

Questionnaire - KDE Architects

Dear Architect

The goal of this questionnaire is to investigate and understand the architectural practices used in the KDE ecosystem in different parts of the world. The research for this thesis is being conducted at Federal University of Bahia, Brazil. The results of the questionnaire will be used in an evaluation study of Software Ecosystems Practice Improvement. Your opinion is very important to the success of our research.

We chose you because we consider you to be an expert in the development of the KDE ecosystem. All the information provided will be kept confidential and we will not publish the names of the respondents in the study.

1 - What is your name and how long have you worked with KDE?

2 - On which projects of KDE do you work?

3 - In which application domain do you work?

4 - What are the sizes of the projects on which you have worked?

5 - What are your activities in the KDE community?

6 - How is done the allocation of responsibilities in the projects that you work at KDE? Allocation of responsibilities means identify basic system functions and architectural infrastructure.

7 - How is choose the communication mechanisms (between systems, between your system and external entities, between elements of your system)?

Please, explain the process that you use to define important properties of the communication mechanisms such as stateful versus stateless, synchronous versus asynchronous, guaranteed versus nonguaranteed delivery, and performance-related properties such as throughput and latency.

8 - How are taken decisions about the data model? These decisions involve defining if the data is going to be kept in a relational database, a collection of objects, or both. Consider the different levels of data as platform level and application level.

9 - How is done the management of resources? These include hard resources (e.g., CPU, memory, battery, hardware buffers, system clock, I/O ports) and soft resources (e.g., system locks, software buffers, thread pools, and non-thread-safe code).

Do you take decisions alone or discuss with some groups? Are there some recommendations about KDE platform and applications? How is defined the sharing of resources?

10 - Is there some practice to build mapping among architectural elements? For example, mapping of modules and runtime elements to each other; the assignment of runtime elements to processors; the assignment of items in the data model to data stores.

11 - How are done binding time decisions? There are practices about selection of modules via a parameterized makefile, runtime negotiation of protocols, and acceptance of new peripheral devices plugged in at runtime?

- 12 - How is done the choice of technology (IDEs, simulators, testing tools, etc.) to be adopted in the KDE?
- 13 - Are there some practices do define the support available for the technology (such as courses, tutorials, and example)?
- 14 - How is determined if a new technology is compatible with the existing technology stack? Are there some practice used to monitor or manage the new technology adopted?
- 15 - How is defined the quality managed at KDE ecosystem? Is there some team specific to treat and define specific practices for quality management?
- 16 - How are defined quality properties and parameters to be adopted at KDE ecosystem?
- 17 - What are the practices for Architectural Knowledge management at KDE ecosystem?
- 18 - What are the practices for changes management in the architecture? How the changes are discussed and communicated to community? How is evaluated the impact of changes?
- 19 - How is controlled the access level for different parts of the architecture?
- 20 - How is done the prioritization of changes in the architecture in accordance with the requirements?
- 21 - How is managed all obsolete code and their different parts of the architecture?
- 22 - Is there a policy or directives to adopt reuse in the architecture of KDE?
- 23 - How is discussed and defined the usability issues in the architecture of KDE?
- 24 - How is managed the security side in the architecture of KDE?
- 25 - How is managed the financial viability of the architecture of KDE?
- 26 - Is there some practice to lead with language differences (English accent, skills, etc.) concerning the software architecture? How is managed the misunderstanding in the team about architecture issues?
- 27 - Are there some practices to manage the concept of time concerning architectural issues? For example, define meetings in different time to encompass different time zones, determine schedule changes considering all KDE community, difficult to find a common meeting time.
- 28 - Is there some practice to lead with differences in national culture concerning architectural activities? For example distribution of workload, synchronism of communication (some cultures assume that colleagues are available 24/7).
- 29 - What are the architectural practices when occurs face-to-face interaction?
- 30 - How is defined and managed the differences in autonomy in the architectural practices in KDE ecosystem? What are the different levels of autonomy to work on the architecture?