

HEARING THE EMPLOYEE: AN AGILE-READY FRAMEWORK FOR THE SELECTION OF APPLICATION SYSTEMS – PRACTICAL GUIDE

1 Framework for the Selection of Application Systems

Based on the findings of the previous literature analysis and the expert interviews, the framework for the selection of application systems was created. In order to have a positive influence on the central problem hypotheses already with the basic structure of the new framework, an adaptation of the MIO approach described in the related work was carried out. The aim is to combine the structured working method of classical project management with the communication-oriented approach of systemic and human-centered agile project management (Trepper, 2012). The missing consideration of a changing environment, a continuous improvement process and the needs of the employees in previous software selection processes should thus be overcome. The abstract description of the MIO approach also provided great freedom in the creation of the new artifact. A visual representation of the framework for the selection of application systems can be seen in Figure 6.

The project management process in the MIO approach (Huber and Kuhnt, 2007) was adopted for the framework. Therefore, due to space restrictions, it is not further discussed here.

The software development process of the MIO approach was replaced by a software selection process. The procedure used in the framework represents a summary of the positive aspects of previous software selection processes, enriched by the results of the expert interviews. Furthermore, it tries to eliminate the previously identified problems of previous work.

Starting with the market analysis, the company should first determine the relevant market of the software vendors. As many different sources of information as necessary should be used and future market trends anticipated (Theobald, 2016). As described by Sheth (1973), it is not easy to achieve a holistic overview. By defining knockout criteria, the company can identify the most suitable systems from the mostly heterogeneous software market with a large number of different vendors (Schönberger et al., 2014, pp. 1-6; Böhn and Kraus, 2017, p. 1980). The information gained about all vendors should be collected in the form of vendor profiles, so that market potentials and limits can also be identified. It is then recommended to make initial contact with the vendors in order to provide them with a transparent overview of the general conditions of the software selection.

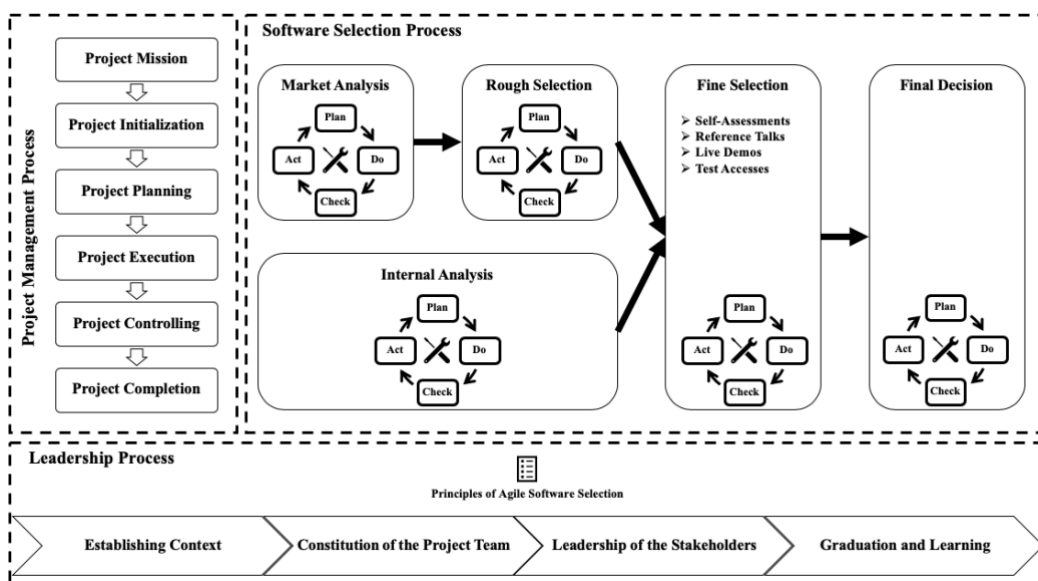


Figure 6. The Framework for the Selection of Application Systems

The market analysis is followed by a rough selection. The software vendors resulting from the market analysis are involved in a tender in which they present their company and disclose as much information as possible about their system. This makes it possible to test not only the quality of the systems but also the skills and motivation of the vendors. In addition, organizational topics such as the signing of a non-disclosure agreement or scheduling for the further process can be discussed with the vendors during this phase. Thereafter the project team has to eliminate unsuitable software vendors from the selection process. With the help of the acquired information, the vendor profiles and the assessment of market potentials and limits can be enriched within the context of continuous improvement.

Parallel to these two phases, the internal analysis of the relevant business procedures will also take place. This includes, on the one hand, the creation of the requirements catalogue. As the literature analysis shows that previous requirement catalogues often lead to communication problems (Hornung and Winkelmann, 2018; Beckmann and Gröschel, 2008) between the different stakeholders, the framework provides a structure to eliminate them. Based on Klüpfel and Mayer (2007), the requirements catalogue is subdivided into the areas of basic information, general conditions, functional coverage, sector solutions and provider services. This structure makes it possible to design the catalogue of requirements flexibly for different application domains, company sizes and other general conditions. Accordingly, the catalogue can be used independently of any general conditions of a software selection process. As part of the requirements catalogue each requirement gets a priority. Like Beckmann and Gröschel (2008) we only decide between *must-have* and *nice-to-have* requirements.

Additionally, each individual requirement also has a clearly defined structure (see right half of Figure 7), based on the findings of Röder et al. (2009). Besides this aspect, general formulation rules should also be used to define the requirements so that communication problems cannot arise. To the solution-neutral and less detailed formulation of requirements proposed by Gronau (2001), care should also be taken that a simple and uniform sentence structure is used. Moreover, it is important to ensure that a requirement can be tested during the selection process. (Grande, 2011, pp. 64-70; Lämmrich and Lorson, 2017) Figure 7 summarises the aspects of the catalogue of requirements mentioned once again graphically.

On the other hand, the analysis, optimization and documentation of the processes affected by the software to be introduced is a central component of the internal analysis too. With the help of the process documentation, a better understanding of the requirements is to be created both internally and externally. Accordingly, they serve as a supplement to the catalogue of requirements. The framework recommends three different types of documentation and gives additional instructions as to in which situations the use of which species is most appropriate.

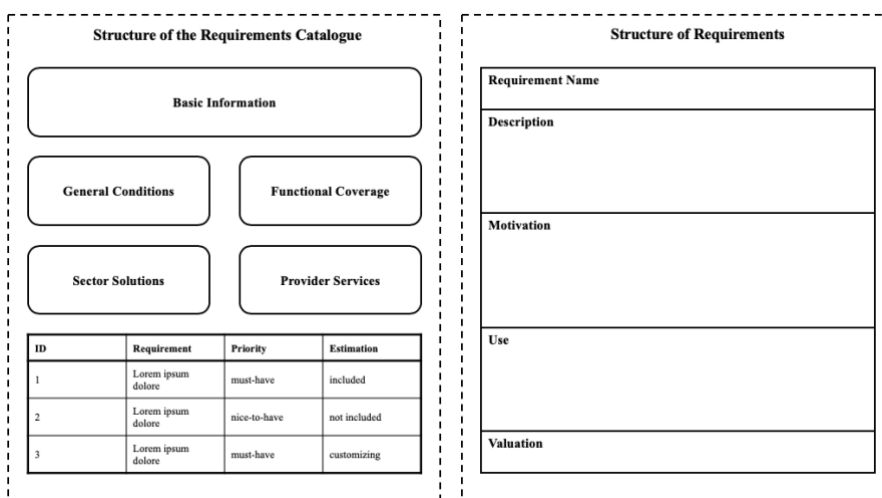


Figure 7. Structure of the requirements catalogue and individual requirements

The use of Hornung's and Winkelmann's (2018) framework is the most complex, but also the most detailed possibility of process documentation. The relevant processes are presented in three different levels of detail, which makes the dependencies between the individual processes easy to understand. Due to the high manufacturing effort, however, the use is only recommended if the company has already carried out a process optimization before the software selection process. Otherwise, there is a risk that the costs of the project will increase considerably and the project duration will be extended.

If there are already modelled processes and they only have to be checked for their optimization potential, the use of classical process models makes sense. Since a heterogeneous group of stakeholders has to be addressed, only modelling methods from the technical concept of the control view of the architecture of integrated information systems (ARIS) house are recommended (Leimeister, 2015, pp. 114-116).

If a company has not yet thought about recording its processes, it is advisable to use so-called process profiles. This is the non-visual summary of the most important information of a process, whereby causal connections between the individual requirements of the future system are recognizable. Since there is no standardized notation for process profiles yet (Gadatsch, 2017, p. 86f.), Figure 8 provides a representation of those developed for the framework.

Process Name	
Process Owner	Process Modeler
Process Goals Why is the process needed? What is the purpose of the process?	
Process Description Technical summary of the contents (what is it about?)	
Process Inputs	Triggers / Input of the Process Which events trigger the process? Which documents, data objects, information, references and orders form the input for this process?
	Predecessors of the Process Which processes provide the input or the triggers?
Process Steps	
Process Results	Results / Output What are the results of the process? Which documents, data objects, information, references and orders form the input for this process?
	Successors of the Process To which processes are the results / output transferred?

Figure 8. Process profile template

After the process documentation and the catalogue of requirements have been created, the internal analysis is completed. By parallelizing these with the market analysis and the rough selection, the framework enables synergy effects to be exploited. On the one hand, the processes used within the standard systems can serve as a thought-provoking stimulus for one's own process optimization (Lämmrich and Lorson, 2017). On the other hand, the knockout criteria for the market analysis can be defined quickly, even if the catalogue of requirements has not yet been finalized. The project team can use these phases to familiarize itself with the relevant subject area and also to continuously improve the quality of the collaboration.

Once the first three phases have been successfully completed, the company must make a fine selection. The remaining software vendors and their systems can be compared using four proposed evaluation methods, adapted from previous work: vendor self-assessment using the requirements catalogue, conducting reference talks, conducting live demos, and testing the software. In this order, the effort required for each method increases, while the level of detail in the evaluation also increases.

In the self-assessment phase, the remaining software vendors receive the catalogue of requirements and the process documentation and must indicate the extent to which their system meets the requirements. This enables the functional scope of the systems in particular to be evaluated. In a joint workshop, the project team then determines which systems are most likely to meet the requirements. During the workshop, the previously used tools must also be enriched again with the newly acquired information in the course of continuous improvement. Self-assessment is recommended in all software selection processes, as the ratio between effort and information gained is quite high. If a company decides against

the other evaluation methods, self-assessment is absolutely necessary. This is due to the fact that the general conditions of the selection process are integrated into the catalogue of requirements. This would otherwise result in a non-consideration of the general conditions.

The holding of reference talks is a good option if the company wants to obtain detailed information about the cooperation with the software vendors. This makes it possible to talk about the introduction and ongoing operation of the software (Nissen and Simon, 2009). In addition, single requirements can be discussed with the reference contact. If a company has not yet had any experience with the selection and introduction of software, it is also a good idea to hold reference talks in order to gain additional knowledge (Nissen and Simon, 2009). Once the company has exchanged all the references, it has to re-optimize the tools it used previously and re-establish a ranking of the systems and their vendors.

The third method in the fine selection process is an on-site presentation by the vendors. In order for these to be as efficient and successful as possible, the company must first define a joint agenda with the vendors and the main content of the presentation. Also to be discussed is the integration of so-called test cases. The vendors have to simulate a given process flow during the presentation (Nissen and Simon, 2009). Using the live demos, usability and user experience criteria can be included in the evaluation for the first time. In this way, employees working with the system in the future can communicate their needs and concerns for reflection in software selection. In addition, the on-site presentation represents the first personal meeting between the vendors and the company. Furthermore, the self-assessments, if obtained, can be checked for their truth content. After considering all criteria and adapting the tools used, the company must again eliminate unsuitable systems from the selection process. If a company decides to use test accesses, the live demos are already an advantageous introduction to the different systems, which can increase efficiency during the tests.

The final possibility for checking the remaining systems is the use of test accesses. This method allows the systems to be evaluated in the most detailed way (Gronau, 2001). In addition to checking the requirements, the vendors' performance can be evaluated again. Moreover, the execution of the tests by the employees enables a detailed evaluation of the usability of the system.

Finally, the remaining systems are evaluated one last time. For this purpose, all collected information is prepared again and discussed in the project team. Due to the structured approach and the continuous improvement of the tools used, it is easy to prepare the final recommendation for the decision makers.

By the free choice of the evaluation methods within the fine selection enterprises have the possibility of defining independently a Trade-Off between efficiency and detailedness of the selection process. The software selection tools included in the framework enable companies to easily adapt, hoping that they will be able to carry out a software selection process independently. Figure 9 summarizes the described software selection process and also gives a recommendation, based on the literature analysis, regarding the effort required for the individual phases and the maximum vendors to be considered.

	Market Analysis	Rough Selection	Internal Analysis	Fine Selection				Final Decision
				Self-Assessment	Reference Calls	Live Demos	Test Accesses	
Percentage of Total Process Effort	20%	5%	25%	10%	10%	10%	15%	5%
Maximum of Remaining Software Vendors	10	7	-	5	4	3	2	1
Tools	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits 	<ul style="list-style-type: none"> • Requirements Catalogue • Process Documentation 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits • Requirements Catalogue • Process Documentation 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits • Requirements Catalogue 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits • Requirements Catalogue • Process Documentation • Test Cases 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits • Requirements Catalogue • Process Documentation • Test Cases 	<ul style="list-style-type: none"> • Manufacturer Profiles • List of Market Potentials and Limits • Requirements Catalogue • Process Documentation • Test Cases
Functions	<ul style="list-style-type: none"> • Induction • First Contact • Preselection 	<ul style="list-style-type: none"> • Induction • Preselection • Organizational To-Do's • Optimization of Tools 	<ul style="list-style-type: none"> • Induction • Process Analysis & Optimization • Requirements Analysis 	<ul style="list-style-type: none"> • Evaluation of Software • Evaluation of Vendor • Optimization of Tools 	<ul style="list-style-type: none"> • Evaluation of Vendor • Optimization of Tools 	<ul style="list-style-type: none"> • Evaluation of Software • Evaluation of Vendor • Familiarization with the Systems • Optimization of Tools 	<ul style="list-style-type: none"> • Evaluation of Software • Evaluation of Vendor • Documentation • Optimization of Tools 	<ul style="list-style-type: none"> • Consolidation of Information • Derivation of Action Recommendation

Figure 9. Summary of the software selection process of the framework

The last part of the developed framework is the leadership process derived from the MIO approach. By taking this process into account, it is possible to establish employee acceptance already during the software selection, so that the risk of later failure is minimized. For this purpose, the abstract recommendations of the MIO approach were adopted and extended by principles of agile software selection, based on the agile manifesto (Cunningham, 2001). Figure 10 shows these principles. The aim is to familiarize the employees with an agile way of working and thus also to consider the three concepts of systemic project management within the selection process. The project staff should be better integrated into the project and the teamwork should improve continuously in the course of the project.

Principles of agile software selection
Changed requirements should always be positively received, even if they show up late in the selection process
All stakeholders work together on the project
Build your project around individual motivated employees. Give them the environment and support they need and trust them to do their job well
The most efficient and effective way to communicate information for and within a project team is through direct communication
The optimal decision for the company is made by a team that organises itself
Building a long-term and successful partnership should be the focus of all interactions with software vendors
Both the project team and the software vendors involved in the selection process should maintain a steady pace forever
The team must consult at regular intervals on how it can work even more effectively and then adapt its behaviour accordingly

Figure 10. Principles of agile software selection