

Plan now or pay later: Sustainable design ideas for digital asset management

Received (in revised form): 13th March, 2018



Michael Gellner

is Chief Technology Officer and Lead Developer for Modula4, the company he co-founded in 2004. He has been directly involved in more than 150 digital asset management implementation projects for the company, and is credited as the chief architect of the company's many Cumulus add-ons.

Modula4, 720 A Street, San Rafael, CA 94901, USA
Tel: +1 415 869 8645;
E-mail: mgellner@modula4.com



David Diamond

has been working in the digital asset management industry since 1998 and is the author of the books 'DAM Survival Guide' and 'Metadata for Content Management'. He has also directed global marketing for Canto and Picturepark and created the DAM Guru Program. He has won both 'DAMMY of the Year' and 'Top DAM Influencer' awards.

E-mail: david@airdiamond.com

Abstract Despite what the vendors promise, 'out of the box' digital asset management software is never ready for use. The ultimate benefit one can derive from such software will depend on careful consideration of one's needs, and the planning of one's solution. This paper summarises some important points to consider, and provides the guidance necessary to move forward with the purchase and implementation of a digital asset management initiative.

KEYWORDS: automation, business process management (BPM), customisation, planning, policy, user experience (UX)

INTRODUCTION

A digital asset management (DAM) system is a tool used for the organisation, preservation and controlled distribution of digital content. A properly planned system adheres to clear, sustainable content management policy, through which it enables users to archive, find and use content with less effort, greater results and fewer errors.

Although content management policy is best defined by humans who are knowledgeable in the workings of the

content, organisation and industry, much of the execution of that policy can be managed by software. This frees human users to do what they do best, while it increases the return on an organisation's investments in software and hardware.

Among the most common types of customisation are the development of automated workflows and purpose-specific user interfaces that free users from having to learn and understand the giant, monolithic interfaces provided by DAM software makers.

The concern is that custom design and development can be expensive and time-consuming, and it can lead to disastrous results if not done properly.

Some manual workflows are good candidates for immediate automation, while others might be better left for automation at a later time, or never. Likewise, some user interfaces — no matter how horrible they appear — cannot be easily improved upon without significant investment.

This paper aims to help DAM managers assess:

- which manual workflows are good candidates for automation;
- where user productivity could be improved through the development of a purpose-specific user interface; and
- how to identify which customisation projects are better done up front, and which are better left for a later time.

It is important to recognise that proper planning for customisation might result in the decision that no customisations are required. What is most important is that a thorough analysis be conducted so that there is agreement on the decision made, and that the decision be documented so it can be reviewed periodically.

CUSTOMISATION VERSUS CONFIGURATION

Before getting into specifics, it is important to understand the differences between customisation and configuration, as they pertain to enterprise software. The bulk of suggestions mentioned herein require customisation services, which should be considered and planned carefully.

Virtually all professional DAM systems enable managers to configure the system for different requirements. ‘Customisation’ refers to system changes or additions that cannot be done using the system’s standard configuration options.

The primary advantage of customisation is that the system can be tailored to meet specific needs that are beyond the scope of the standard system. In some cases, significant return on investment can be achieved only through customisation.

The potential downside of customisation comes from incompatibility with future core system upgrades. In other words, a developer might ‘hack’ into the core system in such a way that the customisation — no matter how well it might work at first — might break when the core software is upgraded by the manufacturer.

To ensure customisations are as future-proof as possible, it is a good idea to commission the guidance and services of a developer who has a long history with the software maker. To further protect customisation investments, ask developers what guarantees and assurances they offer with their services.

FLEXIBILITY VERSUS SIMPLICITY

One concept that workflow automation and custom-built user interfaces share is a sliding scale of flexibility.

On the ‘easy’ end of the scale are customisations that enable users to work with little or no additional training. For example, an upload widget that prompts the user to select a file and then performs the upload, adds metadata, notifies an editor and sends an ‘asset received’ e-mail to the uploading user sounds pretty simple (particularly when compared with the cumbersome upload processes imposed by some DAM systems.).

The value of such a workflow diminishes when the ‘what about when’ exceptions are factored in. What about when the user wants to upload multiple files; what about when one editor should be notified in one situation, but another editor should be notified in a different situation; what about when the uploading user is not known to the system, so her e-mail address is not available for the confirmation e-mail?

The most obvious solution is to ask the uploading user for additional information. But the more that is asked of the uploading user, the more difficult the process becomes. Although it might seem simple to choose a different editor for two different assets, is it reasonable to expect that the uploading user will know whom to choose?

On the more flexible side of things lies greater complexity. Flexibility implies options, and options require decisions. In turn, decisions require user knowledge — and where user knowledge is unavailable, support must be provided.

When an automated workflow can make those decisions based on factors like the uploading user, the location, the type or size of the asset, or some other information that can be gathered without undue imposition on the user, that is ideal.

Where one must depend on the user to make decisions to guide the workflows, it is essential to consider some important factors about that user:

- Will the user know the information required?
- Will the user have time (or incentive) to provide the required information?
- Will the user know how to get help, if needed?

In addition, it is important to ensure the automated workflow can account for missing or erroneous information.

One of the most valuable benefits of workflow automation is reduction or elimination of human-introduced errors. In planning for these automations, it is important to consider whether the workflow will actually improve things in this regard.

Know your audience. Consider carefully what is reasonable to expect users to know and understand. While the goal might be a system that requires no training, there is a certain level of complexity to the

management of content. If making things easier on users results in making things more difficult for the editors or managers who need to clean up the mess, a net benefit might be difficult to realise.

EVALUATING MANUAL WORKFLOWS FOR AUTOMATION

The starting point here is a document that specifies existing manual workflows.

The more complete this inventory of workflows is, the more likely it will be possible to identify which of those workflows are good candidates for automation. Further, when speaking to consultants or developers, it will be possible to provide them with the information they will need to assess the situation.

Such a document can be simple in structure, requiring only a few points per workflow:

- What is the goal of the workflow?
- Who currently performs the workflow?
- How well is the workflow presently working?
- What are the expectations for automation of the workflow?

The consultant will likely have questions about the specifics of each workflow, but this structure can help provide the basis.

Consider the following example:

Our ‘Copyright Verification’ workflow is intended to ensure that a valid and appropriate copyright is assigned to each asset in the system. Until such a copyright is provided, the asset is not available for access by users, though it remains available to editors and managers.

Currently, (human) metadata editors perform a daily search for assets that lack a copyright or that were added to the system the previous day. They manually verify the contributor-provided copyright notices, or they add copyright notices, as needed.

The efficacy of this workflow depends largely on the editor performing the verifications. Junior editors do not always know when a copyright is valid, so they have to ask for assistance. In some instances, typographic errors introduced by the editors result in erroneous metadata values.

Based on the contributing user and other metadata values, we think it would be possible for automation to ‘rubber stamp’ the appropriate copyright for each asset. Ideally, this would happen right after the asset was added to the system so that it could be made available faster, without user intervention. When the software cannot determine the absolute correct copyright, the asset should be flagged and a notification should be sent to an editor.

Within this description is all that a consultant would need in order to start asking the right questions.

For example, while the requirements of this workflow are clear and relatively simple for automation, one good question would be: *How many assets are added to the system each day?* If the system takes on hundreds of assets each day, the value of automation would be much greater than it would be if only a few new assets are added weekly.

An additional advantage to documenting each workflow is that the document can be shared with others to make sure that everyone is in agreement about the workflow.

Good candidates for automation now

The following points can help to identify which workflows should be prioritised for automation:

- the manual workflow is among the most common in use at the organisation;
- the manual workflow is taking considerable staff time or causing considerable delays in content access;
- the manual workflow results in metadata or permissions errors that are costly or that adversely affect the quality of the system; and
- the consultant has demonstrated a clear and affordable path to automation that will yield tangible value.

That last point is particularly important. In some cases, a consultancy will suggest they ‘try a few things’ to see what they can come up with. In some cases, experimentation is unavoidable, and the outcome can be well worth the effort; but make sure the time taken is not at the expense of more easily created automated workflows that could start yielding benefits immediately.

Good candidates for automation later

In some cases, there is no urgency to automate a workflow. Further, *now* might even be a bad time for the automation. Following are some of these considerations:

- changes to the current workflow are expected, based on evolving business requirements or other factors;
- required interfaces to other workflows, hardware or other resources are not yet in place, making even a well designed workflow automation less valuable; and
- funds are limited and the workflow has been identified as not being critical.

Bad candidates for automation

There will also be find workflows that are not worth automating. In some cases, these might be the lion’s share of workflows in use at the organisation. For example:

- the workflow does not occur very often;
- the workflow does not require much human-time investment; and
- there are too many variables to consider, making the workflow difficult for users, or harmful to system data integrity.

These ‘bad candidate’ workflows might also be those that require the most subject matter expertise. As such, they can be tempting candidates for automation.

However, human expertise can often be difficult to automate.

As a case in point, consider the various auto-tagging services available. If you have a photo of a rose on a summer day, and the only tags required are flower, daylight and red, then an auto-tagging automation service might work well.

However, if the tagging requirements include categorising the rose into subspecies, or using industry or company-specific names, or adding tags that are not apparent in the image itself, such as client names or project IDs, then these automation services will not produce the same value as a trained human.

CONSIDERING WHERE CUSTOM INTERFACES CAN HELP

As software application programming interfaces (APIs) have become more flexible, developers have been able to create alternative user interfaces. Most popular are purpose-built interfaces, which are (usually) less complex interfaces that enable users to perform specific tasks, such as uploading content, performing fast metadata edits, or similar tasks.

Software makers typically provide monolithic interfaces that enable users to access all features of the program. Better interfaces are designed with the prioritisation of the most common user activities in mind. But DAM systems as a whole are not known for their elegant user interfaces.

When considering where a purpose-built interface can help, consider the following.

- Is there a group of users that only ever access certain program features?
 - Does the DAM system interface make common tasks time-consuming or cumbersome?
 - Does access to program features need to be restricted for some users?
 - Is it necessary to provide an interface that is branded or designed in a certain way?
 - Does DAM functionality need to be embedded into other business systems?
- A simplified user interface might well help in a number of instances, but, as with workflow automation, ease of use must be weighed against flexibility.
- Manufacturer-provided user interfaces are large and complex for a reason: they must provide access to all aspects of the system. If the developer promises a simplified user experience, that most certainly means reduced functionality. For this reason, it is a good idea to think of custom interfaces as additions to the standard interface that comes with the software, not as replacements.
- When thinking in terms of the management of digital content, there are a number of instances that could benefit from a custom user interface:
- *Uploading assets:* Some users need to upload new content regularly and quickly. Providing these users with added convenience, such as drop folders or simple widgets, can improve their productivity and increase the likelihood that they will contribute content to the system.
 - *Editing metadata:* Simplified metadata editing interfaces can make adding tags and editing textual metadata less fatiguing and less prone to error. In addition, invitations to edit metadata can be sent to external subject matter experts (or crowd-sourced), without having to provide training or create user accounts in the system.
 - *Commenting:* The system might be enhanced by permitting non-users to see and comment on the assets in the system. Alternatively, there might be production processes for which it makes sense for managers or others to be able to easily see and approve content changes.
 - *System administration:* In many cases, the administrator of a DAM system has no involvement with the content therein. To make administration of the system more convenient, a purpose-built interface could offer access to common administrative

features, such as creating or approving user accounts, changing access permissions for user roles, or running reports.

In most cases, a purpose-built interface is designed with one of the following in mind:

- user group;
- system function; or
- platform.

When the system's user administration is based on the roles of each user, such as content contributors, metadata editors, content sharers, etc, one can more easily imagine custom interfaces that provide ready access to the functionality those user groups require, while shielding the users from program features they do not need.

In other cases, there might be specific program functions that could be simplified and made available across a number of user roles. Commenting is one example; social media sharing is another.

Device platform is an increasingly important reason for the development of purpose-built interfaces. Virtually all DAM users would say that mobile phone or tablet access is important today — but for what purpose?

- We take photos of real estate and want to upload them into the system.
- We need to be able to access content on the fly from within sales meetings.
- We need to be able to find and share content in a format suitable for social media.

These (and other) requirements are good reasons to provide a user interface that works well on mobile devices.

The value of mobile access diminishes when the very nature of the device would adversely affect productivity. For example, editing the underlying structure of the system, such as adding fields or changing layouts, might prove so cumbersome on a

smaller display, that the value of offering such a mobile interface is eliminated.

As when prioritising the automation of workflows, consider the realities of the interface requirements. If, for example, only one employee ever makes system configuration changes, the investment in a sexy mobile interface for that purpose might never be justified.

On the other hand, where the goal is to increase system use or user buy-in, the development of custom interfaces may be ideal.

WHAT TO DO TODAY AND WHAT TO LEAVE FOR TOMORROW

In addition to considering the specifics of each workflow or interface idea, there are a few considerations that affect the system as a whole. Use these to prioritise custom development projects:

- *User retraining:* Getting users to adopt a *new* system is not always easy. Getting them to willingly accept retraining to adapt to changes can be virtually impossible. If, while setting up a new system, it is possible to identify areas in which large-scale automation changes are worthwhile, considering developing automation (or custom interfaces) before system launch.
- *Data integrity and completeness:* If automation or interface customisation can be expected to increase data integrity through the reduction or elimination of human error, those projects are worth considering sooner than later. On the other hand, the metadata data set (the fields you use) is expected to change within the next year or so, it might be worth holding off on any custom work until things have settled.
- *Policy changes:* If content-related policy is likely to change substantially, it might be a good idea to wait and see what becomes of that policy. As a reminder, automation

is designed to adhere to policy, so if that policy changes, there might be significant additional development costs to bring existing automation into compliance.

- *Business changes*: Should the organisation acquire other organisations or be acquired by another organisation, it might become necessary to merge systems, data or both. This process could be made more difficult if the system has been heavily customised. In fact, when that time comes, you might wish that automation budget was still available in order to make the merger easier.

WHAT TO DO EARLY, OFTEN AND FOREVER

When introducing a new business system, or making improvements to an existing system, there will likely be some resistance from the very people the improvements are designed to help. Common complaints include:

- I don't have time to learn something new.
- The way we do things now is fine; why do we need a change?
- Will this make my job less secure?

Users also tend to be hypercritical about any such changes, even when the benefit is clear and the execution is flawless.

Most employees learn what is asked of them, improve where they can and do the best jobs they are able. For those directly involved with the management of content, the introduction of a new system can be seen as a not-so-subtle indication that the work they have been doing is not good enough.

These users can become dangerous to the long-term survival of the system, or they can be converted into system advocates — the outcome can be directly attributable to two primary events:

- Is the system performing well and meeting user expectations?
- Are the advantages of the system being promoted to users and management?

It is possible to address the first point using personal experience and through interviews, surveys or simple conversations with users.

The second point requires some planning and budget. Consider the following question, which is better: Android or iOS?

No doubt you have an opinion. Perhaps you are so convinced that one is better than the other that you encourage friends to use your platform of choice.

But the fact is, when it comes to mobile platforms and so many other things, 'better' is a factor of marketing. Do you prefer iOS because Apple has convinced you of special benefits it offers? Do you prefer Android because there is something about Apple that you do not like?

In fact, most users have a preference, although few can quantify that preference.

Short of any system issues or limitations that actually stand in the way of user efficiency and convenience, the same will hold true for any business system that may be introduced or customised. Some folks will like it; some will not.

Basic marketing skills can be used to help users see the benefits the system offers. In doing so, those users will become less resistant and (ideally) might even start encouraging others to use the system.

There are a few things to consider for such marketing campaigns:

- *Early notices of changes*: Give users advanced notice about changes. Enable them to prepare, both mentally and with regard to their duties. Offer them an option to ask questions or offer advice. Listen to that input. Whether or not that input is accepted, explain why.
- *Reminder announcements about changes*: Remind users that changes are coming. Depending on the complexity of those changes, choose a period of a month, week or day — ideally offering multiple reminders. Options for reminders are e-mail, company intranet pages or, if

- possible, mention the changes on the system login screen. Include a link for more information and feedback.
- *Training options:* Provide users with all the training they will need to become proficient on the new or updated system. Some users prefer to read, while others prefer to watch videos. If the system is new, or the changes are significant, consider in-person training or webinars that enable users to ask questions in real time. Schedule training at times that are convenient for users. (Training outside business hours is rarely appreciated by those who have to attend.)
 - *Recurring tips:* Using the same communications channels used for notices and reminders, provide users with occasional tips about system use — even better if the tips come from other users.

When it comes to campaigns, marketers like to think in terms of timelines. What can be done before the event? What can be done during the event? What can be done after the event?

The notices and reminders are ‘pre-event’, while the training and tips happen during and after.

Creating and managing effective campaigns requires staff resources and budget. The more creative the staff resources, the less budget will be required.

The key takeaway from this is that one should always consider the promotional aspect of any new or customised system. If a forthcoming change will affect only a small pool of users, one-on-one communication may well be best. If not, plan accordingly or risk undue resistance, even with perfect technical execution of the change.

SUMMARY

The majority of DAM systems purchased and launched over the years are no longer in use. While the need for content management has increased for most organisations, existing

systems are time and again found to be cumbersome, intrusive or simply not solving the business problem they were intended to address.

There is no Photoshop of DAM, nor is there any ‘winner’ among DAM vendors, as to who leads the industry. Although technology giants Adobe, OpenText, IBM and HP have each launched their own DAM (or DAM-like) systems, none has been able to steer the industry their way. Industry veterans, such as MediaBeacon and Canto, have also failed to take control, despite having provided systems since the early 1990s.

Given the massive increase in content development and use, one would have expected the DAM systems of these companies to have flourished. Instead, the biggest evangelists of DAM continue to be analysts trying to sell research, and marketers trying to sell software. Even in the most successful DAM implementation cases, users seem to have settled for something that simply works well enough.

Some have blamed DAM vendors, saying they lack vision. Others have blamed users, saying their expectations were too high, or that they skimped on training or failed to define and adhere to policy.

The disconnect is more one of marketing. The DAM industry as a whole speaks in terms of ‘solutions’, but DAM software is no more a solution to DAM than a scalpel is a solution to a brain tumour.

Imagine the results of ‘home surgery kits’ or a wizard-based medicine prescription service. In fact, the notion of such products might fascinate many of us: why not just try whatever the wizard suggests for your fainting spells? If it does not work, no harm done — right?

People value the contributions medical professionals provide. These professionals are trusted and people assume they know best. They are expected to see past glossy medication advertisements, and to know the unknowable about our bodies.

DAM software ‘out of the box’ is no different than a scalpel or pill — it cures nothing, and it can easily make things worse if not handled properly. Only in the hands of someone who can diagnose, imagine and implement, can DAM software help cure business problems.

DAM software that is not adapted to the unique requirements of the purchasing organisation will never deliver what users expect or business requirements demand. Unfortunately, DAM vendor sales and marketing departments will rarely admit this because they want to make the purchase process as clean and simple as possible.

Most DAM systems fail for the same reason you would eventually give up on the

home surgery kit — DAM is not a do it yourself (DIY) project.

SEE THE RESULTS OF CUSTOMISATIONS IN USE TODAY

Due to space limitations and confidentiality agreements with customers, Modula4 is unable to include specific case study examples in this paper. However, to see links to recent systems studies from customers like Honda, NASA, Eddie Bauer, Stanford University and others, contact Modula4 at info@modula4.com.

© *Michael Gellner and David Diamond, 2018*