



THE TOP **THREE** REASONS WHY YOUR CMS PROJECT FAILS...

...and what you can do about it

With more than 15 years' experience heading up digital transformation projects, Niteco's Program Manager, **HAMMAD AHMAD** has truly seen the good, the bad and the ugly. As a result, he says, there are tried and tested ways companies can avoid the **THREE** most common pitfalls.

In this e-guide, Hammad, a past Microsoft Most Valuable Professional title holder and Sitecore specialist, reveals what they are and how you can avoid them...



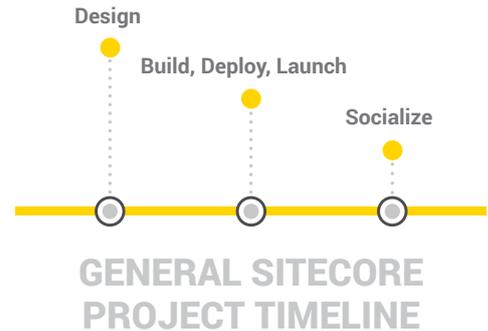
Since the dawn of the digital era, a company that doesn't have its own website, is well...not considered a 'real' company these days. It's become a given among consumers that a legitimate business has a website. However, there's a huge distinction between run-of-the-mill sites and the digital experience we all talk about. And the distinction lies in the detail. As a client, you need to pause and ask yourself: what are you looking to achieve from your website and can you invest the attention, time and budget needed to meet your goal? Furthermore, once you have launched your new site, can you adequately manage the influx of new customers that come as a result of the site?

Knowing the answers to these questions will help you define the end goal before you enter into negotiations with your chosen provider. Why? Because over the span of my career, I've learned that there is a worrying gap between the client's expectations (*which are often conveyed through Digital Agencies*) and the technical know-how needed to accomplish them. Especially when we're talking about CMS projects. Any website - existing or new - is a complex project when you add a CMS to it.

In this e-guide, using a typical Sitecore project as an example, I address the 3 most common reasons why CMS projects fail...

REASON 1

Believing the myth that a CMS project can be achieved in just a few weeks.



Why it's a problem: Don't oversimplify the work and effort required and assume it can be accomplished in weeks or a few months. For example, a design is just a figment of your imagination until and unless you have addressed the following:

- Content Structure
- Date
- User Experience
- Editor Experience, etc

After the design has been overpromised, often the Develop, Deploy and Launch phase is also gravely underestimated, especially when content structure is taken into account. Ensure the project team delivers what you're expecting by asking for their input and recommendations during the initial deliberations. It is the responsibility of the program manager – your main contact, to set out achievable goals.

REASON 2

The myth that content planning and content driven design should be worried about after technical implementation.

Why it's a problem: Time and time again, companies make the mistake of misjudging the amount of content needed and how long it will take to acquire. In my experience, this activity is what delays the project the most. It's important to plan and prioritize content migration and make those page templates ready as early as possible to avoid lengthy delays and minimize the need to revise the content.

REASON 3

The myth that End Users will swiftly get to grips with the new site, and if they don't a complete re-construct is needed.

Why it's a problem: Most people don't like change, especially regular customers. We often tend to forget the end users and their likely reaction to change may not be as positive as marketers may hope. But please avoid the urge to be impulsive. Allow a period of adjustment before making any technical and/or aesthetic changes to your site. Analyze the Usage Report before making a definitive decision.

ADOPTING THE RIGHT APPROACH

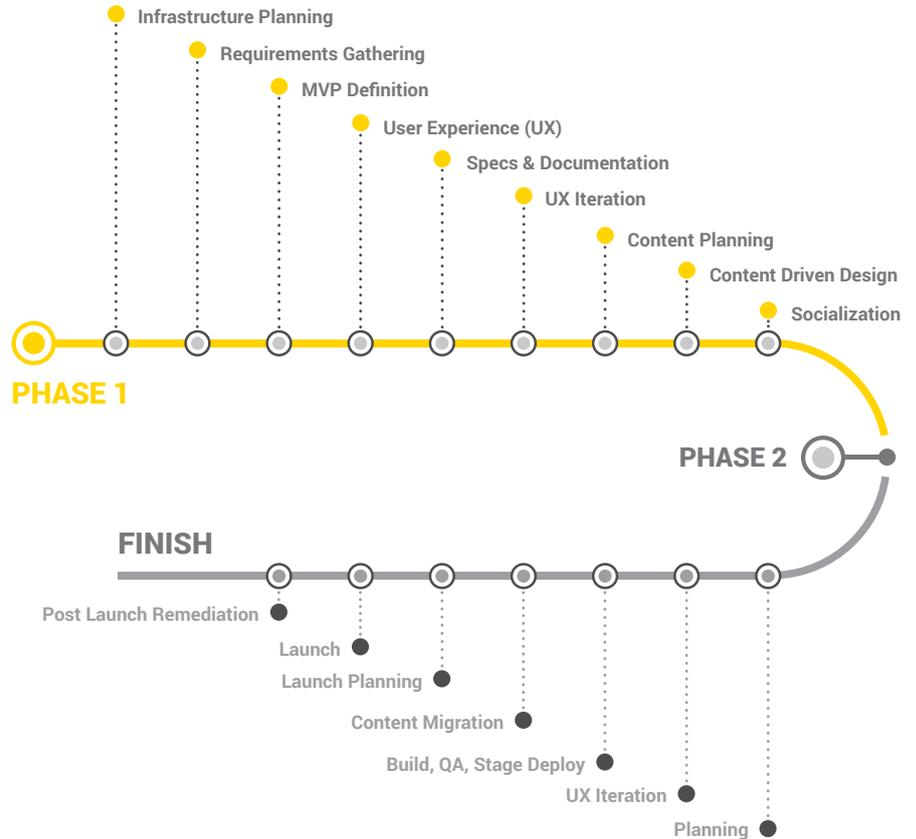
Now, you know what the three biggest reasons are and why they are so problematic, here I share with you the best approach to take when commissioning a new CMS project.

Hint: ***It's all in the planning!***

FAILURE TO PLAN IS PLANNING TO FAIL

As mentioned previously, the distinction is in the detail. To keep the project ticking along nicely, make sure you've adequately addressed the following key areas:

GENERAL SITECORE PROJECT TIMELINE



INFRASTRUCTURE & LICENSING

Sitecore requires a number of infrastructure components in order to operate so it's essential to know and understand how licensing would affect your implementation.

To determine the correct needs, ask yourself the following:

- Will you be utilizing perpetual or consumption-based licensing?
- How many concurrent editor licenses will you require?
- How many database and content delivery servers are needed – for production and non-live environments?
- What about add-ons like Sitecore XP, Sitecore Experience Accelerator (SXA), Federated Experience Manager and other elements?

REQUIREMENT GATHERING

This is crucial and critical if you want a successful Sitecore Project implementation and zero

documentation means certain failure. A lot of people think that just by listing out the main functionality they will be able to cover all the requirements and this ends up in all sorts of confusions for the Dev team. For example, if you want a search function, you must document how this should work; would it display results in a pop up window or in a new page? What would the search result page look like? Would there be any filters?

If you don't clarify and document your expectations in detail and your Lead Developer or Product Manager happens to leave mid-project, what would you do?

This also means that you might have to revisit your design and redo a lot of things. It's more effort than just jumping right into design, but it will be worth it in the end, because UX doesn't always produce the necessary functionality without guidance from strategy first.

- What must your new Sitecore site be able to do?
- What would you like it to do?
- What are your stretch goals and features?

MVP (Minimum Viable Product)

The majority of us do not think about MVP and try to commit and deliver nearly half of the Project in Phase 1 which in most cases is unrealistic. The whole idea of MVP is to meet and set expectations for large builds within realistic timelines. So you need to define your minimum launch criteria before you begin your build. One should start the work towards that initial product launch before starting any major work on your features to ensure your timeline is met.



DESIGN & UX ITERATIONS

This is how a Sitecore or any other CMS Project should look like:

- Design
- Review
- Re-design
- Tweak that design
- Build
- Oops, found a problem, so another tweak to design
- Finish the build
- Test
- Missed a design element, so add that to phase 2
- Test some more
- Deploy

The most important factor is that design iteration is necessary and should be accounted for early on in a Sitecore build project. You won't get it (all) right the first time. You will have to go back to it a few times.

CONTENT PLANNING AND CONTENT DRIVEN DESIGN

Content creation is usually the forgotten factor in technical projects. Avoid the pitfall by thinking about content as early as possible because content usually needs approval which will impact your project timeline. Get wireframes and mockups to help visualize the data for the page, which in turn will help ensure your template has all the required fields – meta data, titles, images etc - to hold the content. Bear in mind also that Content Authors are not usually familiar with Sitecore, so don't rush into content creation without getting the template done first.

As your project develops, content writers should be working in parallel so that one task does not delay the other and the content suits the design of your page.

SOCIALIZATION

The socialization of a technical project like a Sitecore website build is related to, but separate from engaging with other key stakeholders in your organization. Stakeholder meetings and brainstorm sessions seek to inform how the site will behave, how it will look and the goals that need to be set in order to turn the vision into reality. Socialization is more about awareness and acceptance than functional definition or design influence.

In my experience, it's a lot easier to socialize something that you can visually present, but you shouldn't wait until alpha, beta or launch before you begin the socialization process. I accept feedback during the socialization process, but I'm also careful to set expectations of change with those providing feedback during my socialization efforts.

QA & UAT STAGE

It is extremely important that your QA team looks at the site from 360 degree perspective and not only from Tech purpose QA angle. The mindset needs to be of a Product Owner who has been involved at different stages of the Project. One very important point is to be responsible while testing and evaluating the site as you can mark items as improvement or suggestions based on your experience of how a certain feature should work on the site. Trust me, a lot of the time, customers are also looking for further ideas therefore they often seem to welcome such approach.

GO LIVE AND POST LIVE

Other very important tasks to think about before jumping on the Go Live bandwagon are:

- 404, 500, 301 redirect strategy?
- XML sitemap?
- Search Engine submission

- Robot.txt
- Google Analytics, Google Tag Manager, retargeting tags etc
- RSS Feed
- Performance Testing & have you optimized it
- “no index” for Stage site
- Security

Now you're good to go! Just remember, it is always advisable to have a gap between your deployments, and the Go Live Date must be announced to the public after a day or preferably after a weeks' time. I have seen a couple of customers using tools which lets them collect data and also asks the visitor if they would like to visit the new site or the old site, giving them the option to leave feedback as they exit the site. These are great ways to collect some valuable insights.

Follow these recommendations, leave the lines of communications open between the customer and the project team and your Sitecore project will be a storming success.