Evaluation for Practices (Form A)

- **Architectural Knowledge**

Practices	Definition	
AK1	Architect leaders provide some tutorials and/or video courses for the technology available	
NFRs	**Reasoning**	**Contribution**
Functional suitability	The existence of tutorials provides information about resources available to help implementing some functionalities	+1
Performance efficiency	The existence of tutorials provides information about resources available to improve the use of this resources	+1
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	The existence of tutorials disseminates security problems and rules of prediction, improving the development levels of authorization and data access	+1
Maintainability	The existence of tutorials guides the development process, avoiding predictable problems and supporting the correct maintenance	+2
Portability	No influence	0

Practices	Definition	
AK2	Build maps of modules and runtime elements during face-to-face meetings	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	No influence	0
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	The existence of maps facilitates to see security breaches	+1
Maintainability	The existence of maps improving the understanding of developer to know the system to do a correct maintenance	+2
Portability	No influence	0

Practices	Definition	
AK3	Create personal blogs and/or wikis to inform about the development and architectural issues	
NFRs	**Reasoning**	**Contribution**
Functional suitability	The existence of blogs and wikis provides information about resources available to help implementing some functionalities	+1
Performance efficiency	The existence of blogs and wikis improves the access to information and provides information about resources available to improve the use of resources	+1
Compatibility	No influence	0

Usability	No influence	0
Reliability	No influence	0
Security	The existence of blogs and wikis provides information about security resources to be used during the development	+1
Maintainability	The existence of blogs and wikis provides information about code and maintenance rules to help improving the development	+2
Portability	No influence	0

Practices	Definition	
AK4	During code review, provide feedback information about architecture, good practices to code, doing refactoring and show the best way to solve the problems	
NFRs	**Reasoning**	**Contribution**
Functional suitability	The existence of architectural information facilitates the development of desired functionalities	+1
Performance efficiency	The existence of architectural information facilitates the best use of the resources available	+1
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	The existence of architectural information provides data about security breaches and rules to be followed to keep and define the correct levels of authorization and data access	+1
Maintainability	The existence of architectural information provides knowledge about design schemes avoiding misunderstanding	+2
Portability	No influence	0

Practices	Definition	
AK5	Document APIs daily	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	The existence of APIs documented can improve the use of resources available to improve the performance of the system	+1
Compatibility	Document APIs provides information about system interfaces that influence the compatibility with other systems.	+1
Usability	No influence	0
Reliability	No influence	0
Security	Document APIs can provide information about security interfaces for levels of authorization and data access	+1
Maintainability	The existence of documented APIs provides information of interfaces of development facilitating the maintenance work	+2
Portability	No influence	0

Practices	Definition	
AK6	Provide mentoring programs to training new developers	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	Mentoring programs inform to new developers existing resources that they can use properly	+1
Compatibility	No influence	0
Usability	Mentoring programs can training new developers how to develop to build the usability properly	+1
Reliability	No influence	0
Security	Mentoring programs can inform how to follow security rules during development	+1
Maintainability	Mentoring programs train developers to follow development rules and standard to keep easy the maintenance of the systems	+2
Portability	No influence	0

Practices	Definition	
AK7	Keep different architects with different levels of specialty to spread the knowledge in the community	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	Specialist architects can concentrate efforts in the better use of the resources to improve the performance	+2
Compatibility	No influence	0
Usability	Specialist architects have experience about how users and they are qualified to spread the knowledge to keep increase the usability	+1
Reliability	Specialist architects have experience with weak system points. So they can make efforts to keep the stability of the system treating the major number of exceptions conditions to guarantee the system working under specified conditions for a specified period of time	+1
Security	Specialist architects know security issues in their area and they can work to solve these problems	+1
Maintainability	Specialist architects concentrate their effort to improve the architectural design and keep the maintainability in their area	+2
Portability	No influence	0

Practices	Definition	
AK8	Provide code recommendations, defining a standard in the community	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	Code recommendations guide developers to use system resources properly	+1
Compatibility	Code recommendations guide developers to respect rules to exchange information with other products	+1
Usability	Code recommendations guide developers to follow	+1

	usability standards to facilitate the user life	
Reliability	No influence	0
Security	Code recommendations guide developers to respect and follow security rules	+1
Maintainability	Code recommendations guide developers to keep the modularity and reduce the impact of changes in the system	+2
Portability	Code recommendations guide developers to follow standards to facilitate the transference of the system to another operational environment	+1

Practices	Definition	
AK9	Provide a development manual for newcomers to know how to start contributing	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	A development manual informs developers resources that can they use to build code to work properly	+1
Compatibility	No influence	0
Usability	A development manual informs developers about standard usability to follow for the new application	+1
Reliability	No influence	0
Security	A development manual informs developers about security rules to build safe code	+1
Maintainability	A development manual guides developers how to modify the code efficiently	+1
Portability	No influence	0

Practices	Definition	
AK10	Keep a register of meetings available to community to know all decisions of the meeting	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	No influence	0
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	A register of meetings informs the community about decisions taken and avoid security problems caused by lack or misunderstanding of development information	+2
Maintainability	A register of meetings communicates development decisions about interactions of system components avoiding problems because the impact of changes	+1
Portability	No influence	0

- **Choice of Technology**

Practices	Definition	
CT1	The organization board defines some technologies that should be used by the whole community as tools for testing, communication, coding review, bugging manager, and navigation	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	The organization board have knowledge and experience enough to recommend technologies to improve the use of resources	+1
Compatibility	No influence	0
Usability	Technology recommendations were discussed in the community to improve the user experience.	+1
Reliability	No influence	0
Security	Technology recommendations can indicate the better tools to follow safe rules	+1
Maintainability	Technology recommendations guide developers to use the proper resource and avoid or reduce the impact of code changes	+1
Portability	No influence	0

Practices	Definition	
CT2	The project leaders define specific technology that impact into the work of the project community	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	The project leaders have knowledge about specific problems of the system and can recommended technologies that can improve the use of resources	+1
Compatibility	No influence	0
Usability	The project leaders can choose technologies more useful for specific needs of the application	+1
Reliability	No influence	0
Security	The project leaders know the security failures and can indicate the better technology to work with these problems	+1
Maintainability	The project leaders know the better tools that can attend the dynamic of development of the specific project	+1
Portability	No influence	0

Practices	Definition	
CT3	The developer chooses tools that impact only in your local work like IDEs and simulators	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	The developer use proper tools to work with the resources available	+1
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	The developer use tools that avoid creating security problems in other parts of the application	+1
Maintainability	The developer choose the best tool to work avoiding or reducing the impact of their changes	+1
Portability	No influence	0

Practices	Definition	
CT4	The organization board recommends a tool to create UML diagrams of the architecture to be used by KDE projects	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	Architectural diagrams facilitates the knowledge of how to use the resources better	+1
Compatibility	No influence	0
Usability	No influence	0
Reliability	No influence	0
Security	No influence	0
Maintainability	Architectural diagrams facilitates the knowledge of how to code with a reduced impact of changes in the system	+2
Portability	No influence	0

- **Quality Management**

Practices	Definition	
QM1	Define minimal quality criteria requirements to add an application to the ecosystem (documentation, automatized tests, dependence restrictions)	
NFRs	**Reasoning**	**Contribution**
Functional suitability	Minimal quality criteria ensure that the application should attend its specification	+1
Performance efficiency	Minimal quality criteria ensure that the application should do a better use of the resources available	+1
Compatibility	Minimal quality criteria ensure that the application should work with other applications into the ecosystem	+1
Usability	No influence	0
Reliability	Minimal quality criteria ensure that the application should work well under specific conditions for a period of time	+1
Security	Minimal quality criteria ensure that the application should follow and respect the security rules	+1
Maintainability	Minimal quality criteria ensure that the application should not impact negatively in other parts of the ecosystem	+1
Portability	Minimal quality criteria ensure that the application should work well with different operational environments	+1

Practices	Definition	
QM2	Projects of the applications define quality criteria in accordance with the organization board recommendations	
NFRs	**Reasoning**	**Contribution**
Functional suitability	Quality criteria allow tracing the requirements to guarantee the correct development of functionalities	+1
Performance efficiency	No influence	0
Compatibility	Core board recommendations ensure that the application should work with other applications into the ecosystem	+1
Usability	No influence	0
Reliability	No influence	0
Security	Core board recommendations ensure that the application should follow and respect the security rules	+1
Maintainability	Core board recommendations ensure that the application should not impact negatively in other parts of the ecosystem	+1
Portability	No influence	0

Practices	Definition	
QM3	Define a team to test performance and behavior of the application	
NFRs	**Reasoning**	**Contribution**
Functional suitability	A test team should discover if an application is working well into the ecosystem	+2
Performance efficiency	A test team should test if the application are using the resources properly	+1
Compatibility	A test team should discover if an application is working well with different products into the ecosystem	+1
Usability	A test team should discover if an application provide a level of usability for the final user	+1
Reliability	A test team should discover if an application is working well under specified conditions for a specified period of time	+1
Security	No influence	0
Maintainability	No influence	0
Portability	A test team should discover if an application is working well in different operational environment	+1

Practices	Definition	
QM4	Use some tools to compute some quality metrics	
NFRs	**Reasoning**	**Contribution**
Functional suitability	No influence	0
Performance efficiency	Quality metrics can help to verify if an application are using the resources available properly	+1
Compatibility	No influence	0
Usability	No influence	0
Reliability	Quality metrics can help to verify if an application are working well under specific conditions	+1
Security	No influence	0
Maintainability	Quality metrics can help to verify if an application can be modified with a low impact into the ecosystem	+2
Portability	No influence	0