones instead.

- Do you have any requirement to build some kind of page/media repository on the website (for pages, video and audio files, images, etc.)?

Furthermore, ask about the customer's ideas and expectations on the website structure:

- What are the main sections the customer wants to divide the website in?
- What items should the navigation contain?

- Is there any requirement for implementing membership areas or restricted sections on the website?
- Do they plan to display personalized content (changing the layout, styles and look of the page based on the current user)?
- Should website visitors have an option to define content of personalized areas themselves?
- Are visitors able to contribute to the website content (e.g. blogs, press releases, articles, comments, etc.)?
- If you plan to deliver website in multiple language mutations, how do you intent to handle selected language info in URL (if at all)?
- What is the default URL format requirement? How the friendly URLs should look like?
- Do you have any plans for content sharing between multiple websites?

D. Which web standards should be followed in terms of accessibility, and coding?

When building website with accessibility in mind, you want to comply with some specific accessibility standard, or better yet, a group of such standards. It ensures your website is available for everyone including people with different kinds of disabilities. Industry standards suggest implementing requirements defined by the Web Content Accessibility Guidelines (WCAG), the User Agent Accessibility Guidelines (UAAG) and/or Section 508 (applies to the United States). Visit the W3C accessibility page or Section 508 home page for more details on these standards.
Similarly, consider following coding standards available. In particular the HTML, XHTML, XML based standards, mobile client standards, web service standards, etc. Since responsive design is being buzz word recently, driving a lot of trends in design and framework technologies, you also want to look and possibly use HTML5. While modern browsers these days provide quite rich support for HTML5, you still need to ensure client-side technology your target audience uses, is capable of proper processing and rendering of HTML 5 pages (it makes no sense to use HTML5 if you need to support IE7). Please make sure you check the currently supported browsers for Kentico in this context.

E. Who is the target web site visitor?

When it comes to website visitors, using devices/tools they are using to access the web site we recognize three basic groups of visitors:

- Mobile device users (older internet-ready devices, smartphones, tablets, embed devices, etc.),
- Full (desktop) browser users,
- Automated tools (internet crawlers, scraping robots, etc.).

There is no doubt today, the next big thing in web industry is mobile. Anything mobile is taking off like never before, and this fact alone will most likely drive decision to support mobile devices in some way. You should therefore answer following questions:

- What would you say is expected ratio between different types of visitors accessing website (mobile vs. desktop vs. robots)?
- What type of mobile devices should the web site support? How will website display for different categories of devices?
- What strategies to support mobile devices you are looking at?
    - Do you want to go all the way with responsive design?
    - Are you looking to create separate version of the website used for mobile?
    - Do you want to use server-side mobile device detection and layout mapping to address different devices?

F. Which products and technologies will be used?

If your project requires integration with some 3rd party components (Telerik, ComponentArt, PlusSuite, etc.), external software (ERP, CRM, custom .NET applications, etc.) or some other technologies (Adobe Flash, Microsoft Silverlight., etc.) you should keep track of these as well. It will play an important role when you start to work on the development plan. Based on the type and complexity of the integration, your development stage may shrink or expand respectively. Ask your customer what functionality they want to use.

G. What is the content life-cycle? Who is responsible for the content management?

Editors and Contributors

Think about the content life cycle for a while:

- What type of users would be responsible for creating, updating and deleting pages?
- Should website visitors get a chance to contribute to the website?
- If the content will be subject to workflow, what steps should it go through before being published?
- What roles would be in charge of approval in each particular step?

Integration

- If you plan to integrate with 3rd party CRM or ERP system, is goal to synchronize changes from Kentico to the external system?
- What task needs to be accomplished in the external system as a reaction to page / object changes in Kentico?

Staging

In some scenarios the content gets entered and published on the staging server first and then pushed to the live site using synchronization.

- Is your customer interested in isolating the staging and production (live) environment?

Archiving

You have created and published the website content. Imagine the website is running for some time now and you decide to archive pages. Talk to customer and decide on what needs to happen with the content once it passes its life-time.

- Should you remove such content from the website completely?
- Or should content be archived instead, and kept within the content repository?

H. What languages will be used for the content?

The amount of work involved in developing the website obviously grows with the number of different language cultures your project needs to support.

- Talk with the customer about what countries they plan to expand to with the website,
- How different content will be in various language versions?
- Is client looking for simple content translation only? Or is there a plan to have different layout/functionality provided by various language versions?
- Will translation happen manually, using on-line translation services (Google, Bing) or through the translation agency?
- Will look and feel change based on the selected culture?
- What about website structure and page hierarchy in various cultures?
- Does the client need to translate the administration interface UI?
- Is the required UI language available for download?

I. What is the required availability of the website?

As with other on-line solutions, your project might encounter some difficulties running on the live server due to various reasons. It does not need to be related to the Kentico CMS website implementation or your custom code at all. It might be an issue with a web server, SQL Server, network, data center availability, etc. You therefore need to talk to your customer about required availability.

If you are not familiar with the meaning of the word 'availability' in web terminology, take a look at the simple introduction post. Basically, 'availability' expresses a percentage of uptime (site is running and serving user requests) and downtime (site is unavailable) in a given year.

High availability (99% to 99.9999%) could be achieved and managed by facilitating advanced scenarios within your solution. You can consider including RAID fields to provide a disk striping or disk mirroring extension to your storage environment. Furthermore, you may want to consider enhancing the live environment setup to support Failover Clustering in order to provide means for automatic detection of server node failure. It could cover both web servers as well as SQL server nodes. On top of failover clustering, the Log Shipping technique may take place to support failure resistance and log propagation between multiple nodes. Other useful

methods for gaining better availability include SQL Server Replication, SQL Server AlwaysOn and often neglected Backup & Recovery.

Last but not least, you should consider running the web site in cloud environment like Windows Azure or Amazon EC2. These technologies are providing SaaS and IaaS solutions that by definition provide high availability, failover and redundancy features. On top of it all, these solutions allow for fast and easy horizontal scaling in case of sudden increase of load on the servers. You can also use Kentico+ which is an Integrated Marketing Solution provided as a service in the Windows Azure cloud and managed by the Kentico team.

J. What is the production environment type?

As you already know, you can run the web site on shared hosting, dedicated server(s), in your own environment (on premises) or eventually in a cloud. Decision where to run the web site should be discussed with the customer before the project starts in the analysis phase. Of course, in many cases such decision is based on the available project budget, and unfortunately often later rather than sooner.

Ask following questions before you move forward with your development:

- Where is your environment located? Is it shared hosting, cloud, dedicated server?
- What project type you plan to use (web site project, web application, Windows Azure project)?
- What are known environment variables (trust level, pre-compiled web site, data storage and transfer costs, etc.)?
- How often you expect to implement and deploy new features? What is the price of the development and deployment in such environment? What it takes to deploy changes to selected environment?
- Do they plan to use CDN network to serve files?

Cloud services typically allow for cost cuts comparing to standard on-prem solutions. Implementing infrastructure ensuring HA (let's say 99.99% or two-nines) would involve immense initial costs, often many times higher than costs of setting up same environment (or comparable one) in the cloud.

Take for example environment consisting from development server, QA, staging server and production with two data centers for redundancy and performance

reasons located in different geographical regions. Assuming that each environment consists of web farm with two nodes, and SQL cluster underneath, we are easily looking at $50k to $200k for the whole setup. That is just the initial costs for HW and SW. You have to count with ongoing maintenance costs as well. Energy consumption, data center equipment, HW and SW maintenance, upgrades, patches, that all result in additional costs. Same environment in cloud, depending on specific solution, and depending on actual utilization of such environment, may cost from $1000 to $2500/month. You do not have to worry about redundancy, failover instances, load balancing and other features you have to support yourself on-prem, as cloud solutions typically incorporate these automatically.

Cloud has its own downsides though. You are paying for data transferred between nodes in the cloud, cloud and on-prem/external systems, storage, and so on (however, these fees are accounted for in above estimates). Regardless, you have full control over all variables influencing costs of infrastructure in cloud, and thus the cloud would be preferred option for many clients and projects. The key is to realize that cloud IaaS solutions allow you hand off a lot of responsibilities and complexities related to infrastructure configuration, maintenance and support to typically very reliable and flexible 3rd party solution.

That being said, if you are architecting infrastructure for small or medium size project, without any or only limited requirements on HA, you may be still better off with on-prem solution. Especially if you already have an existing infrastructure in place and you do not need to invest a significant amount to incorporate additional elements into the infrastructure.

Please refer to Appendix A – Requirements Template to get a list of all questions mentioned in the text above.

2 Analysis & Design

The Analysis & Design phase takes all the information delivered as the output of the Requirements phase. By applying various techniques, it transforms the requirements into so-called design units — use case diagrams, wireframes (Appendix C – Website Wireframe Example), the website structure draft at the beginning and page templates, page types and physical file organization at the end.

2.1 Content tree

Designing website content structure is one of the most important items on the project to-do list. The structure of the website has an impact on several aspects of the final version of the project.

A. Content structure and URLs and SEO

The organization of pages within the content tree defines the URL format of your pages. The page URL is formed from the page alias (unique name of the page) and the page aliases of all parent pages. During request processing, an SEO friendly URL gets rewritten internally and resolved to a physical ASPX page rendering content.

- Design content tree structure to get URLs in the required format at first place, without need to modify default URLs later on,
  - **NOTE:** You can also use extension-less URLs,
- Utilize wildcard URLs to use a single page for displaying the content of multiple pages to comply with SEO standards.

B. Content tree and mobile devices

There are various options how to build a web site optimized for mobile devices. One of the older and slightly outdated approaches relies on having a dedicated section of the website optimized for mobile devices. Pages in such section apply different CSS styles (optimized for smaller screens), display fewer images, only limited amount of content and so forth.

- Re-use existing data/pages and manage the data from one place,
- Define navigation items for mobile web site.

The dedicated section approach was heavily used some 2 years ago. These days there are more sophisticated ways on how to deal with mobile support (server-side device detection and layout mapping, client-side JavaScript frameworks, responsive

design, and others). You can find out more about developing web site for mobile devices here.

C. Content tree and archived pages

In order to keep outdated content under control consider hiding content from live site, moving it to dedicated location or archiving it with automated (advanced) workflow.

- Identify the period after which content should be archived,
- Automate the archive process with the advanced workflow and optionally move pages to a dedicated content tree section,
- Archive pages with a built-in 'archive' workflow step.

D. Content structure and navigation and sitemap

There is a close relationship between the content tree structure, navigation elements and the sitemap. You can decide what content tree nodes will be displayed as navigation elements in the website navigation. The sitemap is generated based on the content structure, ready to feed the search engines.

- Define the desired navigation elements,
- Define which pages should be excluded from navigation, internal/external search engines and sitemap.

E. Content structure and membership and restricted areas

To create any membership or restricted area within your website, include the area parent page into the content tree first and then place all pages underneath.

- Force authentication when accessing the area root page and subpages,
- Do not forget to setup the page level permissions accordingly.

F. Content structure and system pages

Decide whether system pages (below) should be stand-alone pages in the content tree, updatable via the CMS, or physical ASPX pages managed by developers through Visual Studio instead:

- Logon page & Reset password page,
- Page for invalid/ non-existing pages (404 Page Not Found),
- Access denied page.

G. Content structure and website performance

Understanding the way pages are retrieved from the database and displayed on the live site could help you catch on to the importance of designing the website structure the right way. The more pages exist under a single parent page, the more resources are used when searching for the right content.

- Keep content categorized/ grouped/ structured using some rule (e.g. grouping content based on release month, etc.),

If you ever need to alter the default URL path (URL) of a page, use the custom page URL path at first. You can change the URL path through custom page aliases as well; however, it would introduce greater overhead during the request processing compared to a custom URL path.

- Use custom page aliases only in case you need to specify more than one URL per page,
    - **NOTE:** Try to eliminate extra round-trips (redirects) whenever possible,
- Always enter the primary alternative URL (one used the most) using the custom page URL path.

Sometimes you need to use custom ASPX page located in Kentico project folder on file system. URL of such physical ASPX page is by default handled the same way as if it would be page stored in the content tree. If you create an ASPX page in a folder, which is not excluded from Kentico URL processing by default (e.g. */CMSTemplates*, */CMSPages*, */CMSWebParts* etc.), or excluded via Kentico system setting, processing of such ASPX page will use website resources that could be otherwise used to handle other page requests.

- Exclude custom/system pages (ASPX) from URL processing via 'Excluded URLs' setting or locate them into folders excluded by default.

NOTE: If you want to know more about URL processing and how it can be influenced by content structure, take a look at the following webinar.

H. Content structure and sharing the content

If you plan to use a Kentico CMS instance containing multiple websites sharing the same content, you might want to use linked pages to create a copy of the original page on the target website or in other area of the same web site.

Most of the changes in the settings of the original page (page template settings, page URL path, content changes, metadata, categories and tags etc.) are reflected by all occurrences of the page on other sites. However, some of the settings like page level permissions, related pages and page aliases are not shared by the original and linked pages.

If you want to share only the page content and not the page itself, you may use viewer web parts (repeaters, data lists, etc.) to pull the data out of the source website and display it on the target site.

- Define which pages need to exist on multiple websites or in different sections of the same web site,
- Identify the content displayed by multiple websites.

I. Content structure and multilingual sites

You can specify different content structure (not a content tree structure) for each culture assigned to your website. You could either translate pages from the default culture or create completely different content instead. Page specific information is unique for the given culture. To meet your SEO plan, you can use culture specific URLs for the same page.

- Translate pages from the default culture or create new pages for a specific culture,
- Use translation services to automate the translation process,
- Use different URLs for pages of a particular culture to keep up with SEO standards.

J. Content structure and media files

If you plan to upload a large number of files onto the website, try to avoid using the content tree as storage for files. Otherwise, the content tree could grow immensely resulting in overall performance degradation.

- Store files in the content tree only when:
    - You are able to control the number of files uploaded,

- Content tree features like workflow, multilingual support, page-level security, and tags/categories are required.

Read more about storing files in Kentico CMS in chapter 2.5 Organizing media files.

K. Content structure and user contributions

To allow visitors to contribute to the website, you may want to setup the User contributions (Wiki) module.

- Specify the target location for newly created pages and the source path for the pages available for editing,
- Combine membership areas and the Wiki module to allow only certain users to manipulate the website content from the live site.

2.2 Wireframes

Once you have built comprehensive web site content structure, you can move to the first step in the design process. Wireframes allow you to design approximate representations of pages directly in the content tree as part of your pages which can be later used by web site developers/designers to turn them into real pages. They help developers and designers build a shared understanding of site functionality before spending time and resources on design and development. One of the major advantages of wireframes feature built into Kentico is that they are easy to use and managed from within the same system.

- Design wireframes for unique pages and keep them updated.

Figure 3 Example of a wireframe template in Kentico.

2.3 Page templates

Next step in the design process is to turn the wireframes into real page templates. Each page in the content tree is based on a page template. A page template defines the look and feel of the individual pages - we say it defines page layout. A template also contains elements (web parts) responsible for content displayed on a page or other functionality include on page. The way content is displayed on the page is also affected by the content tree hierarchy.

A. Design template analysis

You have got the design template, the index HTML page and the related CSS style sheet. Process the template in the following way:

- Mark the parts of the design template that should be shared across the website — use them for the master page template,
- Try to find some general layout pattern by analyzing the position of the content elements — two columns, two columns and menu on the left, three columns with a top submenu, etc.,
- Split the design template into zones displaying some kind of content—page data, images, banners, navigation, etc.,
- Make sure the content zones from the previous step fit the layout identified at the beginning.

B. Page templates and zone types

The sole purpose of web part zones is to provide a placeholder for basic building blocks of the page – web parts and widgets. A zone can be of different types. Each zone type serves a slightly different purpose.

- Define what zones should provide content editable by page editors (on the *Page* tab), designers (on the *Design* tab), group administrators or registered users on live site.

C. Page template and the website structure

As far as page template inheritance goes, you also need to think about the way parent pages affect the look of the underlying pages.

- Pick zones from the previous example which should be displayed on all pages (defining the root template) and zones that will be part of child pages (page templates).

D. Page template and web parts

Try to categorize the content displayed by the page template based on the content type.

- Go through the content zones outlined earlier, and define the type of content they display — image, calendar, logon form, current user data, banner, rotating news, newsletter subscription, blog preview, forum, polls, etc.,
- Search through the Kentico CMS out-of-box web parts and modules to find out whether any of the existing web parts could be used for displaying such content,
- If the zone displays very specific content, and there is nothing out-of-box that can be used to display the same content or emulate the same functionality—make a note. You may need to develop a custom web part for this one,
- If you are not sure whether the built-in modules and web parts fit your needs, do not hesitate to consult your requirements with the Kentico team.

Take a look at the sample result of design template processing in Appendix B – Design Template Processing.

NOTE: If you want to know more about how to properly design the web site from scratch, take a look at the following webinar.

E. Page templates and personalized content

To display personalized content to the user, you do not need to create multiple pages. You can setup web parts to display content in personalized form using current user information.

F. Page templates and mobile devices

Mobile device detection allows you to setup different layouts per specific mobile devices, group of devices or simply based on screen resolution on the page template level.

- Define mobile profiles for devices (tablets, smartphones etc.) you plan to support,

- Design layout patterns for each of the device profiles.

NOTE: If you want to know more about mobile development in Kentico, take a look at the following webinar.

2.4 Page types

Structured data are represented by page types. All pages in the content tree are of some page type. A wide range of pre-defined page types is available for you off the shelf. Regardless, you can also create custom page types or modify existing ones according to specific business needs.

Page content is displayed on live site using transformations defined for the respective page type. Transformations are templates used to transform raw data (retrieved from specific data source, like database, web service, XML file, etc.) into meaningful HTML code rendered on the live site. Transformations are used by all viewer web parts displaying content on pages.

- Make a list of all different content types displayed on the website – news, article, blog post, press release, product, etc.,
- Go through the existing page types — search for ones that fit,
- Before you decide to go with page type and content tree route, make sure that using custom table would not be a better choice,
  - Read more about custom page types and tables in C. Page types, custom table,

A. Page type and security

Each page type relies on its own security settings. Page type permissions allow you to control access to pages.

- Make a list of roles authorized to manage the pages of a specific type—read, create, edit, delete, etc.

B. Page types and field inheritance

When creating custom page types, you may realize some fields have same definition (type, size, input control type, etc.) as existing page type. You do not want to manage same kind of fields for multiple page types when these fields can be defined once, managed from single location, and shared by multiple page types. That way you

make changes to field once, and all other types will inherit it immediately. You just need to create a page type (consider its parent) defining all common fields, and let other types inherit it. This can be achieved at the second step of creating a new page type.

- Make a list of page types with similar fields and come up with general page types holding such fields.

C. Page type and workflow

Workflow support allows you to define and manage a content authoring process. Versioning support allows you to store, view (audit) and roll-back previous versions of the content. Additionally, archiving support can be used to archive pages. Archived pages are no longer available on the website, but are still kept in the content tree. You may re-publish archived content later at any time.

The basic workflow page life cycle is depicted below.

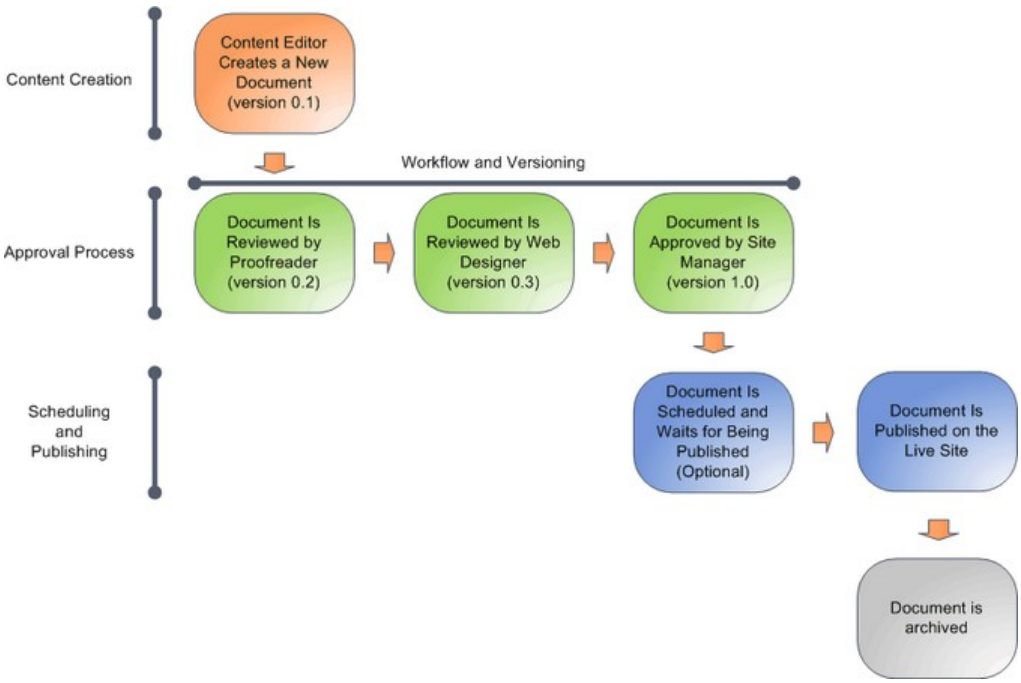

Figure 4 Example of a page life cycle when using workflow

Prepare custom workflow:

- Specify content life-cycle, list steps the page goes through during its life-cycle,
- Define roles or users authorized to modify, approve or reject the page in each step,
- Workflow scopes allow you to define content included in the workflow, either manually or based on a custom macro condition,
- Consider enabling the content locking feature in order to prevent unintended overwriting of content changes,
- Decide whether un-published content should be archived or deleted completely instead.

D. Page types, custom tables and custom classes

All pages are stored within the content tree. The tree structure may become inefficient for really large numbers of pages though. Thus, if your website is likely to contain hundreds of thousands of pages, you should use the custom tables application to store the data of a particular content type instead.

- Should the expected number of pages for the all sites running on the same Kentico instance combined exceed 150 thousand consider using custom tables,
- Use custom tables to store content that does not rely on the tree hierarchy,
- Use custom tables in case features of the content tree like workflow and versioning, page level permissions, metadata, tagging and categories, related pages (and others) are not required,
- You can define custom tables in the same way you define custom page types — by specifying custom fields, transformations, etc.,
- Custom tables can definitely hold more records than the content tree.
- Alternatively you can also use custom classes as data storage. They can be created under custom modules or under the *Custom* module. These classes may also represent relationships between other classes. Data manipulation within these classes can be achieved using the automatically generated API.

E. Page types as containers

A page type can also serve as a container. A container page type does not contain any custom fields. The purpose of such container page types is to serve as placeholder for custom SQL queries and transformations which may not be related to any specific page type. You can also use such container page type to

categorize/group content of the content tree to achieve the desired web site structure.

- Create container page types to hold transformations and queries which are not related to any other page types.

2.5 Organizing media files

It is really important to choose the proper way of storing media files on the website. The proper storage type depends on the nature of the file (e.g. design image, media gallery image, media repository file, etc.) as well as the expected number of files and desired functionality.

You can specify where your files should be stored — in the database, file system (local storage, file server, Amazon S3, Azure Blob storage etc.) or even both. If possible you should store files in the file system. It gives you a performance boost; however, you must provide an adequate amount of free disk space.

You should consider the following:

A. Files in the content tree

Files are uploaded and stored within the content tree as *CMS.File* pages. Each file is represented by separate record in the content tree. The size of the content tree expands with every file uploaded into the system. As mentioned earlier, if the content tree is not maintained properly, and thus gets too large, your website may encounter a performance slowdown.

- Use the *CMS.File* page type for storing files in the content tree if you have control over the files uploaded by editors,
- The *CMS.File* type usually stores files used in multiple places on the website
  - **NOTE:** Best practice would be to store all website design files in the appropriate App_Themes sub-folder,
- Retrieving a file stored as a page in the content tree requires additional resources,
  - **NOTE:** Use content tree as storage for files only in case you require features like workflow and versioning (versioning is also available for media files), page level permissions, tagging and categories, related or linked pages etc.,

- The maximum size of an uploaded file is limited by the maximum size supported by the SQL Server binary type and request length/size,
    - SQL Server is limitation only if you are storing files in the database,
- Do NOT use it for creating a file repository containing a large number of files,
- You can use the bulk import tool to upload multiple files at once.

B. Page attachments

Files can be stored as part of a structured page in the form of attachments. This way, you can attach multiple files to a single page. Attachments are directly bound to their page and follow its life cycle.

- Files attached to a specific page are available only for this page,
- Attachments inherit page level permissions configured for the page,
- You can specify multiple files for a single page with no impact on system performance,
- There are several types of attachments—grouped, unsorted or file field attachments,
- The maximum size of an uploaded file is limited by the maximum size supported by the SQL Server binary type and request length/size.

C. Media libraries

If you plan to create a file (asset) repository available to site visitors on the live site or you are expecting a large number of files (or large files) to be uploaded to the website, you should consider using media libraries as a storage option.

- All files are physically stored in the file system,
- No limit on the number of files uploaded through media libraries — limited only by free disk space,
    - Binary data of the files is always stored in file system,
- You can point a media library to an FTP location and that way upload new media library files,
- You can create custom smart search index which will index content of media library files,
- Files can be organized into hierarchical structure with folders,
- Files are accessible using secured direct path,
    - **NOTE**: Library permissions are applied on per library level, not for individual files,
- Media libraries feature a rich management UI,

- The files can be accessed from various parts of the system and used by multiple pages,

D. Media files support versioning (versions and binary data stored in the database)**Meta files**

Some system objects (like products, web parts, banners, manufactures, payment options and others) allow you to upload metafiles (thumbnails images and so on). If you want to use metafiles in your custom modules/objects, you can manage them with the Kentico API.

E. Unmanaged files

Some files, even when not served by the CMS, can be still managed from within the UI. You may store them in the web project theme folder as a part of the website design template, as a part of some component (web part, web part container, page template layout, etc.) or in the script folder (so that files get exported along with the website) and can be accessed via a direct path. Unmanaged files are typically various design or functional files that won't be used for any purpose other than functional (JavaScript) and styling (including flash animations, etc.).

- You can manage custom JavaScript files in *Administration → Javascript files* application.
- Some components or objects (CSS style sheet, web part, layout, page template etc.) allow us to manage the files on the *Theme* tab available in the object management UI.

NOTE:

- You can use the built-in image editor to perform simple updates to your images (e.g. crop, resize, rotate etc.),
- In case HTML5 is supported on the client machine, you can upload multiple files at once in supported locations.
- Changing the storage location (DB vs. FS vs. both) via website settings will not automatically move/copy the binary data into selected location upon such change. This happens only on demand whenever the page attachment or metafile is requested.
    - Optionally, you can manually re-locate files binary data via file management section in *Administration → System → Files*,
- Page attachments, media library files and metafile can be synchronized using content staging or the web farm synchronization feature, and they are also handled by the import/export module.

- You can edit certain file types (.docx, .xlsx, .pptx, .jpg, .jpeg and more) using local editor tools (like MS Office) if WebDAV support is enabled,
- Custom tables module dos NOT support file(s) fields.

3 Development

Once you pass the Analysis & Design phase, it is time to get down to the actual development. This chapter discusses the most common actions and tasks concerning the development phase.

- First, you should make sure you have all necessary resources available on the team – 3.1 Resources
- To ensure a smooth and trouble-free development process, you may want to setup a team development environment as described in chapter 3.2 Team development,
- You can install a new instance for the development environment using Kentico Installer,
    - **NOTE:** The Kentico Installer lets you select the new installation to be a standard website project, web application or a Windows Azure project. Make sure you should pick the suitable project type at the beginning as later, any changes of the project type are only possible via a manual procedure,
- You usually start the development by processing a design template delivered by a graphic designer – 3.3 Index template processing,
- Next, you process the static HTML and decide on the proper ways to display content, but this time using the CMS – 3.4 Altering static HTML using web parts,
- You create templates for your pages afterwards – 3.5 Page template development, Structured content will be stored using page types. You can either use pre-defined page types or create custom ones –
- 3.6 Custom page types,
- If the out-of-box functionality does not cover all your needs, you may need to develop custom modules or web parts – 3.8 Customization options,
- When the content and presentation is ready, you may approach website security configuration – 3.10 Security,
- Finally, as a last step, review website's configuration and settings to optimize it for the best performance, scalability and stability.

3.1 Resources

There are several roles involved in developing a Kentico CMS project.

The Content Author, Content Editor and the Content Publisher will be responsible for content authoring and publishing. Naturally, depending on the organization, one person may fulfill duties of multiple roles, and perform all content related tasks.

The .NET Developer role is responsible for website development and maintenance. Please note that using Portal Engine development model, no .NET or C# experience is required to perform most of the development tasks.

The Designer role on the other hand, takes care of site-wide visual consistency, and works with developers and builds the website.

The Database Manager is in charge of the operational efficiency of the website, maintaining data consistency, constraints, and work with developers on database optimization.

The key roles are shown in the table below.

Role	Minimum Technical Expertise	Description	Required for Content Publishing
Content Editor	Word Processing, Web Browsing	Creates web content, identifies content changes, determines what, where, and when it will be published	Yes
.NET Developer	ASP.NET	Maintains/extends and modifies CMS application code (code base)	No
Designer	CSS, HTML, FLASH, Silverlight	Maintains/extends or modifies visual design elements inside CMS, using built-in tools	No
Database Manager	SQL Administration	Maintains/manages the CMS repository	No

Table 1 Roles involved in the process of developing the website

3.2 Team development

Kentico CMS offers team development support on source code and object level using an external source control application (e.g. MS Team Foundation Server - TFS). Configuring a team development environment for Kentico is the same as with any other ASP .NET application.

If you need to allow multiple developers to work on the same project, you may setup the development environment as pictured below.

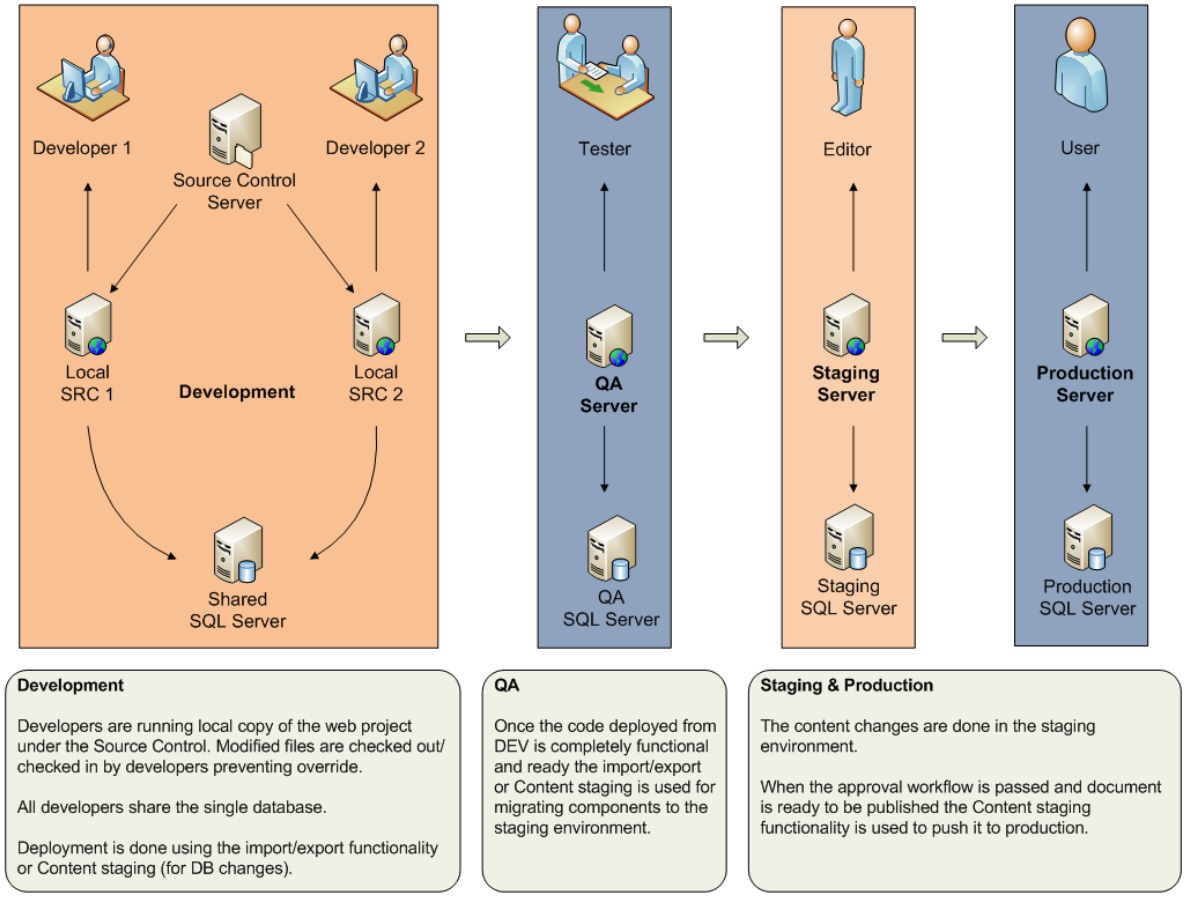

Figure 5 Team development environment setup

As you can see:

- Each developer has their own local copy of the project folder,
- The local instances (running on a local machine with properly configured IIS) point to the same shared database,
- New functionality is managed locally until the change set is ready to be deployed,

  - Modified code files are checked-out / checked-in by developers in the source control application preventing unintended override,
  - Changes made to objects storing metadata in the DB are reflected for all developers,
    - NOTE: You can use object locking mechanism to prevent concurrent changes,
  - Web farm synchronization may be used to notify all the development machines when a change requires the cache to be flushed,
- Once a developer feels confident about a set of changes, the code is submitted to the source control application (to apply them to the entire development environment),
- Changes are then deployed to the QA environment (using export/import)—the QA verifies the functionality,
- New features are transferred to the staging/production environment (using available deployment options),
  - If you want to manage all your content and new functionality in a staging environment, and then push it to production, you may use the Content staging application. It allows you to update content and make changes to the website on a staging server and then synchronize changes with the production environment. That way you do not perform any content oriented operations directly in production,
- For detailed information on deployment options, refer to section 5 Deployment.

Note that in the given scenario, developers use a shared database. This means changes done to the system metadata (and most objects stored in database) are not managed by the source control application.

However, developers can use the check-out/check-in (object / content locking) functionality (available in the CMS UI) to avoid unintentional modifications.

- Object locking is available for layouts, page templates, transformations, web part layouts, web part containers, CSS style sheets, and other objects.
  - **NOTE:** These objects can be also managed by a source control application if you deploy them to the file system. Such virtual controls/files will be stored under *CMSVirtualFiles* folder. You can find source control management can be found in *Administration* → *System* → *Virtual Objects* section,
- You can use check-out/check-in functionality content/pages as well.

Some objects cannot be locked before editing (for example page type form definition, web part, etc.). Thus, it is a good idea to enable object versioning to track changes for these objects. You can compare individual versions in side-by-side mode and roll back to some of the previous versions if needed.

Collision-free approach

Of course, to prevent the overwriting of changes completely, every developer may use a separate database. Although this approach ensures collision free development, changes made locally won't be visible by other team members till they are deployed to a shared database. You may find this approach useful, especially in scenarios where the new functionality being developed is independent on other parts of the system.

3.3 Index template processing

You kick-off the development by processing the design template coming from a graphic designer. Let's consider a design template to be a set of files (index HTML page, CSS style sheet, design assets, etc.) representing static version of the website home page (or any other page).

- Identify common content shared by all pages on the website (used for the master page) and parts of the page specific to certain sections,
- The goal is to specify the content and layout of page templates used for various website sections.

The actions involved in processing and transferring the index page to the CMS are:

1. **Populate master page code**,
   - Take the entire HTML code of the static index page and place it into the master page,
2. **Apply design assets**,
   - Create a new CSS style sheet, copy the available styles into it and assign it to your website. Then upload all design files into the solution and update the HTML code/CSS style sheet references accordingly. Take a look at 2.5 Organizing media files for recommendations on where to store design files,
3. **Identify content used in the header and footer,**
   - Top logo, main navigation, search box, link directory, site map, etc. All of these will probably be shared by all pages on the website,

4. **Move the HTML code of the header and footer to the master page,**
    - Copy and paste HTML code common for the whole website and place it into the master page. Then add a new web part zone to the location previously occupied by content. Insert the Page placeholder web part into the zone to allow the content of sub-pages to be loaded inside the master page,

5. **Create the first page,**
    - Now you need to take care of page specific content (HTML code) used on the index page. First, create a new page template. Grab the HTML code of the remainder of the index page (code that will be specific for this new page only) and copy it into the page template layout of the recently created page (let's call it home page),
6. **Review the index page in the CMS**,
    - If you access the home page now, you can see your index page displayed by the CMS (albeit still just static code). You should be ready to approach the next stage in the development process—replacing static code with web parts.

3.4 Altering static HTML using web parts

So far, you have defined the basic content for the master page and home page. However, the content of pages is still static. To make your website truly benefit from the CMS system, you need to alter the static code using web zones and web parts.

Do the same for all pages no matter whether it is a master page or a regular page:

- Open the layout code of the page,
- Search through the static code for enclosed HTML entities—lists, paragraphs, images, anchors, headings, selection lists, etc. For every entity, you need to decide whether you want to make it manageable using some web part. Otherwise you just leave it as part of hardcoded layout code,
- Each entity is usually part of a certain page element—main menu, logo, secondary navigation, titles, etc. You should check for out-of-box web parts that provide similar functionality as the element that the entity belongs to. That way you move content management from the level of HTML code to a user friendly web part UI instead,
    - **NOTE:** The more dynamic components (web parts, zones) you have on a page template the more overhead processing is introduced to the page build process. Therefore, it is crucial to find the right balance between performance and usability of the system. In other words, do NOT try to make

everything dynamic/editable if you think it does not require frequent updates in future.

- At this time do not replace or manipulate with content (text, audio, video, etc.) data rendered dynamically (e.g. based on the incoming query string or some other condition)—you will most probably create a custom page type and use a viewer web part to output such content.

The process of transferring static content into the CMS is a routine task. It just requires you to know what you can or can't do with pre-defined feature set. You can refer to the table below when transferring basic HTML entities/ elements into Kentico CMS. The table lists some of the most common elements and their CMS counterpart.

Static element	CMS equivalent
Links, headings	Text category web parts: editable text or static HTML, static text, Link/Button web part
Single image	Text category web parts: editable image or editable text, static HTML
Image gallery	Listing and viewers or Media library, Attachments category web parts (depends on the chosen file storage type, see 2.5 Organizing media files)
Editable content	Text category web parts: editable text
Structured content	Listing and viewers category web parts: repeater, datalist, grid
Form	Forms module + On-line Form web part/inline control
Logon form	Membership category web parts
Subscription	Newsletters category web parts

Table 2 Examples of static HTML elements and their equivalents in Kentico CMS

The information above represents just a really small fraction of available web parts. You can check a list of all web parts in the *Administration → Web parts* application or in the documentation. You can also find additional web parts on the Marketplace.

NOTE: We recommend contacting our support team or consulting before you make a final decision to develop a custom web part when you are missing some out-of-box functionality. This may save you unnecessary development time.

3.5 Page template development

Here are several important facts related to page template development that may save you some hard times figuring things out for yourself:

- Every page in Kentico CMS is based on some page template,
- Page templates can be developed using Portal Engine (recommended), ASPX, Mixed Mode or MVC development models,

Portal Engine templates

- A page template consists of two parts: a layout plus web parts and their template specific configuration,
- Page template content (HTML markup, web part zones and web parts) is stored in the database (when using the Portal Engine development model).
    - **NOTE:** When using mixed development model (ASPX + Portal Engine) only web part and zone settings are stored in the database, but not template layout – this is controlled by a physical ASPX page.
    - **NOTE:** Web parts and zones used on a template are stored in the database in XML form along with other template metadata,
- You may want to explicitly specify which website sections allow the usage of a particular page template (to limit the options of editors),
- Page templates can be defined as re-usable or ad-hoc,
    - A re-usable page template is stored under a unique name and is ready for use by other pages,
        - **NOTE:** Changes made to the template on one page influence all pages using the same re-usable page template. You can clone a template as ad-hoc for any particular page to avoid this behavior,
    - An ad-hoc template is used only by a single page unless you save it as a new template,
        - **NOTE:** An ad-hoc template used by a particular page is automatically removed when the page is deleted (unless you save the ad-hoc template as a new template),
- Even when having multiple different page templates, you can still share layouts that are used internally by templates. That applies to ad-hoc templates as well. Any modifications done to a shared layout are reflected by all templates using the same layout. Please be aware of this fact,
- The template used by the root node of the content tree is always defined as a master page template. You can mark any other page template as a sub-section

master page. That way, you will be able to reset the content inheritance level for any nested page,

- If you mark a non-root page template as a master, all its child pages using the 'Inherit only master page' inheritance option will only display content from the nearest parent master page (not from the root node anymore),
- The hierarchy of pages (using different page templates) influences the final look and feel of the displayed page,
- Do not create a new template from scratch every time you need to make a minor change. You can clone an existing template instead and leverage its current settings and functionality,
- If a page is not displayed at all because of some serious error, you can remove web parts causing problems either through *Administration → Pages → <page> → Design* tab or from *Administration → Page templates → <edit> → Web parts* tab in template properties dialog,
- Take advantage of web part zone properties. You can easily hide a particular web zone on sub-pages, display zone content to a certain set of roles or only on specific page types, use macros to evaluate zone properties, disable view state, enable AJAX post backs, etc. This way you influence all web parts in the given zone,

Mobile device templates

Kentico allows you to optimize pages for mobile devices. Please refer below for recommendations:

- First of all, you need to allow Enable device profiles setting in *Administration → Settings → Content → Content management → Mobile development* section. This way you enable system's device detection module. Collected information is then used to optimize presentation for specific devices. It also enables the device preview functionality,
  - **NOTE:** You can change the device view in preview mode from *Pages → Default screen size* selector in Administration,
- Mobile device detection is based on device profiles,
  - Assign mobile devices (smartphones, tablets etc.) to device profiles based on their attributes (screen resolution, flash support etc.),
- Each device profile can cover several devices based on a device selection, custom user agent or custom macro condition (screen resolution etc.),
  - **NOTE:** List of supported mobile devices can be updated manually in the *~/App_Data/CMSModules/DeviceProfile/devices.xml* file or

automatically by setting up premium 51Degrees.mobi account or by uploading a premium data file via the settings,
    - **NOTE:** The system recognizes device profile based on first device matching specified conditions (device selection, user agent or macro condition),
- You can apply different page template layout per device with the automatic layout mapping feature. That way you render different HTML code to clients on different devices,
    - **NOTE:** This feature works ONLY with shared layouts.
    - **NOTE:** Content changes and configuration of web parts placed on the template are stored in the page template, not in the layout.
- The custom device layout feature allows you to define different page template content (different zones and web parts) for each profile,
    - Custom device layouts allow you to have several layout sections (with unique zone and web part configuration) as a part of one page template,
    - **NOTE:** CMSDeviceLayout control is based on more general control called CMSConditionalLayout which gives you even more options,
- An alternative approach is the using responsive design. This can be achieved with standard page template layout, and appropriate CSS styles changing appearance based on the device screen resolution (and other information obtained using media queries),
    - **NOTE:** You can achieve responsive design also with support of device profiles. For each screen resolution you would like to support, define separate device profile with a macro condition selecting the device based on the screen resolution.

CSS Styles

Cascading styles (CSS) allow you to modify the design of your web site by separating the HTML and presentation styles. You can manage CSS style sheets externally using some 3rd party tool. You simply store them inside App_Themes folder as usual.

There is better way to manage CSS style sheets in Kentico though.

- You can manage style sheets in Kentico with built-in editor supporting syntax highlighting. It is available in *Administration → CSS Stylesheets* application,
    - **NOTE:** Use design preview to check your modifications in real-time to increase your productivity,

- To create collision-free editing environment for your CSS style sheets, enable 'check-out/check-in' and versioning functionality,
  - **NOTE:** You can also include CSS style sheets under source control application,
- You can define which CSS style sheet will be used on web site level in site settings (*Administration → Sites*),
  - Kentico will include reference to assigned style sheet on every live site page,
  - You can overwrite site style sheet on any page from *Properties → General* section,
- If necessary, you can reference additional style sheet(s) on the page,
  - You can use *CSS Stylesheet* or *Head HTML code* web parts,
  - You can reference the CSS style sheet from the *Header* tab of the page template configuration dialog or the HEAD section of the master page template,
    - **NOTE:** You can minify and compress your CSS style sheet using build-in support,
  - Most of page components (web part, web part layout, transformation etc.) allow you to specify CSS styles that will be included on page only if particular web part is used on page template.
    - You need to enable *Allow CSS from components* setting which will ensure that page component's CSS styles will be include through dynamically generated style sheet(s),
    - **NOTE:** By enabling *Combine CSS from components* setting, all of the page component's CSS styles will be merged into one (request) which will improve the web site performance,
- Do not forget to validate your CSS styles on page level with built-in CSS validator,
- Keep all of the design assets managed through the *Theme* tab,
- You can combine several CSS style sheets or page component CSS styles into one CSS style sheet by using macro expressions,
- Use built-in browser specific CSS styles feature to address different browsers or mobile devices,

JavaScript files

- You can manage your custom JavaScript files in *Administration → Javascript files* application,
    - **NOTE**: You can also create a dedicated Media Library just for your JavaScript files and manage them via the CMS,
- You can add a reference to your JavaScript files on the page template:
    - With support of *JavaScript* or *Head HTML code* web part,
    - As a part the page template's *Header* section or the master page template HEAD section,
- You can minify and compress your JavaScript code,
    - You can also use Grunt for minification and the combining of JavaScript files
- If you are planning to include dynamic functionality relying on a jQuery library, best practice is to load your own version of jQuery library, but (depending on your Kentico version) you might need to resolve conflicts between your library and the Kentico jQuery library with use of *noConflict* method as described here.

NOTE: Get familiar with macros by watching the following webinar and reading this blog post.

3.6 Custom page types

The concept of page types in Kentico CMS allows you to store and organize structured data in the content tree hierarchy. When we were converting static HTML to web parts, we skipped content that changes under certain circumstances—dynamic content that is regularly added, updated and removed. The goal of this section is to point out important facts related to the development of custom page types.

You can create custom page types using the built-in wizard that guides you through the whole process. If you are interested in page internals, for example how they are stored within the CMS, interconnected with other parts of the system and what their position within Kentico CMS is, you can see this documentation.

The following points should be also considered when using (custom) page types:

General

- Once you define a custom page type, review all available properties (General, Fields, Form, Transformations, Queries, Child types, Sites, E-commerce, Alternative forms, Search fields) to make sure you get the most out of your page types,

- Custom page types can be used by the E-commerce module to represent products on the live site. You need to make sure the *Page type represents a product type* box is checked on the E-commerce tab,
  - **NOTE:** If checked, the *Administration* → *Pages* → *<page>* → *Form* tab also contains product related fields. A new *SKU* tab is added once the product is created in the same area. The option to create a new product or select an existing one is available at the time when you are creating/updating a new or existing page,

Fields

- You can specify the default source field for the page name and page alias (both explained below) on the *Fields* tab,
- In addition to custom fields, you can also add system attributes to page types. This way editors are able to define system field values that only administrators are allowed to manage otherwise. You are able to add custom validation to system fields (e.g. you can make metadata such as tags mandatory for content),
  - Use the *New system attribute* icon, select an attribute name from the required group, and define the details of the field displayed on the edit form,
    - **NOTE:** Values entered for fields from the *Page attributes* group are culture-specific while the *Node attributes* are shared by all cultures,
  - **NOTE:** Avoid exposing system fields that might influence the functionality of the CMS. If you are not sure about the purpose of some system attribute contact our support team for clarification,
- Every field is of some type. The options available in the *Field type* drop-down list depend on the *Attribute type* selection. If you change the attribute type, you will see the available field types change accordingly,
  - You can create your custom field type controls (form controls) and use them for page type fields,
- You can even use macro expressions for some field properties (e.g. a localization macro for a field description, a context macro for the default value of a field, etc.),
  - **NOTE:** You can define custom logic as a part of the *Visible condition* and the *Enable condition* properties. You can use it to hide/show and enable/disable the field according to your custom macro condition,
- **NOTE:** If you add a custom field to an existing page type and do not allow empty values for the field, you will get errors when manipulating existing pages of this type. Therefore, we recommend allowing empty values for any fields added to a page type after pages of this type were already created,

Queries

- You can add your custom queries on *Queries* tab, and use them either in the API or with the web parts accepting custom queries,
- A query may be defined both as query text or as a stored procedure name (then you need to create a SQL procedure in the database) and use a transaction,
    - Use `##WHERE##`, `##TOPN##`, `##ORDERBY##`, `##COLUMNS##` macros in the custom query to allow certain parts of the query to be resolved dynamically before execution. Macros are replaced by the actual values, usually passed in from a property in the configuration of a web part,
    - Automatically generated system queries are not available on the *Queries* tab. If you would like to modify a default query, you would need to explicitly create a new one with the correct name (e.g. select, selectall),
    - **NOTE:** You are not able to change default queries unless the class is marked as customizable. The default queries are used by the CMS to perform all tasks related to the given page type. If you make an inappropriate change to a default query, you may break some functionality of the CMS.

You should also be aware of the following general facts about page types:

- **Page alias**,
    - Each page has its own unique name within the website. The alias path is generated when the page is created, and does not implicitly change with the page name. The alias is used for generating the URL of the page (by default), thus any special (forbidden) characters are replaced,
    - You can specify the page alias source field through *Administration → Page types → <page type> → Edit → Fields → Page alias source field* (at the bottom of the dialog),
    - **NOTE:** The page alias is shared by all language versions of a page,

- **Custom URL path**,
    - If you want to use a custom URL for a page (different from what is generated based on the page alias) you may use the *Page URL path* field to assign an alternate URL,
    - **NOTE:** The URL path is culture-specific, so you can specify a different URL for different language versions of the same page,

- **Name path**,
    - Consists of the names of all pages on the specific path from the first parent down to the selected page. Unlike the page alias, the name path changes

along with the page name. You can force the system to use the name path as the URL path by default,
    - **NOTE:** The name path is culture-specific,
- You can configure IIS and Kentico to support custom extensions or extension-less URLs and that way define the available extensions for a specific page,
- **Multiple page aliases**,
    - On top of a custom URL path, you can define multiple page aliases for each page. That way you make a page available through multiple URLs,
    - Page aliases have lower priority than the custom URL path, thus make sure you remove a URL path that might cause conflicts with the assigned page aliases,
    - **NOTE:** As you may know, having multiple URLs pointing to the page displaying same content is not ideal from an SEO perspective. You therefore need to setup permanent (301) redirection between domain aliases and default URL. Another option is to setup canonical links,
- **Wildcard URLs**,
    - You may consider using wildcards in the URL path or page aliases to use part of the URL to drive content displayed on the page,
    - **NOTE:** Learn more about wildcards in the documentation. There are certain limitations on the use of wildcards if the page has multiple language versions,
- You can exclude certain parts of the website from CMS processing, setup URL prefixes, enable automatic creation of page aliases and manipulate various URL-related settings. Read more here,
- Each instance of a page type allows you to specify its metadata,
    - You can include system fields in page type fields (as described in the previous section) and allow editors to populate metadata without access to the *Properties* tab,
    - You can also use macro expressions for metadata fields.

Custom page types vs. custom tables

Please refer to section C. Page types, custom table to find out whether custom tables or classes might be a better type of storage for your structured data.

3.7 Transformations

As mentioned earlier, page content is rendered using viewer web parts applying templates we call transformations. But not only page types make use of transformations. The custom tables module uses transformations to render content from custom tables as well.

Anyway, no matter whether your structured content is handled by page types or custom tables, transformations are defined the same way in both scenarios.

- You can generate default RSS, Atom and XML transformations when creating/editing a transformation,
- A set of universal predefined transformation can be found under the *RSS transformations* or *SharePoint - Transformations* page type,
- When creating a new transformation, you may use the `<name>_<culture code>` format to specify its name. That way you create a culture-specific transformation. The viewer web part then decides what transformation should be used based on the currently selected website culture,
- You can display a list of functions available in transformations using the link at the bottom of the transformation editing dialog,
    - **NOTE:** The complete list of pre-defined functions is also available in the Kentico CMS API Reference guide.
        - Search for the CMSAbstractTransformation class.
- You can create custom functions or custom macro methods for use in transformations,
    - **NOTE:** Custom functions/methods must be implemented in C#,
- If you want to use a pre-compiled website you need to deploy all virtual objects (including transformations) to the file system before publishing the pre-compiled website,
- Team development may require using the *Checkout* button to avoid overwriting changes in transformations,
- Enable object versioning for transformations to keep track of your changes and to have the option to get back to a previous version if needed,
- Each transformation allows to you define custom CSS styles which are then included on the page if the *Allow CSS from components* setting is enabled,
- When you display multiple page types with a single viewer web part, you may use macro expressions to specify the name of the transformation to be used (e.g. `{%classname%}.default`),
- Search results are also rendered using a customizable transformation,
- Page attachments can be displayed in transformations using attachment gallery controls,
- You can call nested objects in the `Eval()` function,
    - `Eval("SKU.SKUManufacturer.ManufacturerDisplayName")`,

There are several transformation types in the system available (ASCX, XSLT, Text/XML, HTML, jQuery and hierarchical), each serving a different purpose and having specific performance characteristics.

The following description should help you to select the most suitable one in your case:

ASCX Transformation

- Represents a virtual control which requires the Virtual Path Provider (VPP) to load and compile the transformation on the fly,
    - **NOTE:** Each compilation introduces overhead to page processing (consumes RAM + CPU). The application may occasionally restart when the number of re-compilations exceeds specified limits,
- Supports ASCX markup language, inline (ASP) .NET code and nested .NET server controls,
    - Displaying content rating, page attachments, context menus, abuse report control—these are examples of functionality that require nested server controls in transformations,
    - Any other control may be included the same way—even your custom ones,
    - **NOTE:** If you want to call some method (either Kentico API or .NET) in your transformation, you need to use a fully qualified name (including namespace)
        - `<%= CMS.Helpers.ValidationHelper.GetString() %>`,
    - Page fields can be accessed using the `<%# Eval("ColumnName") %>` directive,
- If you need to use macro expressions as the value of some page field, you need to explicitly call the macro resolver in the code of the transformation to evaluate the macro value,
    - The macro resolver is called from the transformation code as `<%# CMS.MacroEngine.MacroContext.CurrentResolver.ResolveMacros(CMS.Helpers.ValidationHelper.GetString(Eval("FieldName"), ""), true, true) %>`,
- To synchronize nested control's (CMSRepeater, CMSDataList etc.) life cycle in the transformation you can use the `DelayedLoading` and `StopProcessing` properties of these nested controls,

XSLT Transformation

- You can't use ASP .NET calls or controls in the XSLT transformations—neither pre-defined functions nor custom ones,
    - XSLT attributes are case sensitive,
- Can be used with any data source serialized to XML,
    - You can use the *XSLT Viewer* web part with XSLT transformations,

- This transformation does NOT require compilation meaning that it is not introducing any additional overhead,

Text/XML & HTML Transformation

- Transformation code is processed as standard HTML/XML,
    - The ApplyTransformation() macro function (to apply a transformation to an object or collection of objects [implementing IEnumerable interface]) works ONLY with Text/XML transformations,
- HTML transformations are editable using the Kentico editor,
- Page fields can be accessed using the `{%ColumnName%}` macros,
- Are updatable even when the VPP is not running in the target environment (in a medium trust OR pre-compiled website),
- These transformations do NOT require compilation meaning that it is not introducing any additional overhead,
- You can NOT use nested controls or inline (ASP) .NET code with these transformations unless wrapped into a custom macro method,

jQuery Transformation

- Represents a template with HTML markup and binding expressions,
- The template is applied to the array of objects and rendered into the HTML DOM,
    - **NOTE:** Template binding happens in the client browser meaning that you can save some application server resources by passing part of the load on the client machine,
- Allows less data to be transferred between server and client - only template text + raw data are sent to the client,
    - Using asynchronous calls to build-in REST interface,
- Updatable even when the VPP is not running in the target environment (medium trust OR pre-compiled website),
- This transformation does NOT require compilation meaning that it is not introducing any additional overhead,
- You need to implement a jQuery call to retrieve source data from (e.g. using REST service to get content from CMS in JSON format),
    - NOTE: jQuery transformations are used internally by Chat and Strands Recommender modules.
- You cannot use nested controls or inline (ASP) .NET code with this transformation,

Hierarchical Transformation

- Used to display pages (and other types of appropriately organized data) in a hierarchical structure based on the hierarchy level (e.g. levels of the page content

tree), type of page and sub-transformation (template for first item, last item, header etc.),
    - **NOTE:** Hierarchical transformation can be used by the *Hierarchical viewer*, *Universal Viewer* and *Universal viewer with custom query* web parts,
- Serves as a container for a number of standard transformations which allows you to display multiple page types by a single viewer web part. The same page type can be displayed in multiple ways based on its position in the content tree,
- Hierarchical transformations are a better alternative (from both performance and usability perspective) compared to nested controls when trying to display hierarchical data structures,
    - **NOTE:** Standard viewer web parts (such as Repeaters, Datalists) do not preserve content hierarchy from the content tree - each page type is loaded as a separate DataTable within a resulting DataSet,
- Executes a single SELECT query for each page type no matter how many parent pages are being displayed whereas when using nested controls (e.g. Repeaters) these execute additional SELECT queries for every parent page,

NOTE: For information about different transformation types take a look at the following webinar.

3.8 Customization options

It does not really matter how hard we try to make Kentico the most flexible CMS solution on the market. You may still need to implement custom functionality or adjust existing features to meet your specific needs. Kentico CMS was designed in such a way so as to provide various possibilities on how to handle any customization challenge.

Review the list of customization options for various scenarios below. The list contains all basic tasks you may come across when developing a CMS project.

Task	Customization options
Hiding certain parts of the UI	Use the UI Personalization module to decide what parts of the UI should be hidden/displayed for a particular group of users.
Administration UI customization	You have the full source code of the Administration UI available in the web project folder. You may copy an existing UI element or create a custom UI elements.

Execute custom code as part of application/ request/ session/ page/ control life cycle	You can make use of system events fired by different application objects. Read more here.
Extend system object	System objects can be extended by custom data and fields using built-in and native support for easy-to-use customizations.
Touch queries before/ after execution	If you need to handle a query before it gets executed or need to manipulate the result set returned, you can easily pre-process or post-process queries.
Modify the layout of out-of-box web parts	You may either modify the ASCX code file related to a web part directly or use a custom web part layout for an upgrade-proof solution (recommended).
Modify web part default behavior	The default behavior of web parts may be altered in the following ways.
Work with a custom class without a physical assembly	Follow the steps for customizing providers.
Perform additional actions when manipulating with system objects	Use global events to react to changes to system objects. It is useful in case you integrate the CMS with an external application like ERP, CRM, etc.
Handle exceptions your own way	Use the Exception event handler to perform additional actions when exceptions occur within the CMS.
Switch from Forms authentication to other authentication methods	You can choose several authentication methods.

Modify authentication and authorization process	Use the Security event handler to integrate external user data storage and modify the default authentication and authorization process.
Execute custom actions when a page is manipulated	You may use page events to perform extra actions, synchronize changes made to pages in external systems, etc. You may handle workflow events the same way.
Need to use a custom database connector, search provider, e-mail engine or e-commerce provider	Develop an alternative provider using Custom providers.
Extend the default editor toolbar	Follow the steps on how to customize the editor toolbar.
Ship the same information with all pages (no matter the page type)	You can add custom data to all pages from one place using the tree node custom data property.
Customize the registration form	You may create an alternative form to be displayed to users using a customized registration form.
Perform custom actions when an On-line Form is submitted	You may run your own custom code when a form is submitted by a user.
Integrate custom/ 3rd party external	You can add your own module into Kentico CMS so that it behaves as a native part of the CMS system.

modules	
Integrate with external systems	You can use built-in support for integration with external systems.
Integrate a custom ASP .NET application	Read more on how to integrate the CMS with an existing ASP .NET application.
Execute custom code in regular intervals	You may use the Scheduler module to schedule custom code.
Include custom functionality / control on a page	Develop a custom web part to be used on your pages (read more on web part development in the next chapter). The other option is to include your control in the page template directly (no web part needs to be created).
Extend an existing page template	To create a custom page template you can modify or clone an existing page template.
Create custom workflows	Simply create a new workflow, add steps and apply the workflow on pages.
Support custom media types	Define a process for handling custom media types.
Searching of custom data	You can develop a custom smart search index as explained here.
Extend the system object metadata table	You may modify the definitions of system objects related to the metadata table using the Modules application.
Changing the behavior of Kentico modules	You can register your custom initialization code for Kentico modules
Changing the behavior of UI components	You can use extenders to add additional functionality to our default controls or pages (you can find an example in the \CMS\App_Code\CMSModules\CMS\Modules\TransformationListControlExtender.cs file).

Table 3 Overview of customization options for certain parts of the Kentico CMS system.

You can see that the most important step is to inherit from the right base class when implementing a custom module or web part. Same principles apply to development of all other custom CMS elements. Take a look at the available base classes you are going to use when creating custom CMS elements.

Element	Base class	Description
Filter	`CMSAbstractBaseFilterControl`	Filters are user controls that are connected to other web parts and allow you to filter out specific data.
Data source	`CMSBaseDataSource`	Data source web parts retrieve data from the database and expose it for use by other web parts. Find more info about filter development here.
Web part	`CMSAbstractWebPart`	Custom web parts allow you to use your user controls (displaying content, providing specific functionality) from within the Administration UI.
Widgets	(no base class – uses web part code file)	Every widget is based on some web part. Widgets allow live site visitors to define a personalized page template layout. Other user roles may use widgets to partially design a page, so an action from a designer isn't necessary.
Form control	`FormEngineUserControl`	Form controls allow you to create a custom input control (e.g. a filtered dropdown control) used in CMS forms running on the Kentico form engine.

Inline control	`InlineUserControl`	Inline controls are controls placed into text in the form of a macro. When the text is rendered, the macro is replaced by a dynamically loaded control.
Administration UI page	`CMSDeskPage`	Base class for custom UI pages integrated into the Administration interface.
Master page	`CMSMasterPage`	To keep a consistent look and feel of your custom UI pages, you should consider creating a custom master page inheriting this base class.
Modal page	`CMSModalPage`	If you plan to enable modal popup dialogs in your custom UI, use this class for dialog pages.

Table 4 CMS elements and related base classes.

If you are concerned about some specific functionality that you feel requires customization, try to search our on-line Knowledge Base. All articles in the knowledge base reflect real world needs of our customers. It is probable that someone has already come across the same functionality you need, we helped deliver it and shared it on our DevNet portal.

There is also a Marketplace section on the DevNet portal where you can find custom modules developed and submitted by other clients using Kentico CMS. It is worth reviewing what is out there, as you can save development time if you avoid implementing something that is already available.

NOTE: If you plan to perform complex operations with pages (and any system objects in general) that consist of multiple tasks, you may want to execute all actions in a single transaction. The CMS API provides you with an easy-to-use transaction mechanism you can take advantage of.

You should get familiar with the Kentico project folder structure and try to avoid modification of system files. If there is no other option to safely customize the system the way you need, you should at least keep a track of all customized system code files that come as a default part of the CMS project. You will benefit from such a list later during the

maintenance stage where you are going to apply a hot fix package or upgrade your website. Needless to say, you will need to redo your customizations whenever we update some of the CMS system files you touched. You will use this list to identify customized files and merge custom changes.

NOTE: If you built a custom UI, make sure it complies with Kentico CMS standards. Use the **UI check-list** that can be found within the Kentico Deliver Now! Methodology package.

3.9 Custom web part development

Web parts are the basic building blocks of your page templates and pages. A web part is nothing more than just an ASCX user control that inherits from a specific base class.

- Web part development is the same process as creating a standard ASCX control,
- Your custom non-Kentico control can be turned into a web part just by changing its inherited base class,
- If your web part works with date and time in any way, consider using proper time zones in its code,
    - NOTE: In code of your custom web part you need to call `CMS.Globalization.TimeZoneMethods.ConvertDateTime` method so that the time zone settings on the web part level are correctly applied as explained here,
- You should ensure that the custom web part produces a valid HTML code. You can use the web part's *Output filter* properties to enable URL resolving, fixing HTML attributes, JavaScript code, closing tags and others,
    - **NOTE:** Output filter's operations introduce overhead processing, so you should consider disabling them and writing code that doesn't need this processing in the first place,
- If you need to achieve a consistent look for all web parts on a page, you may consider using pre-defined web part containers or even develop custom ones,
- In certain situations you may not need to create a web part from scratch. If you only wish to change the default behavior of an existing web part using different initial property values, web part inheritance is the best fit for you,
- You can setup web part properties dynamically in your code,
    - You can also use macro expressions in the web part configuration dialog to evaluate values for the web part properties. Also, you can take advantage of the macro selector when specifying a macro expression (open it using the black arrow button displayed next to the property field),

- Get yourself familiar with the common web part properties. Due to the inheritance from an abstract base class, you automatically include certain functionality controlled by these properties (displaying the web part only for a certain group of users or on specific page types, enabling AJAX or disabling ViewState for the web part, etc.).

When developing a custom web part, you should also be aware of the relationship between the page and the web part's life cycle.

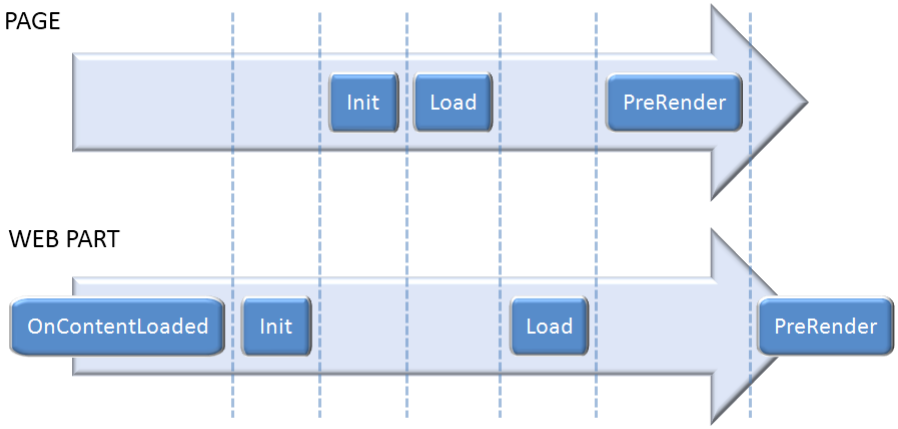

Figure 6 Page and web part life cycle flow.

Events fired during the page/ web part life cycle:

1. `OnContentLoaded` (web part),
    - The event is fired for all controls inheriting from the `CMSAbstractWebPart` base class,
    - The event is fired before the `Init` event,
    - Use this event to initialize the properties of any inner Kentico CMS control used in your web part. Make sure the server control has all its properties initialized prior to its own `Init` event. Read more in the *Initializing Kentico CMS controls in your custom web parts* note (at the bottom) of the web part development section,
2. `Init` (web part),
3. `Init` (page),
    - At this stage all the web parts are loaded and initialized,
4. `Load` (page),
    - All the controls on the page are loaded; `ViewState` and `ControlState` are restored on the `PostBack`,

5. `Load` (web part),
    - As defined by .NET architecture, the `Load` event is fired for all the web parts on the page right after the `Load` of the parent page itself,
    - Register for control events in the `OnLoad` handler, e.g. `Click`, `SelectedIndexChanges`, etc.,
6. `PreRender` (page),
    - This event allows you to make final changes to the content of the page or its controls before the rendering phase,
    - Called first for the parent page and then recursively for all child controls,
7. `PreRender` (web part)

The outlined life cycle matches the default page and ASCX control life cycle defined by ASP .NET architecture. If you are not familiar with the page life cycle read more in the following MSDN article.

NOTE: You can use component events for cross web part event handling. This approach is used also for checkout web parts which can be examined if you need some examples. These events can also be triggered by the *Link / Button* web part.

NOTE: To make sure a created web part complies with Kentico CMS standards, verify that it is using the **web part check-list** which can be found within the Kentico Deliver Now! Methodology package.

3.10 Security

Before we start discussing various facts and concerns about security and membership, you may check out the minimum security requirements for Kentico CMS first. It should give you an idea of what permissions are important for the CMS application to behave correctly. You can overcome some difficult times following the advice given here.

Kentico CMS security and membership general concerns:

General

- The Kentico security approach is very similar to Windows ACLs,
- The security model consists of users, roles, memberships, module permissions, page type permissions, page-level permissions, module permissions and extensive UI personalization features,
- All security related information is stored in the database in the respective tables,
- Roles can be either website specific or global, users are shared between websites,
- Roles are fully-customizable, you may create as many roles as you need,
- Users are members of roles (single user assigned to multiple roles),
- User can be part of a membership to gain permissions of assigned roles for a specific time,
    - NOTE: You cannot assign permissions to memberships. Memberships are only used to group roles and users together
- Page-level permissions can be defined for both users and roles,
- Module permissions are assigned to roles (not to users directly),
- Some modules (e.g.: Media libraries or Forums) provide additional security settings (available through the module UI) on top of the global module permissions,
- If a user belongs to multiple roles, the overall permission set is calculated as the sum of all role-specific permissions,
- Permissions for pages are calculated as the sum of permissions assigned to all roles the user belongs to and user-specific permissions for given pages,
- The DENY permission always takes precedence,

Authentication

- You can switch from the default Forms authentication to a different authentication method according your project needs,
    - Integrate CMS with your Active Directory (AD),
        - **NOTE:** Enabling Windows Authentication does not immediately import users from AD. However, a user and his AD groups are

imported when a domain authenticated user accesses the website for the first time,
    - **NOTE:** You can disable AD groups to be imported automatically as roles by setting *CMSImportWindowsRoles* web.config key to FALSE,
    - Also, you may use the AD Import Utility included in the Kentico CMS installation to import all users in advance,
  - Configure mixed mode authentication to allow users to log in using both Forms and AD credentials,
  - If you run multiple applications under a single domain, you may consider enabling single sign-on authentication,
  - To allow users to register and log in using their Facebook, Windows Live ID, LinkedIn, OpenID (Google, Yahoo! or other credentials), you can take advantage of the 3rd party authentication service support,
  - You can also use **claims-based authentication** to integrate with 3rd party Identity provider as described here.
    - NOTE: Kentico only support **WS-Federation passive endpoint,**
- Use the Security event handler to integrate external user data storage and the modify the default authentication and authorization process,
- You can continuously track on-line users by enabling the On-line users module,

Registration

- You can easily alter the default registration process by modifying the available registration web parts,
  - Use out-of-box registration approval and double opt-in capabilities,
  - Enable password policy with at least 6-8 characters (at least one non-alphanumeric character is recommended),
  - **NOTE:** All users registered on the website can login to all websites running in the same CMS instance by default. Consider locking users only for the sites they registered for by updating shared accounts configuration,
- If you are implementing a community portal, you are most probably going to display user profiles to visitors at some point. To ensure users' confidence, you may allow them to control the visibility of particular profile fields,

Securing the live site and UI

You have several other options on how to increase website security besides page and module permissions,

- You can force the system to require user authentication when accessing certain website areas by creating secured website areas,

  - **NOTE:** By default, only pages within a secured area check user permissions. If you want to check page permissions for public pages, you need to set *Check page permissions* setting to *All pages* for the whole website (*Administration → Settings → Security & Membership*) or enable *Check permissions* on component (web part / widget) level if supported by component,
- If you deal with sensitive data, you may consider forcing a secured SSL (HTTPS) transfer between the client and web server by enabling SSL support,
  - If you need to enable secure transfers for the CMS UI, go to *Administration → Settings → Security & Membership* and enable the *Use SSL for administration interface* property,
    - Make sure that SSL is configured on your IIS server and working properly before you switch on the setting. Otherwise, you may receive an error complaining about SSL settings which then prevents you from accessing the CMS UI,
  - **NOTE:** Kentico CMS does not configure SSL for the website, you still need to setup it up manually on the IIS level,
- To secure your Kentico UI in situations when you might not be currently in front of your computer while still logged in, configure the *Screen lock* settings,
- To deny access to visitors coming from certain IP addresses, you can additionally setup the Banned IPs module,
- You can filter unwanted input from web site users in Forums, Blogs, Messaging and Message Boards modules with Bad words module,
- To prevent robots from submitting multiple actions on your website, take advantage of the flood protection feature,
  - You enable flood protection at *Administration → Settings → Security & Membership → Protection* using the *Enable flood protection* option,
    - You can specify the minimum time interval required between user actions,
- You can authenticate users using the API whenever necessary,
- Set *Maximum number of invalid logon attempts* setting to match your requirements to prevent security attacks on user accounts,
  - By setting a password expiration period, you will force users to change their passwords in a specified amount of time,
  - **NOTE**: If you happen to forget your administrator account password or if it gets locked for some reason, you can either access `CMS_User` database table and clear the password / unlock account or if you do not have access to the database, only to file system on the web server, you can use

CMSAdminEmergencyReset web.config key to reset / create new admin account,

- If you implement a custom web part, module or other object used in the CMS, you should check user permissions using the Kentico API,
  - As mentioned earlier, you can create permissions for a custom module to get control over its usage,
- If you need to achieve extra secure configuration that would even check GET and POST requests for any non-standard parameters, you may explicitly specify accepted parameters.

Security threat handling in Kentico CMS

Today's overcrowded internet environment is fertile soil for various groups planning and launching attacks on public facing websites. These groups usually take advantage of various known exploits. To eliminate possible back doors and allow your website to fend off attacks, review the following list of the most common security threats. Along with definitions of the issues, you get our recommendations on how to protect against each of them. In addition, you will see what means the Kentico CMS API provides to allow you to achieve high security of your custom code.

A. Password theft

Issue	If you store and display passwords in the UI in readable form or have password which is easy to guess/try, it is a security threat. You can never say for sure that nobody is watching your screen or is not be able to figure out your password.

Solution	Protect passwords in the database as well as the presentation layer, setup your password policy.
	
	• Make sure you do not display passwords in readable form anywhere in your custom UI—not even administrators should be able to see the passwords of other users,
	• Consider using SHA-2 with salt hashing to encrypt passwords before they are stored in the database,
	  ○ If you switch the password format from default plain text to SHA-2 with salt, existing passwords will be still working, but you might want to reset all existing passwords,
	  ○ **NOTE:** When using SHA-2 with salt hashing, users are not able to retrieve forgotten passwords; they need to generate new ones.
	• Configure password expiration policy to change the passwords

	once per some time, ○ You can even force users to change their passwords on logon if the password does not meet the policy (regular expression, minimal characters etc.).

B. Permission elevation

Issue	Let's assume there is a user that is only allowed to add new content to the website. Due to a hypothetical security hole, this kind of user could access the section of the website where permissions are managed. If the user can gain higher permissions through this interface, you may have a serious issue. With a new permission set, the user is able to perform unwanted changes to the website configuration or cause some serious information leaks.
Solution	**NOTE:** All security management areas built into the Kentico CMS UI perform an authentication and authorization check by default. • You should always check a user's permissions in custom sections of the UI, especially when you also integrate custom security management areas, • If you define custom permissions through the CMS, you can then use the Kentico API to check whether the user is authorized for specific resources, • Make sure users are not allowed to assign higher permissions than they already have to themselves or to other people, • If you develop your custom UI integrated into the CMS UI, use the correct page base classes (a list of available UI page base classes can be found in **Table 4** CMS elements and related base classes.).

C. Cross-site scripting (XSS)

Issue	A page displaying an input field makes XSS possible if the input is not treated in the proper way. Imagine the following situation. You display an unhandled text box field on the product page. Users enter comments for the products. When a comment is submitted, it is listed on the page. A user enters JavaScript code as a comment and submits the form. The code gets rendered and executed. The attack was successful. The worst part is that an attacker may cause you many sleepless nights by using very simple JavaScript code.
Solution	Always expect that users will enter harmful code. You must handle content entered via input fields and escape potentially harmful code. • Encode content before it is rendered on pages, • Encode all strings coming from an external source (user input, database, context/environment, URL query string parameters, method parameters, etc.), • For strings from external sources, use the `HTMLHelper.HTMLEncode` method, • For URL parameters, use `QueryHelper.GetText()`, • Values coming from an external source rendered as part of JavaScript code must be encoded with `ScriptHelper.GetString()`.

D. Cross-site request forgery (XSRF)

Issue	XSRF is the ability to perform certain actions on the website just by redirecting a user to a page using specific parameters in the target URL. XSRF takes advantage of authentication information stored in cookies. When such a URL is submitted by an authenticated user, a certain action is executed without the user's approval.

Solution	• Do not perform any actions on GET requests, always use POST, • Never turn `ViewState` validation off on a page, • Encode the `ViewState` using machine key, • Do not set the `CMSUseViewStateUserKey` key to false. It enables `ViewState` encryption using a user specific validation key (`SessionID`), • When you create a custom UI page, always inherit from the base page classes listed above (**Table 4** CMS elements and related base classes.) • If you create a new page base class for custom UI pages, make sure it inherits from the `CMSAbstractPage` (directly or indirectly), • You may also consider securing URLs using a hash code.

E. SQL Injection

Issue	Similar to XSS attacks, SQL injection also takes advantage of unhandled input fields. When you use the value of some input field in the text of a SQL query, you give users the possibility to change the text of the original query and execute possibly harmful code. It allows users to execute any database command and gain access to the system, change or delete the data and so on.
Solution	• Use SQL parameters for dynamic parts of the `SELECT`, `INSERT`, `UPDATE` or `DELETE` queries, • Avoid using the `exec()` function in your SQL code, When you build SELECT queries in the code, replace single quotes with double quotes—use `SqlHelper.EscapeQuotes` method for every input coming from an external source, if used in LIKE search, use `SqlHelper.EscapeLikeText(SqlHelper.EscapeQuotes))` • Do not rely on JavaScript validation. JavaScript is executed on the client side. An attacker can disable such validation easily, • Make sure the input used in SQL queries represents a valid value of the expected data type. For example, use the `ValidationHelper.GetInteger` method to validate incoming input if a query expects an integer value.

F. Clickjacking

Issue	A typical attack scenario is that an attacker adds a target website into a frame to his/her website. The attacker covers the frame with different content (a video with funny moments for example) and sends a link to this website to the victim. The victim clicks on the area with the frame (to see the video), but instead he/she is clicking on the target site and performs an action without realizing that.
Solution	• To prevent this attack, the best practice is to send the `X-Frame-Options SAMEORIGIN` HTTP header to the client. This header says that a page could be rendered in a frame only in the same domain as the parent page and it is included in all HTTP responses in Kentico by default, ○ You can exclude certain paths by specifying them in the application setting key *CMSXFrameOptionsExcluded* in the web.config (when you specify "/" you disable this protection completely).

The list given above obviously does not include all vulnerabilities known today. We therefore recommend that you check community groups for the latest information on revealed exploits and how to leverage your code security accordingly.

NOTE: For more information how to properly handle security in Kentico, please refer to the Security white paper.

3.11 Content migration

In case you are migrating existing web site or some data into your Kentico web site there are couple of options that you can use for this purpose.

- You can use external utility called Import Toolkit which is capable of importing content (pages and their attachments) as well as other objects (like users, custom table items etc.) into Kentico from external sources like MS SQL Database, XML, CSV or XLSX files,
- You can use also Kentico API to create new pages or generally any objects,
  - **NOTE:** Since Kentico is database driven CMS, you can directly populate appropriate database tables with your data, but you have to extremely careful not to introduce some inconsistence. Therefore, whenever possible, use Kentico API instead,

- To some point, you can populate your web site with some content by using the REST service,

3.12 Website optimization

The last step in the development process should be performance optimization. To ensure performance is not hurt by inefficient code or overlooked settings, you should devote some additional time to reviewing the available performance indicators and verifying the website configuration.

A. Built-in debugging tools

A kind of best practice would definitely be to use the CMS debug capabilities to review how the website stands and performs from an efficiency and optimization point of view. Before you get your hands on professional optimization heavy artillery, give the out-of-box debugging functionality of the CMS a shot.

You can use debug to review loaded System objects and Cache items, Cache access, SQL query efficiency, ViewState usage, IO operations, Output validity, Security evaluation, Request processing, Macro evaluation, Worker thread performance or Web farm task synchronization. Information displayed on the debug tab is very useful, not only for identifying the cause of potential issues, but extremely handy when you are optimizing a website.

- You can enable all debug tabs at once using bulk keys or settings,
- If you want to get notified when a system error occurs, setup website error notifications,

System objects

- All system objects listed on the System objects tab are loaded into hash tables for higher performance,
- Whenever you make a change to a system object in a web farm environment, the Web farm support should take care of reloading objects in system hash tables for you. Otherwise you may encounter problems with outdated data displayed to other developers in a team development environment,

Worker threads

- Check the duration of active threads,
- Any operation running suspiciously long may require your attention,

- Use context information to locate the source of the call causing an overloaded thread,

SQL queries

- Check the queries log for query duplicity—if the same query is executed multiple times from the same context, it is most probably an optimization issue,
- Keep an eye on data returned by queries
    - Check the number of columns returned by a query—make sure the query returns only necessary columns,
        - Most of the web parts feature the *Columns* property that you can use to specify what columns should be returned,
        - **NOTE:** If you are using the Kentico API to execute queries, always use the override accepting the columns parameter,
- Also check the duration of query execution, if you find a query running longer than others, verify its efficiency,
- Make sure you use the proper indexes for your data tables. We are not able to create default indexes to suit every project and every scenario (double check for custom tables),
- If you are using the Online Marketing (Kentico EMS) separate the EMS database from the main CMS database,
    - **NOTE:** Running the EMS database on dedicated SQL server will give you some performance boost,
- If you are still not satisfied with the performance of your web site, run SQL Management Studio Profiler or Data Tuning Advisor to investigate the efficiency of your query execution plan, index usage statistics, etc.
    - Watch out for the index seek versus index scan ratio,
    - Eliminate as many ad-hoc execution plans as possible,
    - If you are running SQL 2008 or higher, use forced parameterization for queries (parameterization is enforced only for queries without ORDER BY 1 statements),
    - Get yourself familiar with AWE and PAE settings to decide whether they can help your scenario,
    - Consider modifying MaxDOP used by the SQL server,
        - **NOTE:** It is recommended to set MaxDOP to match the number of physical processors available on the SQL server machine,
- You can also setup a multi-server database environment to handle the enormous load your website may possibly get,
    - More information about supported architectures is given in section

- 5.1 Production environment configuration,

ViewState

- ViewState serializes and renders additional content on pages causing the page size to grow accordingly,
  - The advice here is pretty simple: disable ViewState for all controls that do not need it to work properly (typically labels, textboxes, literals, drop-down lists, etc.),
    - **NOTE:** You can also use *Disable ViewState* property on the web part level,
  - Especially try to eliminate ViewState items with the 'Is dirty' flag set to 'Yes' (in red),

Output

- Check the size of the page output (HTML code). If it gets too big (above 80kB), review the code for any parts that could be optimized in some way,

Security

- You should check the security debug tab for any duplicity in security operations,

Request

- On this tab, you may easily check what specific part of the requests caused a delay in response time,
- Review any part of the request processing that took significantly longer than other parts.

NOTE: All debug records are stored within application memory or on the file system. This means that enabling debug facilities puts more pressure on CPU and memory resources on the web server. The higher the load you get with debugging enabled, the greater the overhead your server will face.

- Turn off debugging as soon as you finish optimizing/ troubleshooting your website.

B. Caching options

Caching is the most powerful weapon when it comes to website performance. You should definitely understand how caching works in Kentico CMS in order to be aware

of all the different caching possibilities (webinar) offered by the CMS (read even more in the following blog post).

- If you are developing a custom control that manipulates data in any way, you should consider implementing caching for it,
    - Kentico CMS provides API to help you maintain cache for custom controls. However, in some situations you might not be able use caching in custom code because you always need to work with latest data. In order to prevent system congestion in case of lot of requests/visitors, take advantage of progressive caching while still using Kentico API for caching with `cacheMinutes` set to 0,
    - You can get more examples of API usage in this webinar,
- When you cache content that depends on some other system object and want such content to be updated on the client side correspondingly, take advantage of cache dependencies,
- Good rule of thumb would be that caching for short period of time is better than no caching at all,
- Disable all caching options during the development phase/stage and make sure that the system performs just fine even without cache enabled – if not, optimize your performance until you reach the satisfactory results,
- If you notice issues with the application using too much memory, it could be related to the way caching is setup for web parts and controls,
- If you encounter any performance degradation in the production environment under a heavy load, make sure caching is not overused,
    - If you do not have enough memory to cache all content, you may end up with a situation where items are removed and added from/to the cache too often. In such a case, you should consider turning off caching for some parts of the system or better yet expanding the available memory level,

C. Output filter and other settings influencing performance

Output filters perform additional changes to the HTML output before it is sent to the client. There are several types of output filters you may use. As you can guess, such HTML output manipulation uses resources and increases the overhead of request processing.

An output filter (depending on its purpose) basically searches the output code and tries to fix any problems. In an ideal scenario, we should not need any output filters as long as our code is squeaky-clean. Unfortunately, that is not always the case and that is why you

should understand each of available output filters. Then decide whether you can afford to turn off a certain filter or not.

- You can extend the collection of pre-defined output filters and create a custom one,
- You can enable output filter for whole web site or particular page element only (web part),
- Another group of functionality that influences website performance to a certain extent is related to images and files. You may boost website performance by speeding up the retrieval of your files and images or offloading the file requests to a CDN provider.

D. Scheduled tasks

There are more than 35 out-of-box scheduled tasks available in Kentico. The built-in scheduler executes scheduled tasks on specified intervals. Most of the tasks are enabled by default even though they may relate to modules or features you do not use on your website (e.g. task that clean up e-mail queue does not need to run in your website if you are not using e-mail queue and so on).

While some of the tasks require information about current website context most of them do not. That makes opportunity to move task executing out of the website scope and run it as Windows Service instead. That way you can off load some of overhead from website AppPool and use an external Windows service to take care of it. Not only you save resources, but you also avoid situations where task execution was terminated because of AppPool restart or crash.

- Go through the tasks in your website and adjust the execution interval to better reflect your project needs,
    - You typically want to execute tasks during the time the website has statistically lowest traffic or the web server is facing lowest load.
- When developing a custom scheduled task, recognize whether it requires web site context information to execute properly. If so, implement the custom task in App_Code. If not, create a separate assembly (DLL file) and run it through external Windows Service.
    - It is recommended to implement custom scheduled tasks in App_Code. Then you do not need to rebuild your custom scheduled task assembly each time a referenced Kentico DLL gets updated during an upgrade.

F. Event log

The event log (*Event log application*) may cause performance issues under certain circumstances too.

Event log data is stored in the database. The length of the log history is defined by the *Settings → System → Log size* setting. You can partially control what events are recorded in the log using the *Administration → Settings → System → Log metadata changes* option.

The website slowdown may be caused by the event log being under a heavy load when an error on the website is occurring very often (e.g. references to a missing image). In such a case, new records of the exception are inserted repeatedly. Once the log meets its max length, system starts removing old records. There are many I/O operations invoked during this time. The database server may not have enough resources to handle load, resulting in severe stability and performance issues.

- Always set a reasonable history length for the event log (longer rather than shorter),
- Try to resolve all errors you get in the event log,
- Setup event log e-mail error notifications to the administrator,

If you followed all recommendations and are still facing performance issues, try troubleshooting performance issues or learn from the most common performance mistakes.

NOTE: Kentico Consulting Group offers the Performance and Health Audit package where an experienced Kentico Solution Architect will help you achieve high scalability and performance stability of your solution.

G. SEO optimization

There is no doubt SEO is an important part of the development and optimization process. The same SEO rules apply to your Kentico CMS websites as for any other websites. Unlike ad-hoc websites, Kentico CMS provides easy-to-manage SEO support that you can benefit from.

There are a few basic rules related to SEO that you should definitely comply with. If you are interested in SEO topics, you can download Google SEO Starter Guide to help you understand what it is all about. You will eventually see that SEO is not as complex as one might think.

- You can use external tools to check how your website follows SEO standards,
    - You can even use SEO analysis applications available for free. Web CEO is a free suite combining several SEO tools that help you uncover weak spots in website search engine readiness,
- If you want to hear an experienced partner of Kentico talking about SEO and how Kentico handles real world scenarios, take a look at the following webinar.

NOTE: Check the webmaster guidelines for Google, Yahoo!, Bing and other search engines.

Besides the general SEO recommendations which you should follow, Kentico CMS will help you to manage ~~s~~ SEO in several parts of the system as explained below.

Avoiding duplicate content

- Use 301 (permanent) instead of 302 redirects,
    - You can set 301 redirects by enabling the *Settings → URLs and SEO → Allow permanent (301) redirection* option,
- If your page is accessible by several URLs (Page Alias, Custom Page URL Path or standard Node Alias Path), make sure that all of the URLs are redirected to the main URL with a 301 redirect,
    - You can achieve such redirections by enabling both *Redirect page aliases to main URL* and *Redirect pages to main extension* settings,
        - In special cases you do not want to redirect Page Aliases to the main URL which can be configured on the Page Alias level by setting its *Alias redirection* property to the *Do not redirect* option,
    - If the default (Home) page is accessible by several URLs (e.g. */Home*, */Home.aspx*, */Default.aspx*, */*, etc.), you may want to set a particular URL as primary one in the *Default page* setting,
- You should be concerned about canonical link elements on your website. Even though the CMS does not support this feature out-of-box, you can easily achieve this functionality by using built-in functionality,
- Use extension-less URLs as main URLs,
- Decide whether your URLs should be all in lower case, upper case or case insensitive form,
    - Keep all of the URLs in consistent form by utilizing the *Redirect invalid case URLs to their correct versions* setting,
- Keep the domain prefix (e.g. *www.domain.com* or *domain.com*) consistent in all cases,

  - You can configure URL redirection on IIS level outside of Kentico CMS (better performance) or directly in Kentico by utilizing the *Process domain prefix* setting (slower; easier to setup),
- There are several other URL related settings in the *Administration → Settings → URLs and SEO* section that influence the page ranking,

Content indexing by crawlers

- Properly configure your web site sitemap by excluding pages that should not be indexed by internet crawlers,
  - Each page can be excluded separately via its properties in *Properties → Navigation* section,
- Another alternative to exclude some content from being indexed is to create a robots.txt file for your website. You can handle robots.txt files when running multiple websites under a single Kentico CMS instance,
  - The *Robots.txt path* setting in *Administration → Settings → URLs and SEO* section allows you to select one of your pages as a landing page for your Robots.txt file. Such page should return `text/plain` mime-type response. You can use properly configured *Custom Response* web part for that,
- You can combine both approaches – sitemap and robots.txt files,

Content optimization

- You should include keywords in links wherever possible,
  - Page URLs are, by default, generated based on the page alias (as mentioned in section
  - 3.6 Custom page types,
  - If you want to include keywords in a URL, you can use the *Page URL path* or the *Page aliases* on the page *Properties → URLs* tab to specify custom URLs containing your keywords,
- If you use Flash content on your websites, you should provide alternate content for clients that are not Flash-ready,
- Make sure that you have high-quality and unique content,
- You can specify page metadata (title, description – important, keywords – less important) through the *Administration → Pages → <some page> → Properties → Metadata* tab,
- Do not forget about the ALT attribute of image tags,
  - If you insert IMG tags into the content using CMS selection dialogs, you can specify the ALT attribute directly in the dialog.

4 Testing

The testing phase is crucial for ensuring the overall quality of deliverables. Besides validating specific functionality, you should go through some general quality assurance process of functional testing, site validation and load testing to make sure you deliver a stable, safe and defect-free solution to your client.

4.1 Functional testing

At this stage you should make sure any page type fields, On-line Form fields, custom form controls, custom controls, web parts and basically all input fields across the website apply validation rules. Also verify the functionality of custom developed features and modules.

- For any input field defined through the CMS UI (through the field editor) that requires validation, you should specify validation parameters in the *Validation* section of the field detail form,
  - Do not forget to set the max length for input fields to avoid exceptions from the SQL server due to data length exceeding column specifications,
  - For special types of fields (e.g. e-mail, phone, ZIP, etc.), it is not always necessary to define a custom regular expression. You can use some of the pre-defined field types designed to provide the desired validation functionality out-of-the-box,
  - To implement validation in your custom controls or web parts, you may take advantage of the `CMS.Helpers.Validator` class which features pre-defined methods providing different kinds of validation (e.g. code name format validation, e-mail format, various data type validation and so on),
- Sooner or later you are going to develop/ customize a certain part of the CMS system. Any enhancement done to the CMS should be bundled with an appropriate monitoring mechanism. You should use the `CMS.EventLog.EventLogProvider` class to log any events system administrators should be aware of,
  - Logging exceptions through the `EventLogProvider` ensures that notification e-mails are sent when an error occurs. You therefore need to specify the *Administration → Settings → System → Error notification e-mail address* setting,
- Make sure any custom UI sections comply with Kentico CMS standards. Use the **UI check-list** which is part of the Kentico Deliver Now! Methodology package,
- Review *Administration → Event log* for any exceptions that occurred during functional testing performed recently. You should not see any of those before moving live,
- Perform functional testing to verify use case scenarios.

4.2 Site validation

Time devoted to site validation can eventually be turned into a nice website performance boost and tightened security.

- Make sure the (X)HTML output produced by the website complies with W3C standards allowing you to turn off various output filter options as mentioned in Output filter and other settings influencing performance,
    - You can use built-in HTML and CSS validators on page level,
    - If you want to output valid HTML 5 on your web site, make sure that the HTML 5 output filter is enabled in *Administration → Settings → System → Output Filter* section as well as the *Render HTML5 media tags* is enabled in *Administration → Settings → Content → Media* section,
- Make sure that the website is attack-proof against various types of security threats as mentioned in section 3.10Security
- Check whether the website output matches all the requirements of various web standards as mentioned in section D Which web standards should be followed in terms of accessibility, and coding?,
    - You can use built-in Accessibility validator on page level,
- Check for broken links on page level with built-in Link checker,
- Make sure permissions are applied and validated correctly if you configure advanced security for the CMS users (e.g. for limited access to the modules UI, secured website areas, etc.),
    - Should you encounter any difficulties identifying the causes of permission related issues, take advantage of the built-in Security debug capabilities.

4.3 Load testing

In chapter 3.12 Website optimization, we talked about optimizing the website by maintaining the database indexes and statistics. Part of the testing stage should therefore also be validating the results of such optimization with your own load test preceding the deployment to the live server.

With every major version, Kentico releases a performance report (the latest report was published for Kentico CMS 8.0), which provides you with the results of load tests performed for the specific version of Kentico CMS using one of our default sample websites.

Besides advanced load testing applications and services (e.g. loadimpact), you can use Visual Studio Enterprise Architect – Application Center Test (ACT) to perform a simple load test on your website to verify that changes and optimizations you have done serve a purpose, and do not harm performance in any way. This tool is part of MS Visual Studio Enterprise Edition.

- ACT allows you to fire simultaneous browser connections on your website to execute extreme scenario verification and response time verification,
- Before you start load testing, you should make sure that you setup the website caching the way you plan to use it on the live site,
- Prior to load test execution,
    - Make sure the built-in debugging tools are turned off to get accurate results unaffected by additional write operations caused by the debugging mechanism,
    - Make sure the event log doesn't contain any error messages.

5 Deployment

5.1 Production environment configuration

Before you go live, it is important to consider the production environment setup. You can take advantage of different network architectures to support websites ranging from small to enterprise level. A properly chosen architecture allows a website to perform well and ensures it serves its purpose.

A. Web farm support

A web farm is a collection of computer servers intended to accommodate server needs far beyond the capability of a single machine. It is configured on the IIS level and basically allows an increased ability to serve incoming requests.

- If you expect heavy traffic to hit your website, configure a web farm environment on the IIS level and enable web farm support in Kentico CMS,
- The number of web farm servers (WS) supported by a single Kentico instance depends on the type of purchased license—contact your Account Manager for details about your license if you are not sure,
- Web farm support ensures the synchronization of all changes in the site settings, cache or file system (for changes performed via the UI) between all servers in the farm.

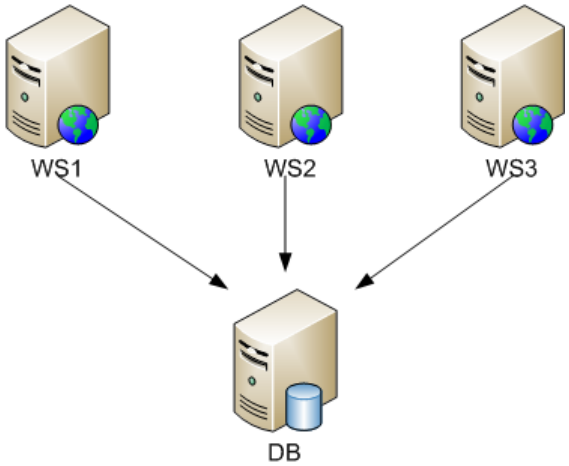

Figure 7 Basic web farm setup

B. Web farm support and a clustered database

Another option is to run the website in a web farm environment and enhance the database level configuration. The database does not need to be just a single physical machine. Utilizing multiple machines and running the SQL Server instances in a cluster could multiply database performance. The higher the load the database can handle, the faster the web site reacts to the users' requests.

- A database cluster acts as a single SQL Server instance for the website. It means web farm support can eventually be applied the same way as in the previous example.

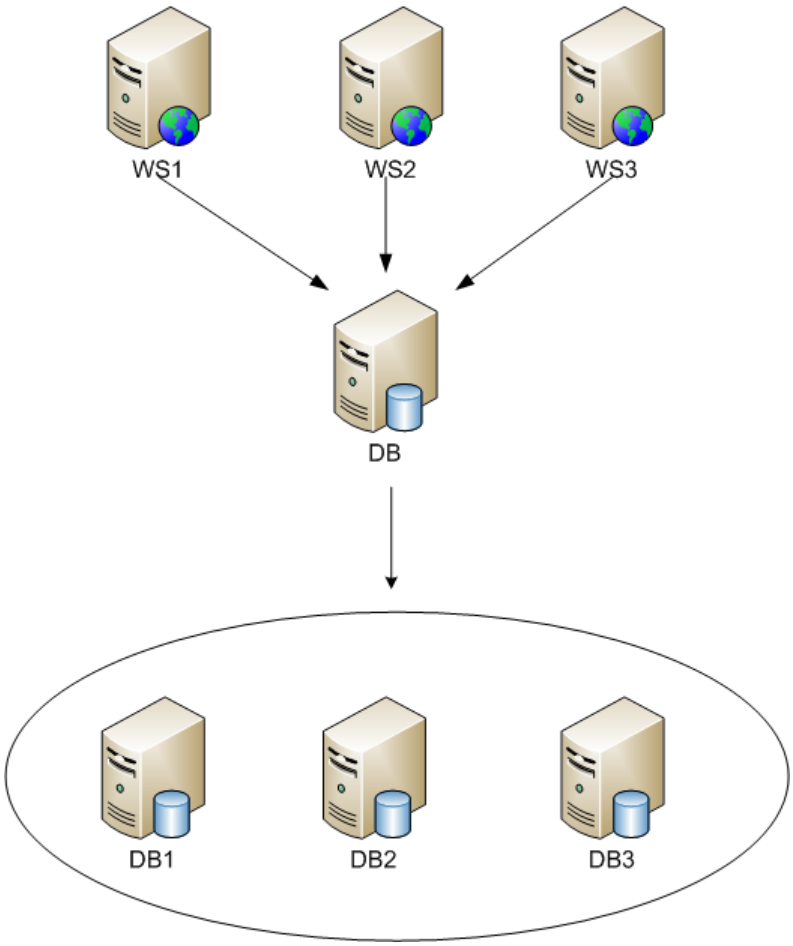

Figure 8 Web farm running on a clustered database

C. Web farm support and SQL replication

The most advanced scenarios may incorporate replication of the data changes on a database level using native support provided by the SQL Server. The website may be a part of a content delivery network (CDN)—a network where web servers and database servers containing copies of the data are located at various points. The purpose is to maximize bandwidth when accessing data. The data are always accessed from the server nearest to the client.

- Each web server makes use of a single database server,
- All database servers establish a replication relationship and propagate changes to each other,
- Web farm support is enabled the standard way.

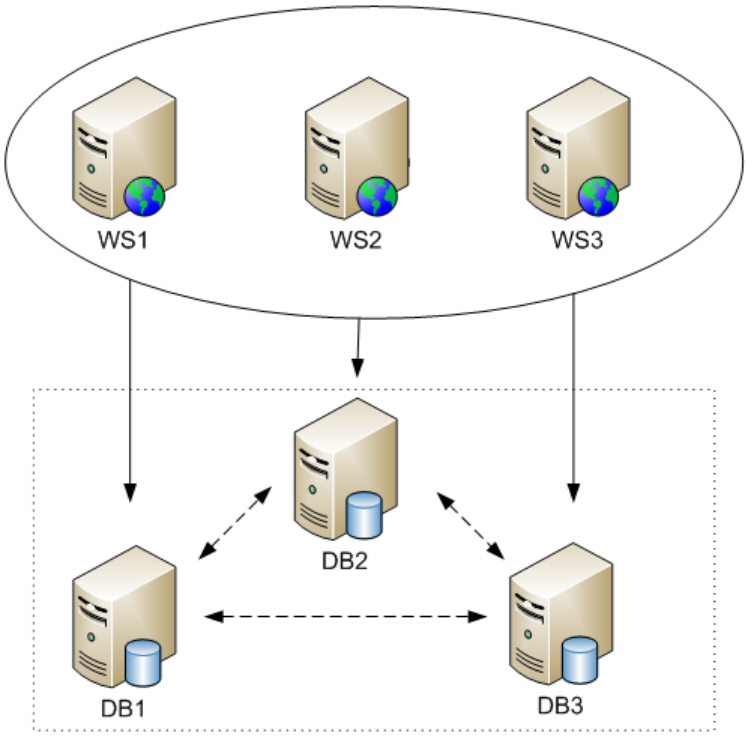

Figure 9 Web farm support and replication

NOTE: If you are interested in more details on Kentico CMS setup for load-balanced, highly available redundant web farms, watch this webinar.

5.2 Deployment options

Before the deployment, you should verify that the production server meets the minimum system requirements. As long as required HW is concerned, we recommend getting at least 2 GB RAM (to be consumed by the website AppPool) and a Pentium Core 2 Duo (or similar) processor. Obviously, the more memory and computing power you supply, the higher the load the website can handle. Although, sometimes it is better to spend additional time optimizing the website rather than spending money on more powerful machines.

A. Deployment in general

Concerning deployment, you have basically two options how to deploy a website to the production server:

- Using the Export/Import functionality of the CMS,
- Restoring the website using a backup project folder backup and DB,

Export/ Import

- Export/ Import allows you to export a website along with all site-specific data and global objects and import it into the production instance of the CMS,
- You need to install a production instance before deployment,
    - You can use the Kentico installer to install Kentico CMS on a remote server,
- If you implemented any custom functionality, you should be familiar with how the project folder structure influences what physical files are included in the export package,
    - If you want to exclude specific files or folders from the export, you may use web.config keys,
- There is no limit on size of the export package—it may contain files of any size,
- If you have an administrator access rights you can upload the export package to target environment directly from Site Manager UI,
    - If you use the publish function of Visual Studio to deploy a pre-compiled website, you will not be able to fully use export/ import - NONE of the physical code files (.cs/.vb, .ascx, .aspx, etc.) are imported, while files that are not compiled (like page attachments, media library files, metafiles, form files, CSS style sheet design assets etc.) are still imported. Also objects stored on the database level are imported,

Restore from backup

- The backup-restore option allows you to restore a production environment identical to what you have in your local environment,
- There is no limit of the maximum size of objects related to the website,
- You simply copy the project folder to the production environment and setup the production IIS to point to this new location. Then you restore the DB backup file and update the web.config file to point to the new database,
    - You need to gain access to the SQL server to be able to restore a DB backup file,
    - Ideally the development and production SQL servers should use the same server settings,
- Because you restore the DB from a backup file, all metadata including website settings are configured for the development environment. Thus, you need to make sure that environment specific settings (in *Administration → Sites* and *Administration → Settings*) reflect the new location (site domain and domain alias settings, SMTP server details, etc.),
- You cannot use this option for incremental deployment as there is no easy way to select individual objects to be deployed,

Incremental deployment

- You can use export/import to perform incremental deployment of new functionality into a website already running in the production environment,
    - Incremental deployment is possible using the export objects or single object export functionality,
    - The other option is to setup Content staging between the development/ staging and production environment,
        - You can synchronize individual objects as well —you can make a selection on the object level,
        - Not all object types are supported by staging functionality (e.g. web part code files, On-line Form data, forum posts, ACLs [page permissions], etc.). A complete list of supported objects can be found on the module's overview page,
        - You can configure all of your environment to be synchronized with bi-directional staging,
        - Content staging uses HTTP POST requests to transmit data between the staging and production server. Due to this fact, the maximum size of any synchronized file is dependent on the maximum size specified for a request.

B. **Deployment to Windows Azure Environment**

When it comes to ensuring high scalability (performance) and availability, running the web site(s) in a Windows Azure environment is definitely an option to consider. All cloud solutions typically provide a mechanism to achieve an HA (High Availability) setup. Windows Azure is a SaaS and IaaS solution off-loading maintenance costs associated with network infrastructure setup and maintenance. At the same time, there are several aspects of cloud solutions you need to consider at first.

- Windows Azure uses the SQL Azure relational database which is almost identical to standard SQL, but there are some SQL server feature limitations,
- You need to re-deploy the whole solution whenever you update/change the codebase in your Kentico project folder (add, edit, remove code files) which results into more work when performing web site management or when deploying new features,
- The Import/Export module does not import physical files (e.g. web part code files, ASPX page templates, and other) when importing into an Azure website,
- The CSS Theme tab is not available. You can't manage files in the App_Themes folder via the UI,

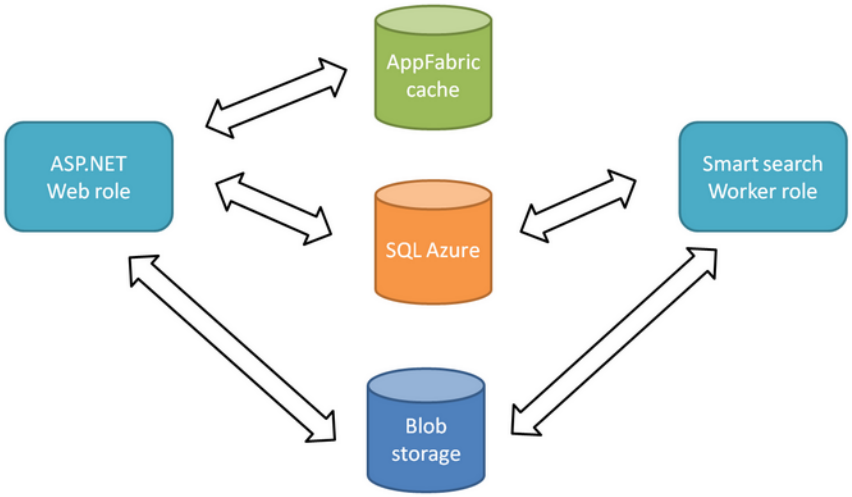

Figure 10 Windows Azure architecture

The Kentico Windows Azure project consists of the web application (CMSApp added as a web role) and the Smart Search worker role project.

The development on a local emulator is rather slow and inefficient process, so in order to keep development up to speed, you can include the CMSApp role project as

a web app in a separate project. Changes will be automatically reflected in the Azure project as well, for easy deployment into cloud,

- Since SQL Azure is almost identical to MS SQL, you can connect the local web site directly to SQL Azure running in the Azure environment. You will avoid issues with custom DB objects when migrating to the cloud instance,
- It is recommended to move media files to the Blob storage at the time of development.

When you are done with the development, you can deploy the updated version to cloud using the build-in Visual Studio Azure Tools. Assuming that you've started development as a regular website project, and you need to convert it to web app that can be included in an Azure project. For that, follow the steps for migrating existing site to Azure application. This is, however, a relatively complicated process. Alternative option to migrate existing website to Azure is to carry over all customizations from your original web site project into the CMSApp project of the new application (recommended approach).

During the deployment of such a project to the Widows Azure environment, you can use KIM with the use of the built-in deployment script or you can directly use directly Visual Studio.

- When development is done on regular MS SQL Server database, you have to convert the MS SQL database to SQL Azure with the SQL Azure Migration Wizard. You can alternatively use the Import/Export module, and import the web site (physical files will not be imported) to the Kentico Azure instance. In this scenario, you can also use the content staging module instead of the import/export approach.

C. Pre-compiled website

Website pre-compilation allows you to publish the website in the form of an assembly (instead of hundreds of project files) and that way make deployment easier and the website faster. However, there are drawbacks to this approach. In particular, there are several limitations how virtual objects (e.g. transformations, templates, etc.) are handled in pre-compiled environment.

You need to evaluate, whether additional time required for deployment (time spent on website re-compilation) is worth the performance gain you achieve at the end.

D. Nested websites

Sometimes you may need to setup nested websites in a production environment. Nested websites allow you to run another non-Kentico web application under the same domain as the CMS website and that way create the consistent look and feel of a single website (even though there are two different web applications running on a separate code base and database).

5.3 Deployment actions

Before you approach actual deployment, read through the Things to Check before you Go Live article to ensure tight security, and prevent possible security threats. Once the website is deployed to the target environment, you should check the integrity of the website by reviewing the functionality of the live site, especially:

- Check the website for broken links,
  - Links may get broken after deployment to a new environment if you have not used relative paths (using '~' in the URL) for links,
    - System generates relative URLs by default. Using the Kentico *Insert/Edit link* dialogs to create links on pages is recommended. It is your responsibility to ensure links in your custom code are using relative URLs,
    - Selecting media files from a different web site running in the same instance of the Kentico but on different domain, could result in broken links, because in this case an absolute URL is used instead of a relative one,
- Perform load tests on the production environment,
  - Check the CMS event log for any errors occurring in the production environment,
  - You may temporarily enable the CMS debugging feature to check for bottle necks in the data access layer,
- Verify the SEO accessibility of your production website.

You can use Kentico validation tools to validate (X)HTML markup, broken links, accessibility and CSS styles.

5.4 Post-deployment actions

To make sure a website in production is continuously performing well, you should regularly carry out several routine maintenance tasks:

- First of all you should keep an eye on the CMS event log. Any error in the event log indicates a possible issue that may be the result of a content update, new deployed functionality, broken service, content staging malfunction, etc.,
- It is recommended to setup the Health monitoring service (available with the EMS license) to continuously monitor the overall health of your website,
- Kentico provides a 7-day bug fixing policy. That means a new hot fix is released every week. The package provides fixes for all bugs and issues reported in the past 7 days (hotfixes are cumulative). You should review reported bugs, and apply the latest hot fix package if you see a potential threat. It is not necessary to apply every hotfix. From both maintenance as well as availability perspective, it may be wiser to approach hotfix strategically, instead of upgrading too often.

    - The hot fix package usually contains updated files for you to copy over to your website project folder and SQL scripts to execute against the website DB. The package always includes a PDF document with detailed information on how to apply the specific hot fix—read the instructions carefully. There is a Kentico Installation Manager (KIM) utility which will help you to automate the hotfix process just by following the hotfix wizard instructions,
    - In case you made any modifications to system files, review your records of customized files before overwriting the changes with what comes in the package. Hotfix/upgrade utility will inform you which system files you have customized and then you need to manually reflect your customizations,
- During the year Kentico usually releases one minor and one major version of the CMS. Even if your website is doing great, you may still want to upgrade to the latest version to get access to all the new features included within the new release.
    - The upgrade process is very similar to the application of a hot fix. You download an executable (.exe) package containing upgraded files and SQL scripts used to upgrade the DB structure. Again, the package contains PDF documentation explaining step-by-step how to successfully perform your upgrade—read the instructions carefully even if you use KIM to upgrade your website,
    - **NOTE:** In case you made any modifications to system files, review your records of customized files before overwriting the changes with what comes in package.

6 Website evaluation

When you have the project live and running, it is time to stop for a while and evaluate your efforts. Answering a few questions should help you get the big picture of how the project evolved, what issues you encountered, how the customer feels about the results, etc. It should help you to answer the most important question of all: was my project a success and what can I learn from it?

- You should establish whether all the goals of the website were met,
- Ask your client if they are happy with the website, which part of website they like or dislike, if there is anything they would change after they got a chance to see the website live, etc.,
- Determine if the website response time is reasonable based on the used hardware, hosting, etc.,
- Summarize the issues you encountered during project development,
    - Based on our experience gained during consulting sessions, it is a huge help for upcoming projects if you compile a list of issues, misunderstandings or harsh times (if any) that you went through during the development,
    - You can use such a list for future reference to avoid the same issues.
- Evaluate the SEO accessibility of your website,
- It might be useful to collect feedback on the functionality from website visitors to get the opinion of real users. You can prepare a simple questionnaire to be handed off to visitors or even create a custom feedback form displayed on the website.

Appendix A – Requirements Template

	Note
A. What is the goal of the project?	
B. What is the expected number of users/ page views during the peak load, what is the expected number of pages on the website?	
Is it going to be static or dynamic content?	
How many pages do they plan to include within the website (hundreds, thousands, etc.)?	
Does your client plan to create multiple websites sharing the same (re-usable) content?	
C. What is the structure of the website and what types of content will be published?	
Does your customer want to include some kind of document repository (for pages, video and audio files, images, etc.)?	
What are the main sections the customer wants to divide the website in?	
What items should the navigation contain?	
Is there any requirement for implementing membership areas or restricted sections on the website?	
Do they plan to display personalized content (changing the layout, styles and look of the page based on the current user)?	
D. Which web standards should be followed in terms of accessibility and coding?	
E. Who is the target web site visitor?	
What type of mobile devices should the web site support?	
Do they want to keep the same URLs for mobile devices or to have dedicated mobile section with different URLs	
F. Which products and technologies will be used?	

G. What is the content life-cycle? Who is responsible for the content management?	
What type of users would be responsible for creating, updating and deleting pages?	
Should website visitors get a chance to contribute to the website?	
If the content will be subject to workflow, what steps should it go through before being published?	
What roles would be in charge of approval in each particular step?	
When you integrate the project with some 3rd party CRM or ERP system, does it call for spreading the same change all the way to the external system?	
What task needs to be accomplished in the external system as a reaction to page changes?	
Is your customer interested in isolating the staging and production (live) environment?	
Should you remove unused pages from the website completely?	
Would it be suitable to archive them instead and keep them hidden within the website?	
H. What languages will be used for the content?	
What countries does the client plan to expand to with the website?	
What would be the differences in the content for particular language versions?	
Do they want to simply translate the text or are more significant changes planned?	
Should the look and feel of the website be modified based on the selected culture?	
What about website structure and page hierarchy for different cultures?	
Do they require translation of the administration UI?	
Is the required language for the UI ready yet (available for download at the Localization Packs page on Kentico.com)?	
I. What is the required availability of the website?	
J. What is the production environment type?	

What are the environment restrictions (trust level, pre-compiled web site, data transfer price, etc.)?	
How often do they expect to implement new features? What is the price of the deployment in their environment? How easy it is to deploy changes/updates?	
Do they plan to use CDN network to serve files?	

Appendix B – Design Template Processing

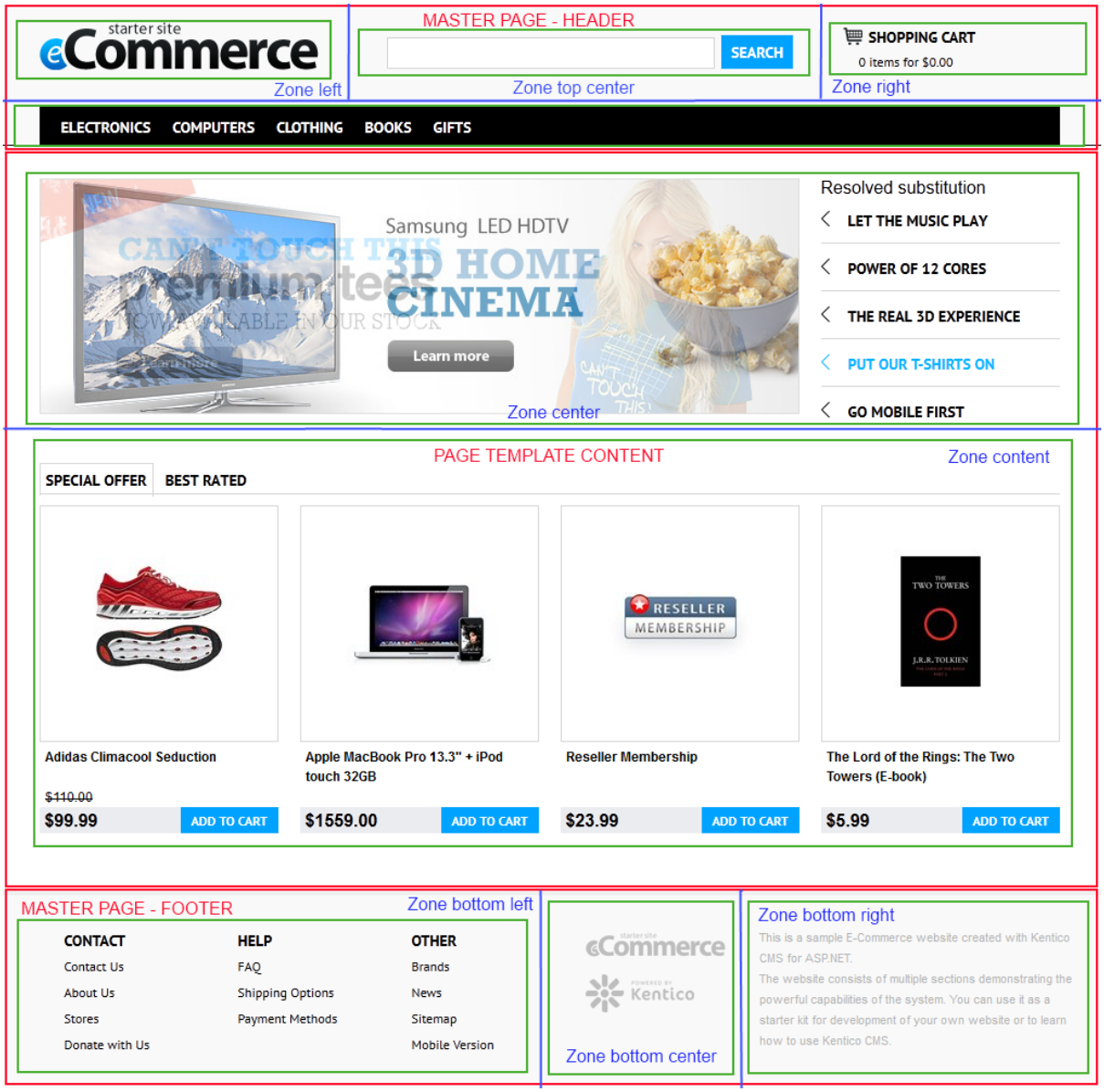

Figure 11 The red box at the top and bottom represents the master page template content while the red box in the center is the content of a page template. Blue lines split each part into zones. Furthermore, each zone is divided into several smaller parts—web parts

Appendix C – Website Wireframe Example