Architectural Content Management in Agile Times

Challenges, Requirements and Solutions Identified in a Large Enterprise

Jörg Ziemann[1]

Abstract: Architectural Content Management is an essential pillar for successful Enterprise Architecture Management. To provide guidance for optimizing architectural content management, we first define and motivate architectural content management. Then we describe requirements and solutions for managing architectural content, including suitable tool chains and ways of structuring content. The content of this article was created in a current, real-life use case inside a large aviation corporation. However, management of architectural content is for the largest part industry-independent, thus, the concepts can also serve as a reference for large enterprises in other industries.

Keywords: Enterprise Architecture Management, Knowledge Management, Architectural Content Management, IT Content Management

1 Architectural Content Management and its Challenges

Creating and maintaining information on IT systems is generally a challenging and unpopular task, which is why examples of unsuccessful IT content management are easy to find. For example, outdated specifications on important, complex applications render these very hard to manage. Obviously, it is in the interest of many stakeholders to have a certain amount of specification and documentation on digital systems available. That applies, for example, to product owners, architects, new team members and application management. For these stakeholders, of course, "the code is the documentation" does not suffice.

An agile paradigm, on the other hand, implies lean documentation and highlights collaborative aspects in IT teams, and it has also been discussed intensively in the context of Enterprise Architecture Management (cf. for example [Br18]). However, in addition to classic means, like code versioning systems and MS Word documents resting on file shares, more lightweight media types have become popular in the last several years. Rather than striving for perfection in form and content, these methods support a more collaborative, decentral and on-the-fly generation of content.

Architectural content management is a subcategory of IT content management, focusing on the creation, communication and maintenance of architectural content. *Architectural content* comprises all artifacts that describe the current or the to-be state of the enterprise

[1] Lufthansa, Lufthansa Aviation Center, Airportring, 60546 Frankfurt/Main, Germany, joerg.ziemann@dlh.de

cc by sa doi:10.18420/inf2019_ws03

architecture, for example, guidelines or reference architectures. *Enterprise Architecture* is the basic structure of an enterprise information system, including its elements and their relationships as well as relationships to the environment. Finally, *Enterprise Architecture Management* (EAM) consists of planning, steering and controlling the enterprise architecture. EAM is applied not only in the aviation industry (cf. [Su16]) but in large enterprises of all kinds of industries.

To be successful, EAM must be in permanent contact with the organizational stakeholders, e.g. business owners, solution architects, projects leads and developers. Optimally, the whole organization understands the enterprise architecture and the stakeholders are closely integrated in the creation and maintenance of architectural content. Unfortunately, architectural content management in practice is difficult as well; and literature on this topic is sparse (cf. [Fa07], [Cl11], [St12]). This is why examples of unsuccessful architectural content management are easy to find. For example, a large enterprise, where many groups are addressing a new technology, but lacking transparency – who is doing exactly what, based on what principles and technologies. Another example is the detailed wiki site on, for example, enterprise-wide integration architecture that is outdated and hence ignored by the organization. Other typical problems are:

- No single point of entry and truth. For example, when content is spread over many different tools which are used differently by different departments.
- Not enough access rights. It is technically cumbersome to provide access to stakeholders and conceptually hard to find the right balance between making content accessible to everybody while producing stakeholder-tailored content.
- The binding character and actuality of the content is unclear, as well as the processes and responsibilities for content maintenance.

As a result, a lot of architectural content is lost, created redundantly or ignored. For the stakeholders, it is difficult to find, access and maintain architectural knowledge.

2 Requirements for Architectural Content Management

2.1 General Requirements and Principles for Enterprise Architecture Content

The proverb “The shoemaker's son always goes barefoot” applies also to IT professionals. However, an ad hoc, unstructured approach for managing enterprise wide architectural data does not scale for large corporations. Hence, the same principles that apply to business data, must also apply to IT internal data management, this includes:

- Single source of truth: If various, disconnected departments address the same content (e.g. rules for integration architecture) this results not only in redundant work but also in confused stakeholders.

- Content management follows organizational structures and responsibilities: If the structure of content management follows organizational structures, the risk of overlapping focus areas is lower and content maintenance is motivated intrinsically. In any case, a responsible owner must be assigned to every content field.
- Easy to maintain, lean and up-to-date: Architecture content is limited to the necessary amount. Otherwise, it will be hard to maintain and keep up-to-date.

In addition, the following principles are typical for architectural content management:

- Widespread: To be in touch with the overall organization, most architecture content (e.g. design guidelines, architectural decisions) should be widely accessible. The tools should be endowed with corresponding licenses and access possibilities.
- Usable: Though for certain content types specialized architects have to coach stakeholders on using the content, in general architectural content has to be both easy to find and easy to understand.
- Collaborative: The success of EAM relies on the close integration of stakeholders and collaborative tools. This includes, for example, the early involvement of stakeholders in content creation and constant feedback loops.
- Decentral and hierarchical: Architecture content relies on decentral creation to integrate knowledge from many technical areas and from different local requirements. Distributing and subdividing architectural topics also leads to lower complexity.

Note that these requirements only represent a generic reference, i.e. every organization has to assess and formulate its own requirements for its specific enterprise architecture.

2.2 Heterogeneous Characteristics of Architectural Content

From the perspective of an IT project, typical content types are solution sketches, architecture descriptions, requirement specifications, technical specifications, source code, test specifications, operating concepts, organizational information (e.g. stakeholders or team calendars) and processual information (e.g. a Kanban board). Enterprise architecture focuses on longer-living material with a broader scope, e.g. architectural guidelines, best practices, reference architectures, decisions, standards, application catalogues, organizational information (e.g. boards and stakeholders) and processual information (e.g. architecture processes and case management). These content types can have different characteristics, for example:

- Level of maturity, e.g. is the content just a first draft, or a polished, final version?
- Level of bindingness and formality, e.g. is it a binding rule or only a recommendation?
- Rate of change, e.g. is the content highly dynamic or rather static?

- Classification level, e.g. is the content private or public?
- Architecture domain or subject, e.g. is the content on data or integration architecture?
- Vertical depth/level of detail, e.g. is it an abstract rule or a detailed specification?
- Horizontal breadth, e.g. does the content address the whole enterprise (enterprise architecture), a business domain (domain architecture) or one application (solution architecture)?

3 Defining Solutions for Architectural Content Management

3.1 Tools for Architectural Content Management

To address the significantly different characteristics of content types (see list above), a selection of content management tools has to be used. Figure 1 provides an overview of a tool chain for IT content management, with a focus on enterprise architecture content. Naturally, depending on the focus area, different tools can be used, e.g. specialized tools for requirements and test management, source code and artefact management or a CMDB.

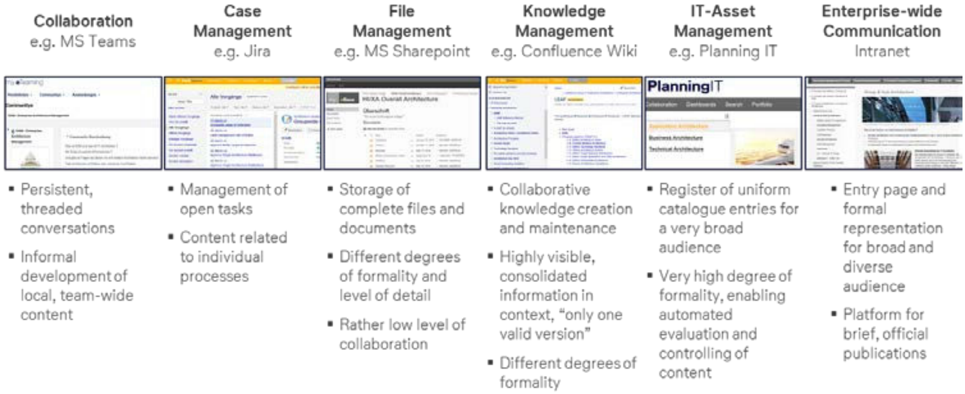
Collaboration
e.g. MS Teams
Case Management
e.g. Jira
File Management
e.g. MS Sharepoint
Knowledge Management
e.g. Confluence Wiki
IT-Asset Management
e.g. Planning IT
Enterprise-wide Communication
Intranet
PlanningIT
▪ Persistent, threaded conversations
▪ Informal development of local, team-wide content
▪ Management of open tasks
▪ Content related to individual processes
▪ Storage of complete files and documents
▪ Different degrees of formality and level of detail
▪ Rather low level of collaboration
▪ Collaborative knowledge creation and maintenance
▪ Highly visible, consolidated information in context, "only one valid version"
▪ Different degrees of formality
▪ Register of uniform catalogue entries for a very broad audience
▪ Very high degree of formality, enabling automated evaluation and controlling of content
▪ Entry page and formal representation for broad and diverse audience
▪ Platform for brief, official publications

Fig. 1: Classes of Architectural Content Management

The main classes and typical characteristics of architectural content management tools are described above. Of course, many of these tools can be used in a broader scope and for multiple purposes. However, their exact usage should be defined inside an enterprise, to avoid redundancies and the inefficient use of effort and spreading of content. A recent example of this can be illustrated with Microsoft Teams. Teams offers team-specific, and thus subject-specific, content collections in the form of threaded conversations. Searching is possible inside as well as across teams and Teams offers team-specific SharePoints and wikis. However, the integrated wiki is very basic and not as powerful as a dedicated wiki, e.g. from Confluence. Thus, instead of using the wiki of MS Teams, in the context of architectural knowledge management it makes more sense to have a link to a Team (inside MS Teams) and a link to a Confluence wiki site.

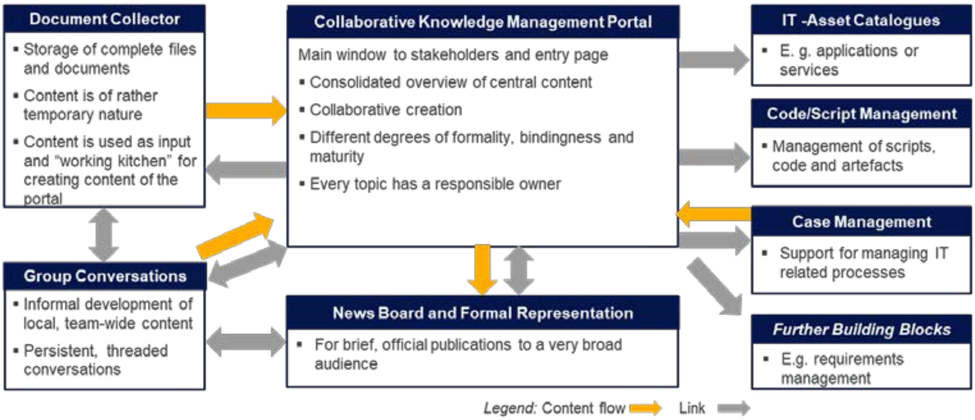
Document Collector
▪ Storage of complete files and documents
▪ Content is of rather temporary nature
▪ Content is used as input and "working kitchen" for creating content of the portal
Collaborative Knowledge Management Portal
Main window to stakeholders and entry page
▪ Consolidated overview of central content
▪ Collaborative creation
▪ Different degrees of formality, bindingness and maturity
▪ Every topic has a responsible owner
IT -Asset Catalogues
▪ E. g. applications or services
Code/Script Management
▪ Management of scripts, code and artefacts
Case Management
▪ Support for managing IT related processes
Group Conversations
▪ Informal development of local, team-wide content
▪ Persistent, threaded conversations
News Board and Formal Representation
▪ For brief, official publications to a very broad audience
Further Building Blocks
▪ E.g. requirements management
Legend: Content flow
Link

Fig. 2: Connected, Logical Building Blocks for Architectural Content Management

Going beyond a classification, Fig. 2 shows how the logical building blocks of architectural content management are connected into a system. Here, a wiki ("Collaborative Knowledge Management Portal") is used as the central portal for creating and communicating content, surrounded by tools for case management, file management ("Document Collector"), a collaboration tool for team conversations, a catalogue and the intranet for rather formal publications to a broad audience.

3.2 Organizational and Logical Structures for Content Management

If the tool chain has been defined, the next question is how to structure the content inside of each tool. Here, the following rules of thumb apply:

As mentioned above, content management must "go with the flow" and follow organizational structures and needs. If an organizational unit is responsible for a topic, identifies itself with it and has incentives for that content to be up-to-date and known to stakeholders, this raises the chances for successful content management. Nevertheless, if the topic is important enough for the organization, content management can also be successful in temporary or less formal structures (i.e. communities) across organizational units. However, *roles* for content management should be in place, e.g. owners for every topic. In addition, *processes* should be in place, e.g. for evaluating and refactoring the content. Here also, automated means are available, for example, to check the filling degree of a catalogue or the actuality of catalogue entries. Moreover, incentives for writing or refactoring content can be established and, e.g. in wikis, metadata can be used to tag outdated content.

Moreover, some architectural content types can be managed centrally or de-centrally. For example, for architecture guidelines, an enterprise-wide, central repository can be used. Such a central approach generally increases transparency, comparability and controllability of content quality. On the other hand, different organizational units responsible for these subjects, e.g. departments for API management, Process Digitalization and Data management, could also maintain such guidelines de-centrally. A decentral approach with proximity to local experts, increases the feeling of responsibility and identification with their content. Thus, for the different content types (e.g. catalogue entries, design guidelines, reference architectures, literature), it must be decided if a central or decentral approach is to be followed.

Therefore, to enable a broad support, architectural content management should have strong bonds with local experts, be organic rather than forced, have 80% rather than 100% maturity, and be concise rather than lengthy. In this context, an important requirement is that enough licenses and access rights are in place; encountering “access denied” on a wiki-page leads to demotivated knowledge workers. Thus, to enable widespread collaboration, access rights for architectural knowledge should be provided generously.

4 Conclusion

Architectural content management is a highly important yet challenging and thus heavily discussed topic. It is not a “one size fits all” situation, but, according to requirements related to the content type, the corresponding tools must be chosen. In addition, the structure of content, its logical allocation and further parameters have to be defined, e.g. an effective distribution of central/decentral responsibilities for content management and roles and processes. Current technologies, like collaboration tools that more organically feed into the knowledge creation process, make a sustainable IT content management easier. However, going beyond technology, IT content management is just the tip of the iceberg of the overall digital enterprise, whose core business it is to create and maintain content that shapes digital systems. I.e. the success of knowledge management is also dependent on core company parameters like the willingness to share, to be transparent and to create content collaboratively. However, in an age where major vendors also provide open source software, the general direction towards openness and knowledge sharing should be obvious.

Literature

[Br18] Brockmann, C., Sultanow, E. & Czarnecki, C.: Enterprise Architectures between agility and traditional methodologies. In: Czarnecki, C., Brockmann, C., Sultanow, E., Koschmider, A., Selzer, A. (Eds.): Workshops der INFORMATIK 2018 - Architekturen, Prozesse, Sicherheit und Nachhaltigkeit. Bonn: Köllen Druck+Verlag GmbH (p. 15), 2018.

[Cl11] Clerc, V., Lago, P., Vliet, H. v.: Architectural Knowledge Management Practices in Agile Global Software Development. In: 2011 IEEE Sixth International Conference on Global Software Engineering Workshop, Helsinki, pp. 1-8, 2011.

[Fa07] Farenhorst, R., Lago, P., van Vliet, H.: EAGLE: Effective Tool support for sharing architectural knowledge. In: International Journal of Cooperative Information Systems, Vol. 16, No. 03n04, Special Issue Software Architecture — Towards the Software Engineering Core, pp. 413-437, 2007.

[St12] Stewart, D.: Solution Path for Optimal ECM in Cloud, On-Premises and Hybrid Environments. Gartner, 2016.

[Su16] Sultanow, E., Brockmann, C., Schroeder, K., Breithaupt, C.: Lufthansa aviation standard: developing an open group reference architecture for the aviation industry. In: Mayr, H. C. & Pinzger, M. (Eds.), Informatik 2016. Bonn: Gesellschaft für Informatik e.V. (p. 825-836), 2016.