

Scaling by Optimising: Modularisation of Data Curation Services in Growing Organisations

Hagen Peukert
University of Hamburg

Abstract

After a century of theorising and applying management practices, we are in the middle of entering a new stage in management science: digital management. The management of digital data submerges in traditional functions of management and, at the same time, continues to recreate viable solutions and conceptualisations in its established fields, e.g. research data management. Yet, one can observe bilateral synergies and mutual enrichment of traditional and data management practices in all fields. The paper at hand addresses a case in point, in which new and old management practices amalgamate to meet a steadily, in part characterised by leaps and bounds, increasing demand of data curation services in academic institutions. The idea of modularisation, as known from software engineering, is applied to data curation workflows so that economies of scale and scope can be used. While scaling refers to both management science and data science, optimising is understood in the traditional managerial sense, that is, with respect to the cost function. By means of a situation analysis describing how data curation services were applied from one department to the entire institution and an analysis of the factors of influence, a method of modularisation is outlined that converges to an optimal state of curation workflows.

Submitted 29 January 2020 ~ Revision received 20 January 2021 ~ Accepted 20 January 2021

Correspondence should be addressed to Hagen Peukert, Monetastraße 4, Email: hagen.peukert@uni-hamburg.de

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: <http://www.ijdc.net/>

Copyright rests with the authors. This work is released under a Creative Commons Attribution Licence, version 4.0. For details please see <https://creativecommons.org/licenses/by/4.0/>



Introduction

The beginning of the 20th century witnessed a most noticeable reorganization in the production workflows of goods. Even if not often recognized in the field of pioneering management theory, it was Charles Babbage who was the first to put down on paper quite specific thoughts on the specialization of work processes together with quite modern ideas of profit sharing dependent on the company's output (Babbage, 1832). Yet, it needed more than a century of experience and theorizing before Babbage's ideas were taken up again (Duncan, 1989). Around this time, Henry Towne, coming from an engineering background, made explicit that a scientific inquiry into production workflows and worker's skills is most beneficial (Towne, 1886). It was just then that Frederick Taylor formulated his version of scientific management that was really rather a consequent enhancement of Towne's theoretic frame (Taylor, 1985), however, differing in the way employees were looked upon, i.e. as soldiers. Taylor's foremost goal was to diminish the soldiering problem, which describes employee's behaviour to work less than they could. Its successful application by Ford and the advent of mass production seem to be the alleged proof of the correctness of Taylor's conception and his name would coin several generations of thinking in management theory.

Based on Taylor's scientific management, the years to come revealed a developmental path of management science going through stages like bureaucratic and administrative management featuring the functional approach by Henri Fayol or Chester Barnard, turning just then back to behavioural science (certainly initiated by the Hawthorne studies showing the power of motivation to production output) with prominent figures as Hugo Münsterberg (industrial psychology) and Mary Parker Follett (group dynamics in organizations), Abraham Maslow and Douglas McGregor, and finally arriving at quantitative approaches to management, in which mathematical models and statistical methods are used to find optimal working organizations, as well as applying *Systems Theory* and *Contingency Theory* to management studies resulting partly in re-engineering attempts and lean management at the dawning of the *new economy* towards the end of the 20th century.

From this admittedly short survey, it is still unclear how the present phase, characterised by the new development stage of digitalization, relates to the evolution of management studies in the last century. However, *Industry 4.0* is the catch phrase referring to the digitalization process in production of goods whereas the counterpart of digital services is *Web x.0* reflecting the needs prevalent in a digital economy. By observing these developments, what can be inferred in today's management task is a radical restructuring of all organizational functions due to various kinds of digital technology. The classic distinction between services and goods is blurring more and more. And along with these fundamental changes the tasks of management also change rather radically. Digitalization involves digital data in the first place and consequently digital data management. Thus, managerial tasks shift towards new forms of (data) management in the course of the digital revolution.

Entrepreneurial action in the digital age means rethinking traditional managerial practices. At the same time, digitalization is a chance for all data stewards to embellish data management in the way that is most beneficial to the users of data, i.e. us or, in a wider sense, society as a whole. In other words, the vision of shaping appropriate data management is not supposed to parallel the evolution of classical management studies, in which the more appropriate idea by Babbage was interspersed by Taylorism and which needed about a century to resurrect behavioural approaches focusing on the human being, its traits, and its motivation with respect to organizational goals. As we know today, Babbage's approach is by far the more effective way of managing organizations (Moldaschl & Weber, 1998; Zapf & Semmer, 2004; Conway & Briner, 2005; De Cuypter et al., 2008; Frone, 2003).

The idea behind *Open Data* (Open Knowledge Foundation, 2021) is one of the most valuable tenets of research carried out in the digitalization age. At the same time it seems to convey a

direction for conceptualizing the managerial action involved in data management. *Open* suggests both chance and key to advance modern scientific thinking backed by a larger (and well-informed) research community. The term *Open* furthermore underlines the general principle of accessibility to data regardless of the institution, financial resources, origin or status of the user, and technical barriers. It involves a high degree of transparency and more equal opportunities in most research communities and plays a catalytic role in scientific enhancement by admitting diverse thinking and participation.

It is in this spirit in which data management is understood here when specifying the optimization problem and a possible solution in the case of the research data management center at the University of Hamburg. Indeed, some methods of the traditional inventory from general management must not be designed from scratch for the analysis at hand. They can be applied to the case of data curation and other data services where appropriate. What is meant more specifically by the spirit of devising data management practices constrained by the tenets of *Open Data* is the renouncement of any competitive advantage based on proprietary data, which parallels Taylor's retrogressive thought of looking at workers as soldiers, that is, to be treated as a kind of property by the company owner.

Scaling versus Optimising

Scaling has different meanings in data science and management science although the general idea is the same. In management science, scaling refers to the effects found if larger quantities can be produced because the fixed cost decrease per unit. It is a central role in operations management to keep those costs, that are independent from unit output, i.e. fixed costs, as low as possible (Porter, 2008; Reading, 2002, pp. 160). In fact, it is the primary source of savings. In data management *scaling* basically means if larger quantities of data can be used by the same application (scale out) and infrastructure (scale up), i.e. if retrieval and database technology can handle huge numbers of data within an acceptable time frame.

As it is applied in the definitions of scaling, *optimizing* suggests finding the best quantity either of – as in the case of operations management – the production output – or as in the case of data management – the data to be stored and retrieved. It is also possible to emanate from a given target of data or output respectively and optimize the (production) processes, that is for the case of data management, which application and infrastructure best serve the needs of a given amount of data. Yet, it is fair to assume that the amount of data is hard to estimate in advance so that data machines are designed to the maximum of their technical limit (scale out). This is especially a preferred solution if more recent technological advances, such as object storage, allow for dynamically allocating storage to the needs of a specific data management service.

The two scaling conceptions in management science and data science involve two different kinds of savings. While the first can rather be attributed to the allocation of human resources, i.e. homogeneous tasks and processes enable data curators to perform faster at better quality, the second, i.e. savings, met by a well-scaling application or infrastructure, are related to the cost of technical devices and maintenance.

Although technological advancements will keep on reducing the scaling problem in data management, the effect gained through optimization of technical resources can still be substantial and should be seen as a chance by data managers. By the same token, the allocation of human resources to well-arranged tasks unfolds to several multiples (Muchinsky, 2006; Burke, 2005, pp. 18). However, there is no simple unilateral relation between scaling and savings, that is, the more scaling the higher the savings, does not apply. Now, this is what optimization alludes to. It is about finding out at which scaling factor the overall savings is highest. It may involve several variables as will be disclosed in the section on *Factors of Influence* below.

To illustrate the optimization potential for the economies of scale of curation tasks, one should recall that scaling effects can be found if tasks become more homogeneous (and standardized). This is exploited in mass production. From this literature, however, it becomes

also clear that a task too homogeneous will counter the effect because it becomes tiring to repeat a dreary job over and over again (Morrison, Cordery, Girardi & Payne, 2005). Here, all symptoms of demotivation come into play and these prevent optimal execution. So optimizing in this context means, to specify a task in such a way that it neither under challenges nor overburdens the staff (Tuckman, 1965). Equally important, high standardization of tasks runs counter to the flexibility and adjustability of permanently changing requirements in the work to be done. So, to find the right measure in the composition and allocation of tasks is subject of optimization (Albrecht, 1979).

Scaling problems at the level of data infrastructure and repository applications occur if the scaling factor of a repository is too low or high respectively. While scaling too high does not justify the cost of the infrastructure, a scaling factor too low leads to abandoning the service. Put differently, many low-scaling repositories could still be a good alternative to one high-scaling repository if maintenance and overall costs are weighed against each other. Again, optimization involves the best possible relation between amount of data, type of data, access time, and cost of infrastructure.

In sum, *scaling* effects are key in both data and management science, and in both managers strive to use it in an optimal way. Yet, optimizing refers to the fact that maximum scaling may not pay off at the end, so that scaling should be achieved as a net effect in due proportion to its costs, be it the cost of infrastructure or human resources. The main task is, therefore, to solve the scaling problem, both from a traditional managerial perspective as well as from a data management perspective.

Modularisation as a Basis of Optimisation

In the previous section, it is implicit that the two kinds of scaling effects relate to each other at the intersection of human resources, i.e. when the cost of maintenance of the infrastructure services and repositories are considered, then usually, staff is involved for maintenance, which also includes human tasks that can be well or badly arranged and allocated.¹ It is these economies of scale that are primarily considered here since the arrangement of tasks and workflows and its allocation to staff is most relevant in the daily business of sustainable data curation management, however, it has been largely disregarded so far in the discussion on optimizing data management workflows.

To understand what is meant by modularization, the concepts of task, submodule, and module will be introduced first. It is illustrative to think of a hierarchy whereas the task is the smallest unit, followed by submodule – a container for tasks – and module – yet another container for submodules. The specific *task* definition differs from organization to organization and even if the outcome of the task is the same, it could involve different task definitions in different organizations. Tasks evolve by established practices in the curation work. Although such natural evolvement of tasks does not imply that the established tasks are ideal (optimal) constellations, it is still a good starting point for workflow and task analyses. To identify tasks, a data manager needs to carefully observe the practices carried out by data curators in an organizational unit and, given an equal problem space to solve, managers should note down which procedures are (nearly always) carried out together, that is, at about the same time and in the same order. These *conglomerates* of operations occurring together should be documented as a task. In analogy to the organization of a software program, the task corresponds to a *routine*, a *function*, or a *method* respectively. Modules could be related to the notion of *packages* and submodules to *classes*, which are ideally *loosely coupled* parallel to what is done at the submodule level in the given approach (Hagel & Brown, 2005).

Sets of tasks are called submodules, forming a natural unit on the basis of efficiency to carry out these jobs or by some other salient classification criteria. In other words, *submodules* are

¹ See e.g. Barsky, Laliberté, Leahey and Trimble (2017,p.92) for the Canadian case

building blocks featuring the smallest set of tasks that are appropriate for goal attainment in relevant subfields in data curation. It is important to note that, in contrast to tasks, submodules can be loosely re-arranged although they may depend on each other as regards content (but not time).

Modules, then, should be constructed from submodules in such a way that they are maximally independent whereas it is totally clear that absolute independence is not realistic as envisioned in the example below. It is rather a question of how much independence is possible without claiming that absolute independence be feasible. Modules are composed with respect to the needs of a specific curation project. Since not every curation project needs the exact same jobs to be carried out, modules ensure the much-needed flexibility. To illustrate, a curation project might already have got a well-established data model, but just needs an implementation and a redesign of the web interface. When combining modules to a work package for this curation project, independency of modules ensures that a possible module consisting of data model development does not entangle with the implementation module. The project would then just need a definition of a new module by flexibly combining e.g. submodules like *data model implementation*, *front end design*, *front end implementation*, and optionally *data consultancy*. A standardization approach lacking the right granularity of its tasks into submodules, misses out on the specific needs of a data project. In the worst scenario from the above example, the data curator would develop a new data model since they are unable to keep the modelling tasks apart from the model implementation and front-end implementation workflows since most of the (unidentified) tasks are intertwined in such a way that at least some processes cannot be isolated from others, leading to delay and needless work. Thus, standardization in the sense of one size fits all, would not allow for composing a new workflow ensuring that it functions well.

Depending on the intended specificity, it is, of course, possible to define additional levels in the hierarchy, such as subtasks or units larger than the module, but considering the size of most research data centres in public institutions at the time being,² two to three hierarchical layers are sufficient. Really, the division into layers is a method, in colloquial terms a “trick”, to better understand the processes prevalent in the organization. At the end, it merely serves to find the best possible compromise between standardization and adaptability of tasks, i.e. efficiency and dependency for the curation task on average and in perpetuity. It remains up to the reader to parallel this method to recent approaches in software development and to reflect why these approaches continue to be successful.

Now, modularization involves two steps. First, it refers to the process of figuring out from a rather unsystematic bunch of tasks, the number, composition, and content of the smallest possible set of submodules appropriate to the organizational goals, and second, the optimal combination of these submodules to modules that are most suitable to the median of data curation projects in the organization, i.e., the most frequent projects to be carried out.

The knowledge what is the appropriate size of a module comes from practices in the organization, i.e., the actual curation work established over years in the institution, where it is applied. So the composition of tasks in a submodule can differ from one organization to another and cannot be generally stated. To illustrate, one module at the Centre of Data Management in Hamburg is modelling, delivering a complete data model specification of a given project. It comprises three submodules: data field definition, attribute definition, and data relation definition. It is clear that the respective implementation module depends on the specification with regard to its content and time of application. Still, it makes sense to treat them as independent modules because, in the organization in question, there are a growing amount of curation projects that have a standard data model available. Yet another module is data consultancy, which fulfils the dependency constraint although at times it could draw on the results of the module data analysis. As a conclusion, independencies of modules are seldom absolute, but follow the minimization principle.

From the above example it is clear that the formation of modules probably differs substantially in each organization. Highly specialized data centres may choose a finer

² For the Germanic region, see <https://dhd-ag-datenzentren-dienstekatalog.github.io/profil-der-datenzentren/>

granularity of modules than organizations offering more breadth in their services. The organization of tasks into modules, that can be flexibly arranged, is the basis for optimizing the workflows. A real optimization gain results from the neat and subtle arrangement of these modules. So the question of what neat and subtle actually means has to be answered. The most exact response one can give in absence of the specifics of the organization is to choose the smallest possible number of modules to fulfil all requirements of a given curation project. If the minimization principle of the used modules is met, then is optimization as well. To be clear, the potential of optimization lies in the correct compilation of modules and configuration of submodules from individual tasks. This is where the analytic work has to be done but once. The routine managerial work, then, is to select the modules to a specific curation project. Most of the time, the selection is self-explanatory and directly results from the preliminary analysis of the project.

Factors of Influence

Based on five years of experience at the Center of Research Data Management, six impacting factors on optimization were identified, that is, parameters that significantly influence the task definitions and configuration of submodules. Here, they are put in order for their relevance to the data centre at Hamburg.

- the level of heterogeneity and consistency of the research data
- the technical and human resources available
- the requirements and functionality of custom applications
- the technological knowledge of the staff
- the quantity of service requests, and
- the requirements and expectations of faculty members.

Looking at this listing the reader might figure that some parameters also impact on each other. The degree of heterogeneity of the data and the number of service requests may trade off, e.g. a high number of service request and curating consistent data projects harmonise while curating more heterogeneous data would not. Hence, in the case of available human resources and their technological knowledge, fewer staff have to possess more detailed technological knowledge (Lengnick-Hall & Lengnick-Hall, 2005).

In what follows, these factors of influence will be briefly introduced. The order from above is changed for reasons of clarity. First, the level of heterogeneity and consistency of the research data is the main reason to deviate from complete standardisation and consider the appropriateness of modularisation. Generally, one can say that the degree of modularisation parallels the degree of data heterogeneity. The same is true for data consistency. Very consistent data, i.e. data in total compliance with the established standard, needs no adjustment. Inconsistent data, on the other hand, needs an adjustment for each deviation from the standard. Hence, a different, less complex, workflow can be chosen for the case of curating consistent data. As an illustration, data from the humanities is widely considered to be more heterogeneous and less consistent than data from other disciplines of science, especially natural sciences (Sahle & Kronenwett, 2013). Taking an increasing proportion of data from the natural sciences into account might cause less difficulties and thus a low level of heterogeneity.

Second, the quantity of service requests marks a critical mass at which it pays off to introduce a modularisation strategy. If all service requests can be executed in due time, it might not even be necessary to take the burden of change and initiate modularisation. But if not, modularisation should be considered as an option. To get a concrete idea of the contribution of

the net effect of a modularisation strategy, it is advisable to calculate some figures before and after the implementation. To compute a cost function, it is necessary to have measures of the time and effort of previous projects and the costs of the infrastructure. If not available, estimates could be used in the form of the salary of the involved staff and the base price of the server infrastructure. Normalised by a time scale (month or year) and plotted against the number of projects for both, before and after modularisation, gives the net effect as the difference of both functions on the ordinate.

Third, the technical and human resources available directly impacts launching modularisation. As discussed above, a high scaling technical infrastructure takes some of the burden of maintenance and customising data applications. A sufficient number of data curators, however, often leads to avoiding change, i.e. efficiency analysis and workflow redesign. Even if enough personnel are available and the need for change is not pressured by the amount of work, data managers could still launch modularisation and compensate time gains for additional services or, better, for keeping up with recent technologies in a continuously learning approach (Tuckman, 1965).

This directly depends on, fourth, the technological knowledge of the staff. As human resources are limited, so is the scope of technological knowledge of the staff. Not every existing or emerging technology used in custom built applications can be dealt with. In addition, knowledge and experience crucially impact the processing time even of minor adjustments and even the more so for fundamental changes. So human resources have to be allocated as a function of processing time, importance of the project (e.g. security issue vs. *nice to have*) and availability of staff. Data curators disposing of deep knowledge of various technologies are shooting stars since they are flexibly suitable for most curation jobs and need substantially less time to finish it. Unfortunately, breadth and depth of technological knowledge are the exception because, based on the time the employee has to invest, depth and breadth of a topic trade off.

Fifth, the requirements and functionality of custom-built applications are likely to be most influential and most difficult to estimate (Albrecht, 1979). Generally, curating legacy systems (custom built database applications, virtual research environments, or the like) for long term sustainability gets more challenging the further one moves away from *standard* applications or repository systems.

Last, faculty members have different levels of knowledge of what they exactly need. The communication process is less effective if a customer has only a vague idea of what the application has to do. Sometimes it comes along with a high expectation of solving problems that are not really related to the application (e.g. categories to be searched or how to present the data). It is the ineffective communication that cost additional time to figure out what is needed or to correct and redesign a first prototype.

A summary of how modularisation of modules and impacting factors are composed into a new project workflow is given in figure 1.

Situation Analysis

Independent from the management approach an organisation likes to follow in managing its data, the general requirements, that is which data services are needed or must change, should be clear. So, in what follows, the situation as it is at departmental level and at the entire institution will be given as a concrete example with respect to the factors of influence identified above. The situation analysis implicitly reveals the requirements in data management and curation services.

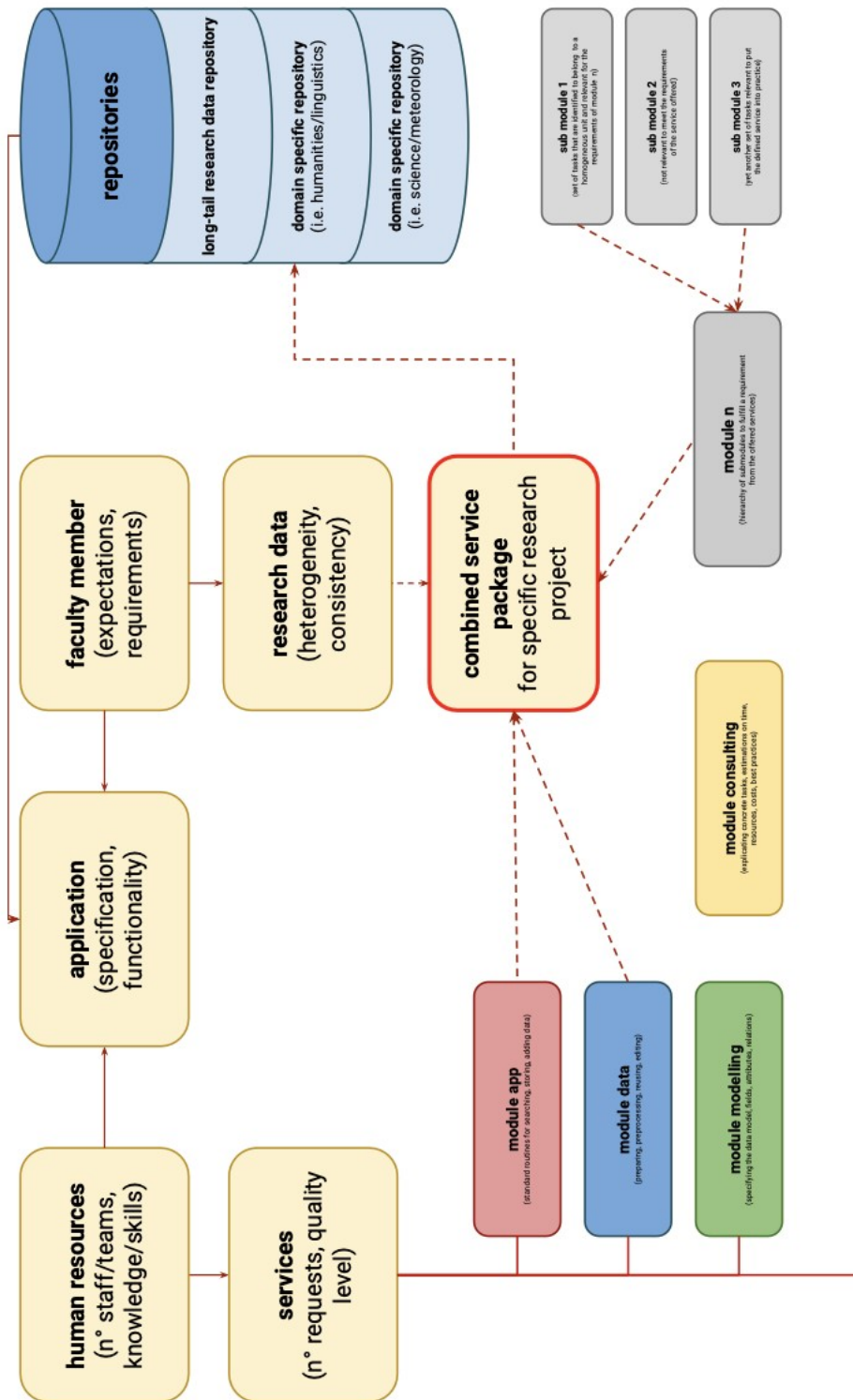


Figure 1. Impact Factors and Modularization in Data Process Optimization

Data Services at the Humanities Department

Offerings of data services at the Humanities department adapted, by and large, the research data life cycle (plan/design, collect/capture, analyse, manage/preserve, share/publish, discover/reuse) whereas collecting data was restricted to consultancy in the planning and designing process. Any form of general archiving was also not covered. Yet, due to a high demand, the largest share of work is done in designing customised repository applications for data retrieval and formatting data. So managing active data and data repositories is the main focus of the curation work to be carried out. In addition, data consultancy becomes more and more important for each of the stages in the data life cycle. Especially with respect to third party funding proposals, consultancy during the planning phase is important.

It soon became clear that communication to the research staff was particularly crucial. Research staff's knowledge differs tremendously. The distribution of data knowledge ranges from professional to no knowledge at all with a rather steep skewing towards the latter. It became apparent that it is a wise investment to spend more time to find out what is really needed (as opposed to wanted) and communicate directly what is possible to readjust expectations on what the data service can actually comprise. In cases where data awareness is high and previous experience exists less, problems in the communication occurs.

The heterogeneity of humanities data is characteristic. This is in part due to the variety of disciplines subsumed under this heading, that is humanities. The inventory of research methods produces data formats that are incompatible with a common standard. Even within a discipline, little agreement is reached towards schemas that best describe the data. Together with low awareness and, in part, the reluctance to accept the relevance of specific data knowledge for one's own field of research, carrying out effective data services becomes a challenge for maintenance services and communication issues.

In sum, the humanities department is a perfect model case because it reveals the entire spectrum of the challenges of data management at all stages of the data life cycle. First, all data types and formats are represented. Second, the communication and expectation problem between researcher and data scientist can be experienced in a variety of ways and prepares well for the cases to come. And third, the dynamics in the estimation of time and resources in planning and implemented data services as well as flexible workflow design and analytics can be put into more concrete alternatives of taking action.

Data Services at the Entire Institution

The Universität Hamburg is a classic representative of top-down developed data services. It engages the services of a computing center. In general, these services comprise components of the technical infrastructure such as providing network drives, virtual machines, web servers, or data base servers. More specifically, designed services for the handling of data curation are not available. They remain at the department level. It is common practice to leave data curation at the level of the chairs or even at the single researcher.

It follows that an unmanageable diversity of single solutions exist, some of which do not meet the minimal standards of good programming or data practice respectively. In absence of practical data sharing solutions and despite the institutions data policy, proprietary software and web services by third parties are used for the simple reason of usability. These, however, bear the risk of exploiting sensitive data stored there for the purpose of business and profit strategies.

Some of the University's departments have their own data centers, namely Medicine, Chemistry, Earth Sciences, and Physics. Their services meet a highly specialised demand for e.g. DNA data and applications, satellite data, or the particle accelerator and collider data at the Physics department, whose mass of data needs a totally different data strategy and infrastructure. Data services there are tailored to the specific needs of the discipline, which a general research data center could not provide.

It is important to note that neither a general data repository nor domain specific repositories for archiving and long-term storage exist. In addition, consulting adjusted to the needs of the disciplines is also not offered. The Document Management remains a vague offering without an explicit portfolio of concrete services that could actually be drawn on. Assignment of persistent identifiers, such as the digital object identifier (DOI), never made it beyond a stage of testing into real world applicability and accessibility as demanded by the researchers.

As a conclusion, one can see that data management services are not implemented globally at the university as an overall and comprehensive strategy. Apart from the basic components of the technical infrastructure, there are no reliable research data management or curation services offered. The many individual solutions feedback unproductively since a mass of virtual machines have to be technically maintained, from which little information on what they are used for is available.

Devising a Modular System of Data Curation Services

Because of the data heterogeneity, the Humanities are like a perfect sample of the data demands prevalent in the entire university. Hence a portfolio of services will be based on the use case collected in the Humanities. The array of services will now be modularised and optimised in accordance with both a quantitative estimation of the data service demands and the available resources, as suggested in the introductory chapters.

The established data centres at the departments of Medicine, Physics, Chemistry, and Earth Sciences will continue to provide their specialised services. Indeed, collaborations with these centres exist. Modularisation of our service portfolio enables us to integrate a subset of research data services that are attractive, but unavailable to the staff of these departments, i.e. the research data centre's services complement the established services. For example, the open access publication of dissertations or master thesis can be done in the publication server of the general repository. So the modularisation strategy is keen to avoid redundancy of data services.

The following modules, submodules, and tasks have been created since the inception of the centre. As work and experience with the different departments continue, the offerings will likely be adjusted and complemented. This initial start is not static, but modularisation also allows for flexibility. So far we have defined eight modules: Planning, Consulting, Training, Dissemination, Storing, Modelling, Application, and Data. As a sample case, planning will be described in more detail at the task level. Thus, the principle of modularisation – the interplay of tasks, submodule, and modules – becomes tangible. For the sake of simplicity, suffice it to describe the remaining modules at the level of submodules since the single task level is too manifold and specific to be fully described in this survey, e.g. a task description of the module *application curation* would be too fine-meshed and differs widely from institution to institution.

Module Planning

The planning module consist of four submodules: cost calculation, time and resource estimation, task definition, and DMP (Data Management Plan). The DMP-submodule aims at explicating the requirements of data management planning as formal criteria. It is not about writing up the plan, which remains the responsibility of the researcher. The tasks in this submodule are designed to communicate what goes in a data management plan and how best to go about receiving this information. So it is rather a matter of taste to integrate the DMP-submodule into the planning module. It could as well be involved in a module, e.g. *consulting*.

The DMP submodule can be broken down into seven tasks (see Table 1). As part of the planning module, DMP tasks involve the communication with the client and some analytics in the first place. A good start is to find out what kind of data, which formats and size, is collected. A follow-up task usually includes the question of how to describe the data, i.e. are there

metadata standards available or could the implementation of a new schema be advised. Oftentimes this task incorporates some research into the respective discipline's best data practices. A further task embeds a possible access strategy. In analogy to the previous task, data policies will be communicated if known, otherwise the ingredients for one's own access policy will be devised, so that the researcher is able to write it up in the DMP. Closely related to the question of access are the conditions under which data can be re-used. This also has to be made explicit. In the present case, it made sense to outsource the answer to this question as an extra task. And it is important to keep subsequent data re-use apart from the question how the data should be accessed during the process of data collection. Here, another application or platform is often used or extra software is programmed. From the above, one is able to estimate the future storage demands and the budget. Both, storage and budget estimations, are allocated to separate tasks.

Usually most of the planning tasks can be communicated in a single meeting with the researcher, in which the researcher describes their project and the data curator ask more specific questions to find out which exact file formats are needed or if access policies are known. In some exceptional cases a second meeting is necessary either because the researcher, our customer, does not know the answer to the questions or we, as the service provider, need to look into the issue under investigation, e.g. the data policy of very sensitive data.

Although not being a requirement, the availability of a DMP makes it easier to carry out all the remaining tasks in the planning module. Yet, the DMP is not a requirement as such because oftentimes a DMP is already available or, if not, the information needed can be inquired directly for either time, cost, resources, or tasks separately. Within the submodule *task definition*, the most basic information on what to do is collected. It is a starting point for further estimations.

Time and resources depend mutually on each other as discussed in the introductory chapter. Therefore, the two corresponding submodules that could also be treated separately were merged into one to avoid task overlap. All estimation and calculation is treated in the submodule *task definition*. From there the task load can be evaluated. From the task load the right combination between time and resources can be estimated. Finally, cost calculation addresses the different scenarios of expenditures: alternative costs, outsourcing options, actual and target performance, and variance analysis.

For the same reason as explained above, it is a good idea to start with the submodule *task definition*. It involves four tasks: *collect to-dos*, *describe as tasks*, *arrange tasks*, *choose modules and submodules*. Collecting *to-dos* happens in a prolonged talk with the customer who is asked to give a detailed exposition of what is needed. Dependent on the chosen service, application building or data handling, the curator has to elaborate on the usually very general descriptions, and thus the research center's staff filters the relevant information and notes down the *to-dos*. In a second task, these *to-dos* will be described and transformed into manageable tasks that are known from the submodules of the collected portfolio of similar jobs. These tasks are in a third step, re-arranged into a workflow by considering similar requests of past projects. It is, then, possible as a last task, to map these arranged tasks to the submodules and design the relevant module composition.

The submodule *time and resources estimation* takes the task definition as a prerequisite. The arranged set of tasks mapped to the workflow, i.e. to modules and submodules, makes it feasible to carry out a first task, which is to estimate the workload (Boehm et al., 2000). It is common in project management to use a neutral figure as *person months* here despite the fact that it considers the job qualification only indirectly. It is important to keep in mind that this task involves a projection in the future, that is, how many staff are available for a year and how much time would a task need based on the past experience. The variation by unforeseen events, such as sickness or labor outage, can be included by a percentage. An empirical value often used in project management is to plan a workload of 60% (PMI, 2013; Wischniewski, 2002). The second task in this submodule seeks to gain insight into human resources that are planned to be vacant. The resources are allocated to the task. Depending on the qualification of the staff, the length of the time slots is adjusted in a third task. And lastly, the different resource time combinations (scenarios) are calculated whereas the optimisation criterion is time minimisation.

Person months can be directly transferred into cost of human resources. Yet, this gives the wrong impression of what costs really are. First, person months give only a vague idea of the actual cost since they consider an average standard performance, such as a very general qualification level and salary. Second, in an ideal world, the different levels of qualification expressed in higher salary measures and the time, in which a corresponding task is carried out, pay off linearly in person months. Actually this is not the case. Hence, the submodule *cost calculation* accounts for a more thorough analysis of the cost of human resources, but it also includes tasks to calculate opportunity costs, the cost of outsourcing options, the actual and target performance, and a variance analysis. All these numbers are important for an elaborated planning of the project and they are also crucial for sophisticated consulting.

Modules Consulting, Dissemination, and Training

Consulting is closely interconnected with planning. So most of the time the consulting module is needed, the planning module is also applied. However, planning does not fall back to consulting. Hence, to keep the two modules separate is the correct choice from an organisational perspective. Oftentimes it is advisable to integrate e.g. the DMP-submodule or the cost-submodule into the consulting module if for some reason the planning module cannot be carried out completely. As set up here the consulting modules comprises seven submodules: *DMP explication*, *DMP checking*, *data strategy*, *task evaluation*, *service specification*, *requirement analysis*, and *SWOT analysis*.

The DMP-explication submodule aims at clarifying what a data management plan is to a customer. So the task focuses on the adjusted definitions of DMPs and also lists a steadily growing base of model cases that are specific to the project under investigation. Moreover, chances and limitations of a DMP are given, again, with regard to the given project. A second submodule is defined for checking a written DMP, embracing tasks such as reading and marking, best practice comparisons, or specification of experts that can give further assistance.

For good consulting services, it is paramount to understand the requirements and goals of the project in depth. So this submodule comprises a task named *understand project*, behind which a semi-standardised survey of questions has to be answered by the customer. These questions are different from, i.e. more specific than, the information determined in the DMP. It is about the needs of the target group and how to satisfy them with regard to the goal of the project by giving a precise service specification. Based on that, other tasks are meant to take the target group into account: *define target specification*, *find best practice*, *compare to best practice*, and jobs aiming at adjusting the gap between best and current practice.

The submodule *data strategy* elaborates on properties of the users of the data as well. In addition, it prepares the specifics of the *access policy* and how to *make the data anonymous*. Evaluation is part of solid consulting services. Here, it is the submodule *task evaluation* that considers the quality of the jobs to be carried out. It contains task packages as a comparison with use cases and feedback from customers and target groups (users of the data or the application). In some cases, a SWOT (strengths, weaknesses, opportunities, threads) analysis is asked for. Hence, we set up a submodule that handles the four areas in extra *find-out-and-specify*-tasks.

The *dissemination* module satisfies the need for knowledge transfer and to a certain extent marketing measures, such as placement of the center's services and communication of its benefits. All submodules included here correspond to the traditional channels of scientific knowledge sharing as *conference papers* or *conference attendance*, but also the writing of news feeds and interview preparation. Lastly, the module *training* includes submodules that define tasks required for teaching a class and holding workshops. They range from *material preparation*, *class organisation*, and *class implementation*.

Modules Modelling, Application Curation, Data Curation, and Storing

This set of modules is designed to tackle the actual curation work. As already mentioned above, the task definitions are plentiful and their description would probably comprise a couple of

pages each, which are unlikely to be of good use for other organisations. However, it is likely that their containers, the submodules, have the potential to be similar in other institutions.

The module *Modelling* makes do with three submodules, provided that the analysis is done. These submodules are *data field definition*, *attribute definition*, and *data relation definition*. All three submodules can be applied to both designing a new data model from scratch given a specification or a DMP and to the analysis of an existing model that possibly needs to be transferred or changed. These three submodules reflect the three components in each data model description: the field, the attributes and the relations among them. Another possibility to design a *Modelling* module is to analyse the data in terms of classes of complexity or types of data.

Table 1. Sample of curation services as modules, submodules, and tasks

Module	Submodules	Tasks
Planning	DMP	find out metadata
		find out data format
		find out data size
		find out access policies
		find out data reuse
		find out data storage
		find out application for data collection
		find out budget
	task definition	collect ToDos
		describe as tasks
	arrange tasks	
	choose modules/submodules	
	time and resource estimation	estimate task load
		allocate resources
		allocate time slots
		make scenarios
	cost calculation	alternative costs
		outsourcing options
		actual and target performance
		variance analysis
Consulting	data strategy	define data target group
		define anonymous data
		define access policy
	DMP explication	give DMP definitions
		define chances, limitations
		give model cases
	DMP checking	...

Dissemination

The module *Application Curation* links closely to the *Modelling* module. The two modules work hand in hand if a new application that needs persistent data is set up. Again, it makes sense to keep them separate because in some cases only application curation is needed but not modelling. Keeping these two modules apart reflects the tenet to keep the program logic separate from the data. Yet, a third module on the view layer has lost its module status and was merged into *Application Curation*. This is because this service is no longer part of the center's core competency. A corresponding submodule to the outer appearance of a web page involves minor

changes on the general university's layout policy determined by a different department. The focus of the submodule *view* is to implement the given layout to the programmed applications.

Other submodules in *Application Curation* are based on the functionalities that a typical research data software should possess: searching, storing, and ingesting; whereas storing is not meant in the sense of archiving, but for developing logic to load and save different forms of persistent data. The submodule's task descriptions are inspired by the software development cycle. Which development philosophy one should follow depends on the knowledge and working culture of the staff. At the FDM center, a form of agile programming has emerged as the apt method on which to build our development process for curating existing applications.

The *Data Curation* module is also closely linked to *Modelling*. It may happen that data is incomplete, broke or that data have to be transferred to a different model. For these cases the *Data Curation* module is designed. It consists of *data editing*, *data preparation*, *data reuse*, *data pre-processing*, *data processing*, *DOI*, and *metadata*. As can be implied from the submodule titles, the focus of the module is to expunge errors in a given data project, validate the syntax and semantics of both content and meta data description. *Data reuse* refers to meta data that needs to be made accessible if, e.g. the data became inaccessible for technical reasons, or for the case of data loss, it involves a work package that attempts to reconstruct data if at all possible. The *DOI* submodule deals with the allocation and assignment of persistent identifiers.

The *Storing* module is currently the least developed in terms of the submodules and task definitions. Yet, it is planned to join the core competencies, application and data curation, in the future. As of now, an object store with an initial size of six petabytes is acquired. This storage will be extended on a yearly basis dependent on the actual demand of previous years. The data repositories are connected to the storage device by a S3 interface mainly used for data less than 50 gigabytes of size. Data of larger quantities are handled with a different process. It is for this process that the storage module needed to be defined. So far the module consists of three submodules: *quality assurance*, *data upload*, and *data-screening* (by periodic checksum mapping).

The submodule *quality assurance* reviews the metadata by checking for the syntax (well-formedness of data) and semantics (data validity). Usually the quality criteria are based on the DMP. If not available, the task of finding the appropriate metadata standards from the *planning* module has to be integrated in this submodule. A possible deviance in the metadata schemes between the standard and the data to be archived is a major quality criterion. As such it is weighted highly. Further, the content of the data itself is not checked. However, indicators such as data consistency, grant size, project reputation, access frequency of the data, or simply the consultancy with us are monitored to evaluate the status of data and the probability of its data reuse. The shortcomings of this approach are well understood, yet, a more workable solution is not available. It is not always possible to keep all produced research data for more than the promised ten years. Research data low in the quality ranking will not be kept in the long run, i.e. more than ten years.

Data exceeding 50 gigabytes are not supposed to use the web interface for uploading. For these cases, data upload will not work automatically in the repository software, but after contacting the center's service staff. They will provide an alternative upstreaming process or carry the upload out ourselves manually. Before uploading, quality checks in the metadata and general consistency of data is done.

The submodule *data screening* checks the usability of the data; whether the data is still readable with the application that was originally designed for its use or another software with equal functionality. The time intervals, in which these checks are carried out depend on the data format, the application that is needed to access the data and the quality weight. The data screening is planned ahead and resources are allocated accordingly. Furthermore, scalability checks are carried out.

The process descriptions in this chapter is supposed to give some insights into the optimisation principle implemented in the research data center. Even if the details of each task could not be fully elaborated on, the general mechanism is similar to putting together software components without having to understand the level of methods. A rough documentation of *what* (as opposed to *how*) the methods do should be sufficient.

Discussion

In the previous sections, the modularisation of organisational tasks is explicated. It is also stated that this kind of modularising is reminiscent of the procedure in software engineering. In fact, recent advances in software engineering are the primary source of inspiration of specifying the presented method. The design of sustainable software architectures (e.g. Lilienthal, 2017) as well as the analysis and the transformation of non-durable software into sustainable software programs retraces an important management technique in digital management. Looking at the history, in which designing large software programs has developed over the last 50 years (Royce, 1970; Parnase & Clements, 1986; Boehm, 1986), may teach data curators an important lesson in managing digital data. Really, analysing intransparent software programs parallels the work that needs to be done in organisational processes, more specifically in data curation workflows. So, it is plausible to test and see if the techniques and solutions in software engineering would also hold in managing digital data.

The two most crucial issues to consider here are, first, the difference between organisations and software programs in aspects that are relevant for workflow design, and, second, the general applicability of the approach to a wide range of data curation centres differing in their culture, philosophy, and prospects. As far as the former is concerned, it is clear that generally organisations as *living* entities work differently from software programs. It makes little sense to work out the many differences. Instead – for the purpose of this paper – it is wiser to look at the tasks of an organisation, i.e. its organisational processes (Grasl, Rohr & Grasl, 2004). At the process dimension, there are indeed differences between software programs and organisations. Importantly, within organisations people are at work and so psychological parameters like motivation or in-group behaviour are at play. Due to psychological parameters, the prediction of the work outcome differs from software, which is most valued for its reliability. Both, software and organisational processes, could be described by means of algorithms. Yet, organisational algorithm's outcome are more fuzzy. This means that the outcome's variance is larger or less predictable in organisations; hence less reliable.

Closing this gap is an idealisation from reality and will not be completely achieved. What matters is if the degree of deviance is small enough, so that a process analysis and process optimisation still make sense. In other words, the algorithm a program performs is like an idealised organisational process. Consequently, one could think of the organisational tasks in terms of tasks that a software procedure has to carry out even if the way, in which the task is done, differs. Abstracting to the organisation's tasks and leave the human factor out is not to say that human factors are irrelevant, but helps the manager to focus on an idealised version of task performance. The underlying logic is that the organisational processes combined to entities that are most productive in an ideal world, will also be most productive in the real world, but at lower rates given that the task definitions consider the psychological constraints of the staff, e.g. a task should be challenging but not over-strain the employee (Meijman & Mulder, 1998; Marks, 1977; Rohmert, 1984)

One could now raise the argument that Taylor's way of looking at workers as soldiers forced into standardised workflows is paralleled by viewing them as robots prescribing what needs to be done by algorithms. To avoid this confusion, it is possible to think of two independent domains, i.e. a domain of designing optimal data curation processes and a domain of how the staff uses them as well as which responsibilities and freedom are given to the staff by modularisation. While the former is the main theme of this paper, admittedly the latter falls short in the above analysis. Since it is another wide field worth of investigating in follow-ups, suffice it to mention here that modularisation has a lot to commend to motivating human resources. More specifically, modularisation as proposed, delegates more responsibility to the data curator by incentivising to compose new workflows out of the repertoire of submodules and tasks that can be flexibly rearranged as required by a specific data assignment. Indeed, this is part of the job description in organisations introducing modularisation. It is no longer necessary, and maybe not even possible without additional layers of management, to implement modularisation as a rigid form of standardisation in a Taylorian understanding. Modularisation per se presupposes a

high degree of flexibility, but also responsibility to the staff. Responsibility is inbuilt. A lack of self-dependent staff counteracts a healthy working atmosphere in the respective institutions (Sekigushi, 2004; Semmer, Zapf & Greif, 1996; Semmer, 2003). Further, modularisation allows a team of data curators to assign submodules of tasks to each other according to their abilities or preferences (West & Borrill, 2006). Thus, modularisation embodies quite the opposite of Taylor's idea.

Keeping the process perspective apart from the personnel perspective bears another advantage. Organisational processes themselves are more easily separated from psychological variables that are hard to predict and distract from the optimisation objective. By abstracting away from the human factor, organisational processes become algorithms that are in all relevant features similar to the algorithms a software program carries out and, thus, an upper performance limit can be set. To stay in this analogy, software that has undergone many uncoordinated changes and new functionalities in the course of time often ends up in a badly written program that is hard to maintain; it is probably error prone. To refactor *spaghetti*-code into a well-organised robust piece of software that can be easily extended in the long run is broadly speaking the same thing as reorganising the intertwined processes of a naturally grown institution. As in the case of refactored software, the organisational processes of an institution become more efficient and very likely converge to an optimum once restructured by modularisation. This optimisation process seems to be as general for institutions as it is for software programs. By scaling, i.e. repeating, the modularisation principle to the many different workflows of an organisation, we find that scaling leads to more optimisation as a side effect.

But how well do the proposed modules scale in general? This is the second issue to be discussed here. As of now there cannot be an exact answer beyond some general thoughts and a proposal for a solution in the future. This question concerns the circumstance that every organisation is different and whether the degree of difference makes the approach suggested in this paper vulnerable, i.e. subject to substantial change or non-applicability. So the question is how scaleable are the identified modules and impacting factors in organisations with different organisational cultures and stakeholder interests.

To begin with, there are corporate cultures in small and medium-sized businesses that deny efficiency and explicit performance measures. In fact, staff in those organisations show openly disrespect and resistance towards any form of direct managerial intervention. Although these business models work for niches, in the long run they will face the pressure of societal and technological advances resulting in profit orientation as a necessary condition to survive once their organisation will grow. This is not a particular problem of establishing a management method tailored to the demands of digitalisation, but a problem of admitting diversity in an organisation's corporate culture. Any other approach of traditional management would fail, too. Digital management will not solve issues unrelated to management and change in general. So, the focus of the argument brought forward here cannot be concerned with these cases.

Organisations clearly dedicated to constant change, performance and open to managerial action certainly facilitate the introduction of modularisation. Even if restricted to the small subdomain of research data centers they may still show a tremendous degree of diversity in their business processes and cultural values. Again along the lines of the software analogy a possible solution can be outlined here for future work to find out how well the proposed modules and impacting factors scale to other research data centers. This is yet another case of how digital management infuses with traditional management practices, since software programs may be equally diverse as organisations. It could be revealing to see how software engineers – the process managers of software programs – deal with a similar problem. One solution that emerged in software development over several years is design patterns. A design pattern suggest an algorithm to a set of problems that are assumed to be solved equally. Software design patterns evolve from programming (best) practices bottom-up. Really, it is about the structure of the process leading to the best solution. Applying the idea of design patterns to organisations means the identification of tasks and its arrangement to larger building blocks according to the best practice within their particular business environment. So, the work of the data manager would be to recognise in which fields and processes his or her organisation differs and pick a

design pattern that best suits to it. As in software development, organisational design patterns should not violate the principle of loose coupling of the submodules because they guarantee the flexibility needed to optimise the workflow. If data managers observed the identification of similar modules and impacting factors as suggested here, the community had a good indicator for high scalability. Thereafter it would make sense to invest in more fine-grained studies to define the exact scaling measures. Until then, it would be advantageous to establish communities of practice enabling data managers to share the patterns they consider effective.

Surely, there are more limitations of applying the suggested digital management technique to classic organisational management as could be considered here. At the end, the approach at hand is one of many building blocks in digital curation management that still has to stand the proof of time. Parallel to how software design patterns emerge from the implicit agreement of many programmers who, independently from each other, arrive at the same solution, it also will be the practices of digital curation managers who may come to similar results in their management and process design of digital curation workflows.

Summary

Based on a short survey of the history of management science, it is argued that the era of digitalisation marks a turning point in management methods. Management of digital resources will become more and more important since it percolates private life and business processes alike. There is hardly any organisation that may abstain from some form of digital management. As a consequence, new management approaches, much more appropriate to dealing with digital data and workflows, enter the field of traditional management. In the design of workflows in digital curation management, the integration of new management approaches is developed further resulting in scaling effects if workflows are optimised in the way software is.

In the paper at hand, the idea of modularisation as practiced in writing sustainable software is conveyed to organisational management, that is, to the design of workflows in the research data center of the University of Hamburg. By paralleling the organisational processes (described here as *tasks*) to the functions, routines, or methods in software development, the problem of process optimisation is addressed. As it is the case in software development, composing packages, modules, submodules, and methods is guided consequently by the optimisation principle. This is necessary to make software sustainable, but also more readable and transparent; in short a software program is easy to understand and adjustments to new trends in handling data are more quickly done. As a consequence, saving effects for both time needed to work through a software and resources/staff employed evolve naturally. These savings are scaling effects as understood in traditional management science, yet not in data management. Scaling, and more particularly *scaling up*, in the reading of data management, however, outlines the underlying idea of the present optimisation approach, that is, how to make data curation workflows of an organisation, firstly, explicit, and secondly, optimal. In addition, scaling in the understanding of both, data management and economics, occurs when applying the modularisation approach to as many organisational processes as possible. Thus, scaling (data management) is needed to arrive at an optimal state of all involved workflows and modularisation itself is driven by scaling effects (economics). For this reason, it becomes obvious that scaling by optimising is particularly important in rapidly growing organisations with little budget flexibility. To make this idea tangible in the practical work of data curators, the implementation of the data curation services at the university's research data center is described.

Starting with a situation analysis, the demand of data services is derived from previous curation projects in the humanities department. The techniques of task identification and allocation to modules and submodules is addressed. And a sample description for a selection of modules and their inter-relation is explicated. Lastly, a brief discussion hints at the analogy to software development and elaborates on two possible limitations.

References

- Albrecht, A. J. (1979). *Measuring application development productivity*. Proceedings of IBM Applications Development Symposium.
- Babbage, C. (1832). *On the economy of machinery and manufactures*. London: Charles Knight.
- Barsky, E., Laliberté, L., Leahey, A. & Trimble, L. (2017). Collaborative research data curation services. In L. R. Johnston (Ed.), *Curating research data* (p. 79-101).
- Boehm, B. W. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11, 14-24.
- Boehm, B. W., Clark, Horowitz, Brown, Reifer, Chulani, . . . Steece, B. (2000). *Software cost estimation with cocomo ii with cdrom* (1st ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Burke, R. J. (2005). Human resources as a competitive advantage. In R. J. Burke & C. L. Cooper (Eds.), *Reinventing human resource management: Challenges and new directions* (p. 17-33). London: Routledge.
- Conway, N. & Briner, R. (2005). *Understanding psychological contracts at work. a critical evaluation of theory and research*. Oxford: Oxford University Press.
- De Cuypter, N., De Jong, J., De Witte, H., Isaksson, K., Rigotti, T. & Schalk, R. (2008). Literature review of theory and research on the psychological impact of temporary employment: Towards a conceptual model. *International Journal of Management Reviews*, 10, 25-51.
- Duncan, J. (1989). *Great ideas in management*. Jossey-Bass: San Francisco.
- Frone, M. (2003). Work-family balance. In J. Quick & L. Tetrick (Eds.), *Handbook of occupational health psychology* (p. 43-162). Washington, DC: American Psychological Association.
- Grasl, O., Rohr, J. & Grasl, T. (2004). *Prozessorientiertes Projektmanagement*. München: Carl Hanser Verlag.
- Hagel, J. & Brown, J. (2005). *The only sustainable edge: Why business strategy depends on productive friction and dynamic specialization*. Harvard: Harvard University Press.
- Lengnick-Hall, M. L. & Lengnick-Hall, C. A. (2005). The hr function in the new economy. In C. L. C. Ronald J. Burke (Ed.), *Reinventing human resource management: Challenges and new directions* (p. 35-52). London: Routledge.
- Lilienthal, C. (2017). *Langlebige Software-Architekturen: Technische Schulden analysieren, begrenzen und abbauen*. dpunkt.verlag: Heidelberg.
- Marks, S. (1977). Multiple roles and role strain: Some notes on human energy, time and commitment. *American Sociological Review*, 42, 921-936.
- Meijman, T. & Mulder, G. (1998). Psychological aspects of workload. In P. Drenth, H. Thierry & C. De Wolff (Eds.), *Handbook of work and organizational psychology* (p. 5-33). Hove: Erlbaum.

- Moldaschl, M. & Weber, W. (1998). The "three waves" of industrial group work: Historical reflections on current research on group work. *Human Relations*, *51*, 347-388.
- Morrison, D., Cordery, J., Girardi, A. & Payne, R. (2005). Job design, opportunities for skill utilization and intrinsic job satisfaction. *European Journal of Work and Organizational Psychology*, *14*, 59-79.
- Muchinsky, P. (2006). *Psychology applied to work. an introduction to industrial and organizational psychology*. Belmont: Wadsworth.
- Open Knowledge Foundation. (2021). *Open Definition. Defining Open in Open Data, Open Content and Open Knowledge*. Retrieved 15/01/2021, from [https:// opendefinition.org/od/2.1/en/](https://opendefinition.org/od/2.1/en/)
- Parnase, D. & Clements, P. (1986). A rational design process: How and why to fake it. *IEEE Transaction on Software Engineering*, *12*, 251-257.
- PMI. (2013). *A guide to the project management body of knowledge*. Newtown Square: Author.
- Porter, M. E. (2008). *On competition*. Harvard Business Review Book: Boston. Reading, C. (2002). *Strategic business planning: A dynamic system for improving performance & competitive advantage*. Kogan Page: London.
- Rohmert, W. (1984). Das Belastungs-Beanspruchungskonzept. *Zeitschrift für Arbeitswissenschaft*, *38*, 193-200.
- Royce, W. (1970). *Managing the development of large scale systems*. IEEE WESCON.
- Sahle, P. & Kronenwett, S. (2013). Jenseits der Daten. *LIBREAS. Library Ideas 23: Forschungsdaten. Metadaten. Noch mehr Daten. Forschungsdatenmanagement*, 76–96.
[doi:http://dx.doi.org/10.18452/9043](http://dx.doi.org/10.18452/9043)
- Sekigushi, T. (2004). Toward a dynamic perspective of person-environment-fit. *Osaka Keidai Ronshu*, *55*, 177-190.
- Semmer, N. (2003). Individual differences, work, stress and health. In M. Schabracq, J. Winnubst & C. Cooper (Eds.), *The handbook of work and health psychology* (p. 83-120). Chichester: Wiley.
- Semmer, N., Zapf, D. & Greif, S. (1996). 'Shared job strain': A new approach for assessing the validity of job stress measurements. *Journal of Occupational and Organizational Psychology*, *69*, 293-310.
- Taylor, F. W. (1985). *The principles of scientific management*. Easton: Hive.
- Towne, H. R. (1886). The engineer as an economist. *Transactions of the American Society of Mechanical Engineers*, *7*, 428-432.
- Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological Bulletin*, *63*, 384-399.
- West, M. & Borrill, C. (2006). The influence of team working. In J. Cox, J. King, A. Hutchinson & A. MacAcoy (Eds.), *Understanding doctor's performance* (p. 106-122). Oxford: Radcliffe.
- Wischniewski, E. (2002). *Aktives Projektmanagement für den IT-Bereich*. Vieweg: Wiesbaden.

Zapf, D. & Semmer, N. (2004). Streß und Gesundheit in Organisationen. In H. Schuler (Ed.), *Organisationspsychologie – Grundlagen und Personalpsychologie* (Vol. 3, p. 1007-1112). Göttingen: Hofgreffe.