In last decades the process of website development and dedicated systems development has changed completely. From the simple text files, with the power of PHP and JavaScript, when everything had to be done manually, through Excel and database files, thrrough simple CMS systems to advanced visualization tools and dedicated environments. Currently, there are many different systems, that differ a lot in functionality, a built-in set of tools and purpose. But, first things first, let's take a closer look at what management system is itself.

What is "Management System" Exactly?

Term "management system" is very wide in it's nature and doesn't necessary constrain to business side. For example, any tool that helps to manage your home financials, is also a management system – essentially we could call it HBMS (home budget management system). It can be online or offline – you could write advanced macros for local Excel file, and it would be local management system for your purpose. So widest definition would be "any tool that helps you to manage something"... but this sounds a bit to wide, because then even a piece of paper could be called "management system". If we define a "system" as "something that automates or helps to automate tasks" this becomes clearer, as then "management system" could be defined as "*any tool that helps you to manage something with built it automation and facilitation of processes*".

But enough of the theory – obviously such topic would be too wide in it's content to be constrained in one article. Let's then constrain our topic: today we are interested in business use cases and ONLY tools that work online. This still gives us pretty wide range of tools to consider, but their usability is crucial for any business nowadays and tool comparison will be possible within similar aspects.

Let's start with most common one and some explanation – CMS which stands for "Content Management System". It is so widely used, that it is overused and essentially became synonym to most online SaaS tools, which simply is not true. On the other hand, many of such tools allow you to manage *some* content – for example, articles, media, users, and so on. So as you can see, the border is quite fuzzy and tools often have functionalities from "different category". Because of this I will avoid using CMS abbreviation and will try to use more precise definitions that by default will adhere to **main management system functionality or purpose**.

First thing that comes to your mind when talking about CMS is probably WordPress.

Second one would be Joomla, Wix or any other tool for managing websites and blogs. So for our needs, we will call this group of tools WCMS, which of course stands for “Web Content Management System”. So essentially it’s a system, that allows you to relatively easily manage the content of your website along with other elements connected to it. I might be here in minority, but I like precision, so I’ll keep using it for this article. But let’s start with listing some types that we might want to discuss.

Kinds Of Online Management Systems for Business

Predefined Management Systems

There are many of such tools specializing in different aspects of business, and we’ll go through most popular of them today. Such tools can be narrowly specialized in their aspects. Here is some short list of such:

1. Content Management Systems (**CMS**) or Web Content Management Systems (**WCMS**): These platforms allow users to create, edit, publish, or modify content on the website. Examples: WordPress, Drupal, Joomla
2. Customer Relationship Management (**CRM**) systems: CRM software helps businesses manage customer interactions and data throughout the customer lifecycle. Examples: Salesforce, HubSpot, Zoho CRM
3. Enterprise Resource Planning (**ERP**): ERP integrates all departments and functions across a company into one system to improve efficiency and productivity. Examples: SAP, Oracle Fusion Cloud ERP, Microsoft Dynamics 365 Business Central
4. Human Resources Management Systems (**HRMS**) or Human Capital Management (**HCM**): These systems help manage the human resources of an organization more efficiently. Examples: Workday, Bamboo HR, ADP
5. Learning Management System (**LMS**): LMS platforms are used to plan, implement and assess a specific learning process. Examples: Moodle, Blackboard Learn, Canvas by Instructure
6. Project Management Systems: These tools help teams manage projects more efficiently from start to finish. Examples: Asana, Jira, Trello, Monday.com
7. Supply Chain Management (**SCM**): SCM systems are designed to execute supply chain activities in an efficient way and better meet customer requirements at lower costs. Examples: SAP Ariba, Oracle SCM Cloud, Workday

Supply Planning
8. Transportation Management Systems (**TMS**): TMS software helps manage the day-to-day operations of a transportation network. Examples: BluJay Solutions, MercuryGate, TMW Systems
9. Financial Management System: These systems are designed to help businesses handle financial transactions and reporting more efficiently. Examples: QuickBooks, Xero, Sage Intacct
10. Marketing Automation Platforms (**MAP**): MAPs automate marketing tasks such as email campaigns, social media posts, etc., helping marketers reach their target audience effectively. Examples: Marketo, Pardot by Salesforce, HubSpot Marketing Platform

Wide Range Use Management Systems

Besides such groups that are comparatively easly to be defined there exists a group of tools that connects functionalities of the above to provide possibly best outcome. A great example can be **e-commerce tools group**, to which belongs to systems such as Magento or PrestaShop. You can easily see that websites utilizing such tools can be used not just as a selling platform, but also as simple warehouse management system, dropshipping automation, finance management system or even blog. It is truth that you can build your business around such tool and adjust to it’s limitations and concepts, but it is also truth that you can use only functionality that you need and what it was designed for.

It is also important that you actually can use plugins to extend or amend core functionality of such tool. Great examples can be WordPress or Joomla. And while WordPress is maily a blogging platform (and all concepts, structure, assets management etc) is done around blogging, by utilizing poweful plugins – such as WooCommerce – you can actually craft your own combination that suits your needs.

I must admit though that this can sometimes be a short term approach because of the core concept; remember that no matter the plugin selection you will always be constrained by it. I have seen for example WordPress sites that tried to be e-commerce with thousands of products, which often caused troubles, because WP core is simply not optimum for such use. And vice versa – I have seen Magento shops with just dozens of products, which was total overkill for such small amount, causing unnecesarily high operating costs. So be sure to really know the tool before commiting whole business to it – or schedule a call with software house such as Sailing Byte when looking for advice.

Another interesting tools are – let's call them – ecosystems, that actually consist of many complimentary tools. Such ecosystem can be found of course at Google itself, but Odoo might be actually a better example as **general online business toolset**. Zoho and Hubspot are also a great examples of such approach. Undeniable benefit of such approach is having everything in one place and often seamless synchronization of data between the systems. Problems often are: high price, someties complex usage or hard-to-impossible migration to another tool. Although definitely worth considering.

No-Code Platforms for Business Use

This is itself quite wide group of tools, where we should define subgroups to think what they are designed for, although they do have a lot in common. Most important thing is probably that they are designed to provide flexibility and nice look, but not necesarily optimum, secure or open code and functionality. And once you are stuck on the platform it might be hard to migrate. But they might be optimum selection for some businesses or part of businesses, so let's take a closer look at their groups.

First group are website builders – such as Wix, Webflow or Squarespace. They are designed as systems which allow you to create and manage websites, often advertised as alternative to WordPress (which can also utilize plugins to provide similar functionality – like Bakery or Elementor). They can help you to launch website fast – for example to test idea against market.

Second group are app builders – such as ToolJet, Budibase, Bubble, Flutterflow. And they aswell provide great starting point, but building more advaced systems can be really a bottleneck or problem generator. I also know some cases that needed to migrate because of performance issues.

Third group are "excel on steroids" or "databases on steroids". If you are strugging with Google Sheets or Excel files, or you would like some additional features – like form creation, column type constraints, collaboration, easy charts and so on – then you shoud definitely take a look at Airtable, Baserow, Seatable or Clickup. Be sure though that you test it out first, because while they seem good, migrating from Excel can be pain – especially if you are using advanced formulas. In that case it is probably best to contact us so we can analyze your Excel files and turn them into SaaS.

Fourth group are visual automation tools. And first thing that comes to your mind is

probably Zapier or Make. Those two tools are good, but for me N8N worked absolutely best, and without any privacy concerns as it can be self hosted. Anyway, such tools are great for automating small business processes, messages, strategies or prototyping. An interesting example for you might be DMARC monitoring automation, although I must admint that it's just one of over 20 (maybe 30) other automations I have on N8N (such as website monitoring tool, automated Slack monthly summary, AI analysis and more).

And recently there is more and more apparent new – fifth group – that is AI-based builders. This is very interesting group, as it can speed up development process at the very beginning, but doesn't have constrains of other no-code platforms. While in general such generated code is useful for fast start, it also needs to be improved and adjusted before production use. But at adjusting stage you can add functionality that wouldn't be possible with no-code platforms, and which can make your SaaS absolutely unique.

To sum up, I will repeat this again as this is very important conclusion for all cases above, but no-code platforms in particular: they can be great tools, but when selecting such tool you need to understand their limitations and use them in proper-designed way, otherwise you might be bottlenecked at 90% usability and expensive migration in near future, effectively paying twice for desired result.

Unique Management Systems and Platforms

This category consists only of systems that are designed from the very beginning to the very end to serve specific purpose. They are often startups and have very limited competition on market that serves the same exact purpose. That's because they are dedicated for specific functionality by their nature.

One example can be CargoSELLER – a system that calls itself "not a TMS", but "Spot Freight Management System". Currently there are very few on the market and I do really doubt that you could find open source alternative that has similar functionalities. And their AI addon is really doing magic!

Another similar example can be MyRotat – which is not typical HR nor worktime tracking system, but lies somewhere nearby them. Although, how would you call such software – RMHS (Rotational management human resources system)? A bit too long I'd say… and definitely unique proposition on the market.

How to Find Management System for Me?

First, **define main purpose** of tool. If you want absolute best tool for the process, it must be in it's core. For example, there are CRM plugins for WordPress, but if your main use case is CRM, then choosing WordPress because of plugin is most probably wrong way to do it.

Then, **make a list of functionalities** that you need to cover. And – what is very important – order it from most important to least important, possibly with adding property if it is "required". For example, your top 10 functionalities might be an absolute required, next 10 might be "nice to have", next 10 might be "would like in the future" and so on. This actually can be a really good practice for working with or refining your Business Model canvas.

Third, consider **who will be using this system**, and maybe ask them for opinion. System will be useless, if you need people to cooperate on it, and they will refuse because it will for example cost them more time than doing same work manually. This also applies to yourself – you should not be forcing yourself to use system – it should be useful in a way that you will want to use it.

Fourth but not last – **test it out**. It is absolutely necessary and possbly can be a first or second thing on your list, if you are on "discovery phase". Besides lerning more about necessary functionalties, it can help you filter out tools that are buggy or hard to use.

What Aspects to Consider for Management Systems?

Of course – this list is definitely not complete, but should give you the base for thinking about finding right system. So every aspect below should be refined with statement: **How important is cosindered aspect in perspective of desired use?** And that question should be easily answered if you considered point above about "making a list of functionalities".

Designation and Functionality

Is main system designation compatible with main aim you want to acheive? Or if system gives you ability to craft it's main purpose? If you combine it with desired

functionality, then you have covered what was discussed in previous paragraphs.

Performance and Scalability

Some management systems are often written in a way, that allows you to do a lot, so are flexible – but this also means that they might not be optimized for performance – sometimes need to load a lot of dependencies to be working properly. They are also not free from backward compatibility, which adds to the stack a lot of code, that could've been avoided. Completely unnecessary requests to the database take up your visitor's time and increase website loading time.

Perofrmance is crucial not only for your visitors convenience, but also impact your position on Google – and if tool is used internally, it can add additional cost of time for running your business, multiplied by number of users.

It is also common case that with number of assets or number of users perofmance significantly drops, to the point that system is not usable. I have seen such cases in self-written systems and they often could be avoided if programming engineer was included in the process. Scalability is not limited to performance of the code and system itself – it is also a matter of ability to launch parts of the system on different servers, redundance readyness, ability to run in swarm or behind load balancer and more. You should especially consider this when ordering SaaS, and if you are working with Sailing Byte then you already know that we take this elements seriously.

Extendibility and Security

Some open source systems have repositories full of plugins or extensions. Those plugins can greatly extend the basic functionality by adding extra options that can be helpful or even essential. But there are few problems with this: anyone can post a plugin – this results in publishing on live systems homemade codes, that are neither secure nor properly written. Any modifications to plugins lead to a lot of additional work: digging in code, paying extra developers for help or result in breaking compatibility with the update system. Most website hacks happen not because of security issues within system core, but in the plugins!

On the other hand commercial or proprietary systems often have limited amount of plugins in their repositories. The difference is that code should always be developed by professionals, which also leads to more security and consistency between

plugins. Modification done for the client can be tailored exactly to user needs without breaking any compatibility; so it’s a win-win situation.

So, the more plugins you have, potentially the more website or SaaS is in danger of being hacked. Also, the more plugins, the more dependencies and potentially more fragile system can be – especially on processing updates. Answer “where is best point between extendability and security” is not easy, because it is not a one-dimentional dillema.

Additionally for decades there is discussion on the Internet which is more secure: open source systems where anyone can check code and many people can fix code, but also leaves hackers free hand to analyze code and use potential issues? Or non published private code that only limited number of people check and know (so potentially not that well-reviewed), but is also a *blackbox* for potential hackers so any hacking attempt is a big guessing game? I don’t believe there is definite answer to this question.

Overall: there is no one-fits-all advise for this aspect, so consider this elements wisely.

Licensing and Commercial Use

Open-source software is sometimes based on licenses, that complicate the commercialization of project under some circumstances. Also on some licenses, you are actually forced to publish all the code you have written or paid for! But what if code has to contain information or know-how that is patented or has to be especially secured?

Proprietary systems do not suffer such limitations. Especially if some coding work is being done exactly for you, you can agree on licensing with the development team. And you can be sure, that know-how, flows and other sensible elements are treated with proper care and security. At Sailing Byte, we always transfer all rights to code by default to our clients.

Prices and Costs

Of course, having free CMS might sound terrific, but will not always be the best option. As a concious businessman, you should calculate long term costs of operating such system, not just buying cost. So think about such aspects:

- What is the initial launch cost?
- What is hosting cost?
- How much do I need for additional plugins that require my attention?
- What is cost of keeping software up to date?
- What are operational differences cost (if there is something that takes a long time on one system but is very short on second system)?
- Can system be adjusted to my needs and if so is this costly?

These are just sample questions and you should expand them according to your needs.

So what is the best Management System for you?

Again, there is no one-fits-all answer to that. A lot depends on your actual needs. In general, in some cases, open-source systems can be safely used and provide a lot of value. Although you must take into notice, that they will require more work from you (or delegating this work) – along with possible problem fixing, updates or adjusting plugins code. But if you are looking into a very professional coding service or some advanced data processing, then proprietary software is the way to go – and you might want to contact us using form below. So thank you for reading and I hope to hear about your system or SaaS idea soon!