

Evaluation of a new Methodology for creating Architectural Solutions

The purpose of this survey is to get feedback about a new methodology for creating architectural solutions.

PRIVACY INFORMATION:

The outcomes of this survey will be shared with the scientific community. Participation in the survey is anonymous, and the results will be presented in anonymized form without revealing details that could associate individual claims or details with corresponding individuals or companies. Contact information is optional and will be solely used for communicating any further development of activities as defined by the purposes of this survey.

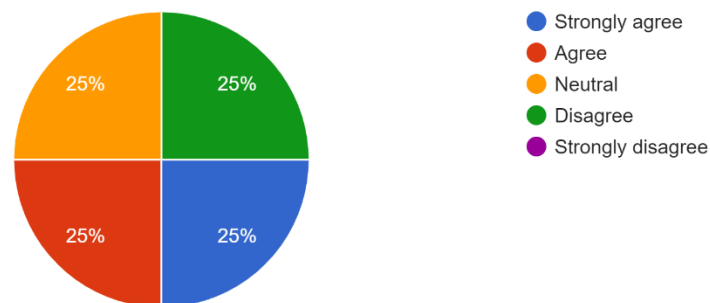
Contents

C4 System Context Diagram	1
C4 Container diagram and Component diagram	5
Toolchain	9
Conclusion.....	13

C4 System Context Diagram

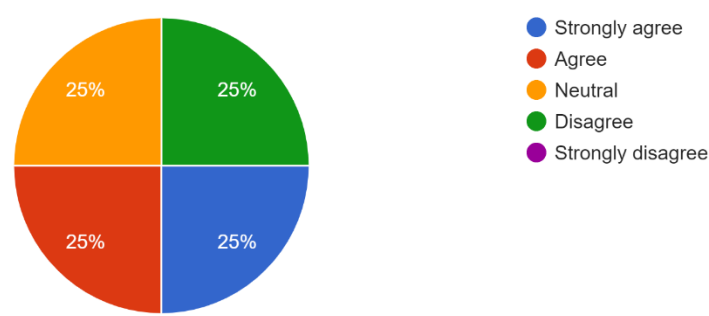
The presented alternatives (at least one alternative) capture the most relevant stakeholders and external systems for the problem at hand?

4 responses



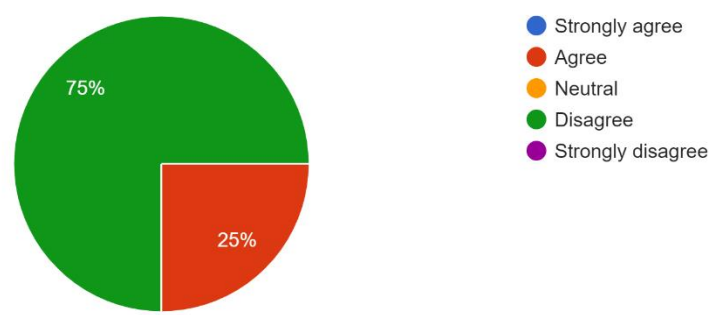
The number of components and their level of details (at least one alternative) are adequate for the problem at hand?

4 responses



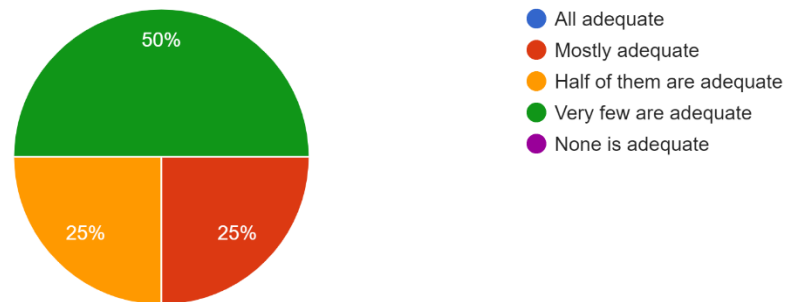
The graphical representation of architecture is adequate?

4 responses



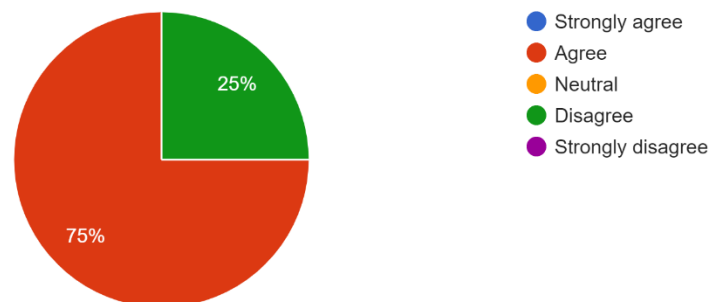
The suggested alternative solutions, for the problem at hand, are

4 responses



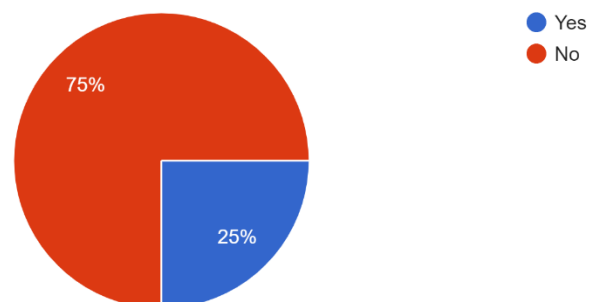
The connections between components (at least one alternative) are adequate (contain adequate data description)?

4 responses



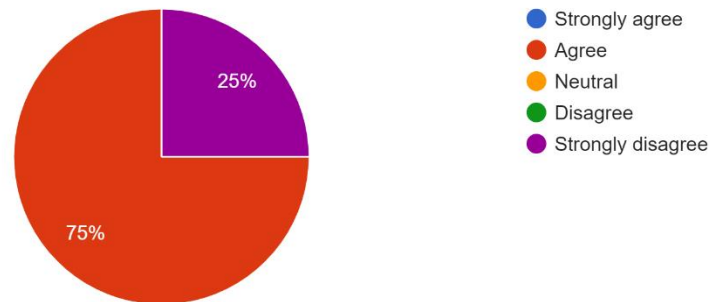
There are some alternatives that you would choose (e.g., more complete, better structured) to replace your original solution?

4 responses



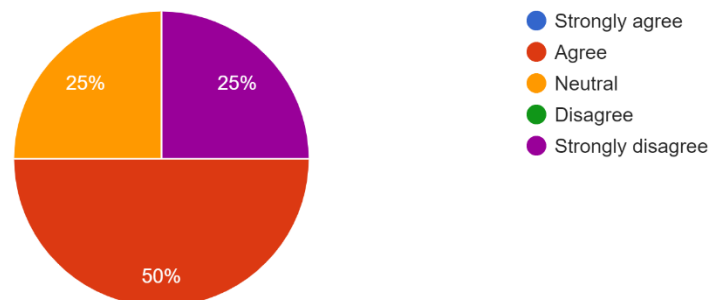
If you had the alternatives upfront, they would help you to reach the solution faster?

4 responses



If you had the alternatives upfront, they would improve the quality of your existing solution?

4 responses



What is missing (in general, on these diagrams)?

3 responses

The alternatives provided do a quite good job of capturing the system requirements and turning them into design solutions. However, what I feel is missing is hierarchy between presented data. In some alternatives we have high level concepts coupled with very low-level concepts. Furthermore, the colorization of some text can be a little misleading. To summarize, I feel that clarity and hierarchy between concepts is missing.

not all actors are listed, for example the architect and devops engineer, if we assume that the developer performs these as activities, it has to be shown as different actors. having different users is critical because of the different roles they perform

n/a

What is improper or misleading (in general, on these diagrams)?

4 responses

System boundaries are not well defined

Data flow or in general the abstraction level of components is not similar.

the level of abstraction is not well maintained in these diagrams, like mentioning microservices which is more on container level or even lower at system level diagram, this should be avoided, same applies to showing tests. not showing all actors

The directional flow is misleading in some of the system context diagrams, which can make the starting state/End state ambiguous.

What is redundant and unnecessary (in general, on these diagrams)?

3 responses

Colorization (red, blue, orange). If one component (one box instance) is not able to convey its meaning without colours, than maybe it should be redone. I guess only bolding and not bolding should be enough.

having more detailed info like mentioning microservices and tests on system level diagram,

Components such as 'pre-processing' is not needed.

What is new and useful (anything that you saw in any alternative)?

3 responses

For someone who is experienced at a particular technology that these alternatives present, than you can clearly see what it does not work. It helps you focus and it narrows down your scope. Overall, it can improve productivity.

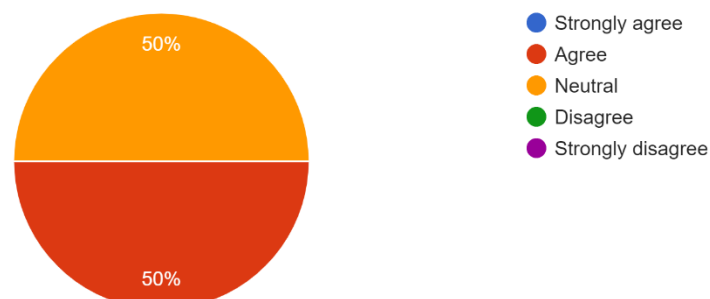
showing the tools used in the process as part of the system, like CI system or source code repository, showing the DevOps process as a system is useful

Alternate 6 seems most useful, as it gives a possibility for interacting with separate stakeholders within in the overall system design

C4 Container diagram and Component diagram

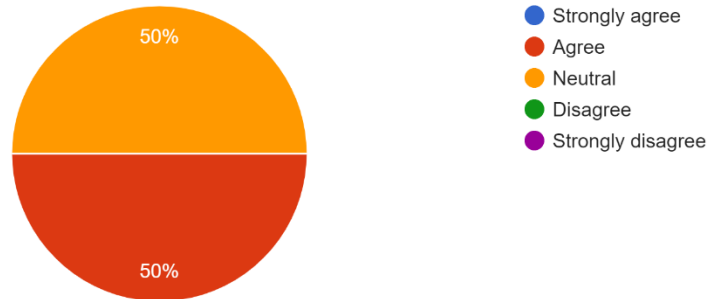
The proposed alternatives (at least one alternative) tend to capture the most relevant high level system functionality for the problem at hand?

4 responses



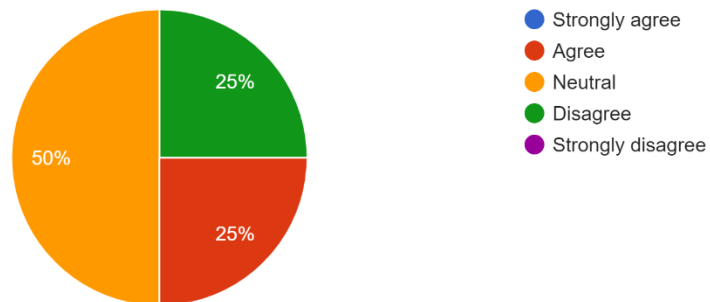
The proposed solutions (at least one alternative) suggest solutions that adress non-functional and quality requirements for the problem at hand?

4 responses



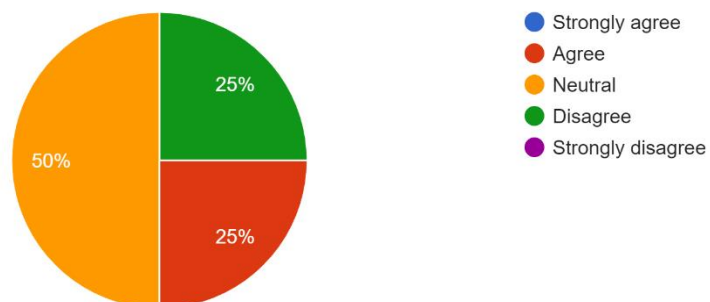
The number of components and their level of details (at least one alternative) are adequate for the problem at hand?

4 responses



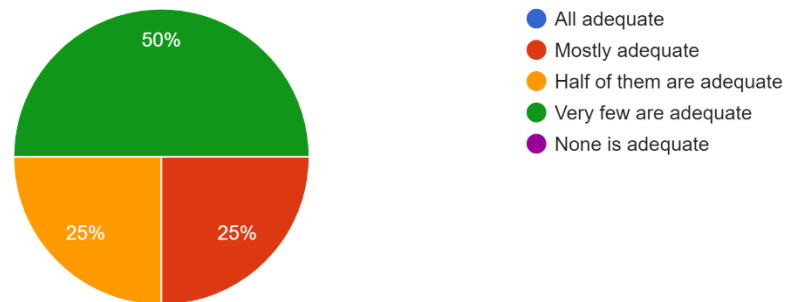
The graphical representation of architecture is adequate?

4 responses



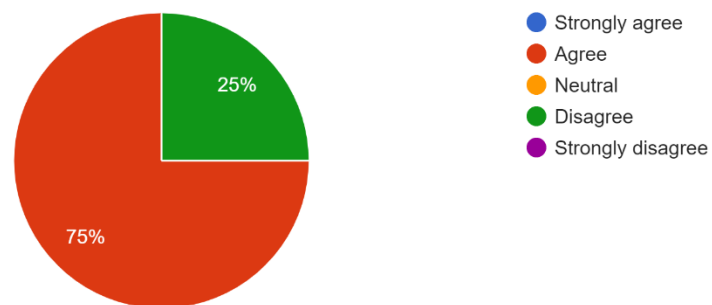
The suggested alternative solutions, for the problem at hand, are

4 responses



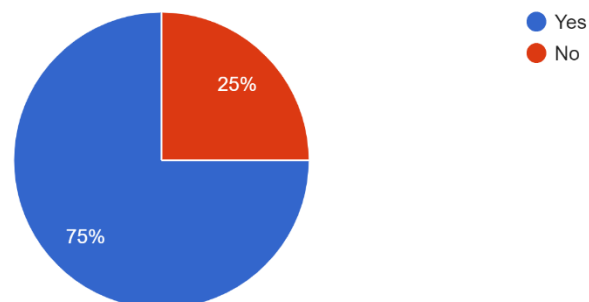
The connections between components (at least one alternative) are adequate (contain adequate data description)?

4 responses



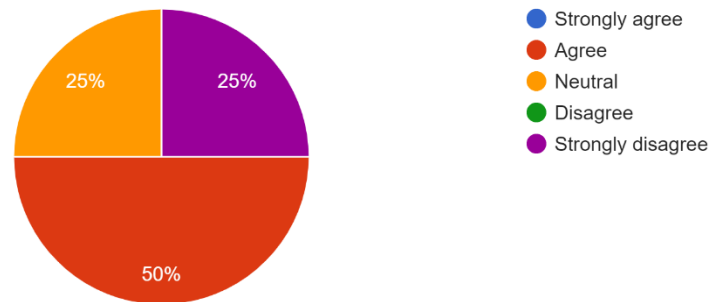
There are some alternatives that you would choose (e.g., more complete, better structured) instead of your original solution?

4 responses



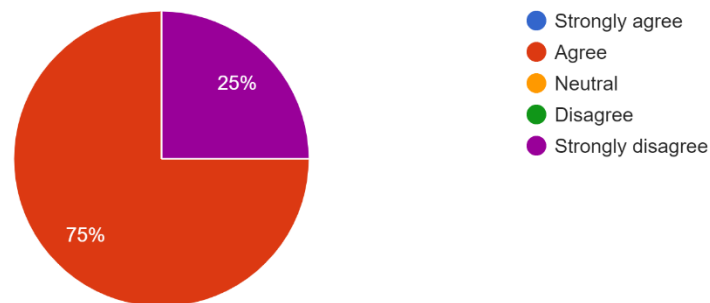
If you had the alternatives upfront, they would help you to reach the solution faster?

4 responses



If you had the alternatives upfront, they would improve the quality of your existing solution?

4 responses



What is missing (in general, on these diagrams)?

3 responses

Legend as far as Container and Component Diagrams are concerned plus things mentioned at Context part of evaluation.

there is no clear indication of how the interactions are done, basically on the original diagram I tried to highlight the feedback loop the developer has, however I can't see it clearly in these diagrams

The container view, should had elaboration on the AI phase, in regards to separate pre-processing that is done.

What is improper or misleading (in general, on these diagrams)?

2 responses

Hierarchy. The inexperienced people would have the feeling that everything is of a same importance.

some mixture of abstraction on some of the diagrams, for example showing devops process as a container and the api gateway interacts with it. listing client, who is the client: developer, architect, end user? "alternative 04"

What is redundant and unnecessary (in general, on these diagrams)?

2 responses

Colours

services breakdown, protocol information "HTTP, JDBC"

What is new and useful (anything that you saw in any alternative)?

2 responses

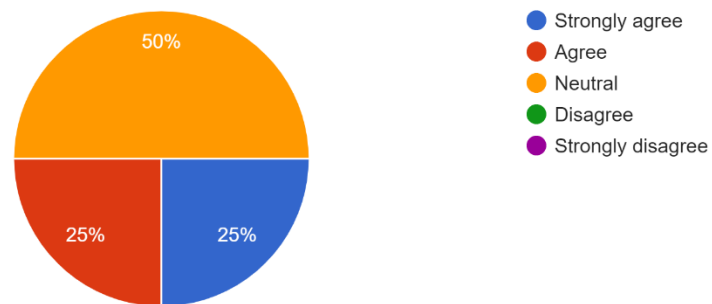
I saw the different ways that data can flow and how abstractions can be organised. It was like looking at Rubik's Cube from different angles.

showing the api gateway as a container, although I think it is bit detailed on such abstraction level, but it gives more details of the architecture of the system

Toolchain

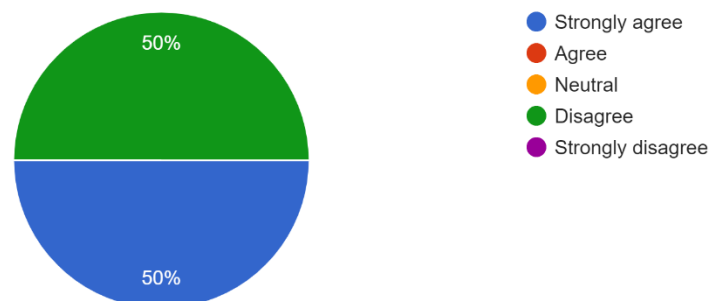
The proposed alternatives (at least one alternative) tend to capture the most relevant tool functionality that would help for the problem at hand?

4 responses



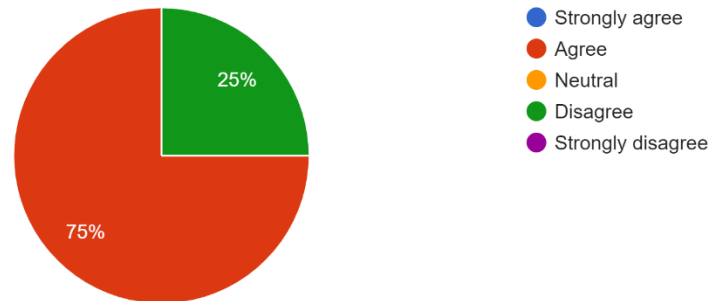
The proposed toolchain, including suggested programming languages (at least one alternative), is adequate for the problem at hand?

4 responses



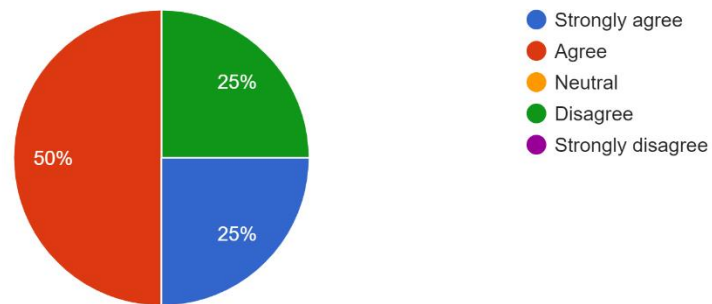
The number of components and their level of details (at least one alternative) are adequate for the problem at hand?

4 responses



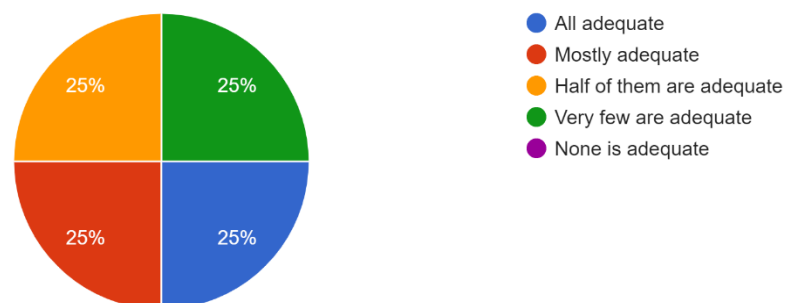
The graphical representation of architecture is adequate?

4 responses



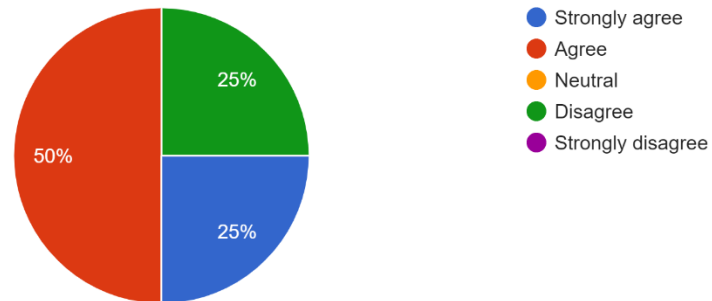
The suggested alternative solutions, for the problem at hand, are

4 responses



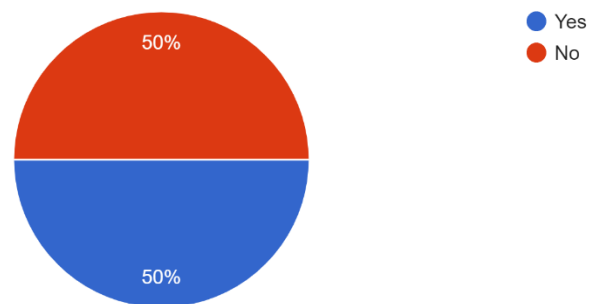
The connections between components (at least one alternative) are adequate (contain adequate data description)?

4 responses



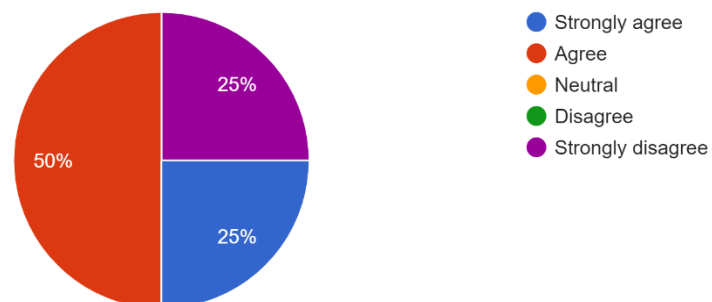
There are some alternatives that you would choose (e.g., more complete, better structured) instead of your original solution?

4 responses



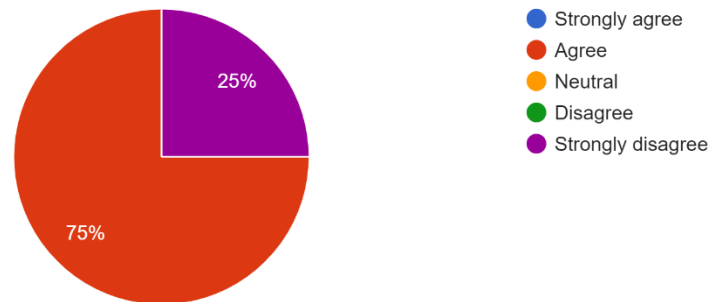
If you had the alternatives upfront, they would help you to reach the solution faster?

4 responses



If you had the alternatives upfront, they would improve the quality of your existing solution?

4 responses



What is missing (in general, on these diagrams)?

3 responses

I had a feeling that diagrams captured well toolchain process. Since selecting a particular toolchain is a very technical process which can be specified by logical rules, diagrams manage to capture well what is necessary.

task management tool, knowledge sharing and documentation tool "jira and confluence", some relations are not clear "what is the relation between aws cli and test

General implementation of a REST API is missing

What is improper or misleading (in general, on these diagrams) ?

2 responses

I did not have a feeling that something is misleading.

some relations are not clear, for example having aws cli and then automated tests and then test runner. same for DevOps Platform (Java, Ruby, Python), these are not DevOps Platforms

What is redundant and unnecessary (in general, on these diagrams)?

2 responses

I would go again with colour.

in some diagrams there are tools plus functionality, like saying REST and Kafka "kafka is a tool while rest is more of a concept or methodology"

What is new and useful (anything that you saw in any alternative)?

2 responses

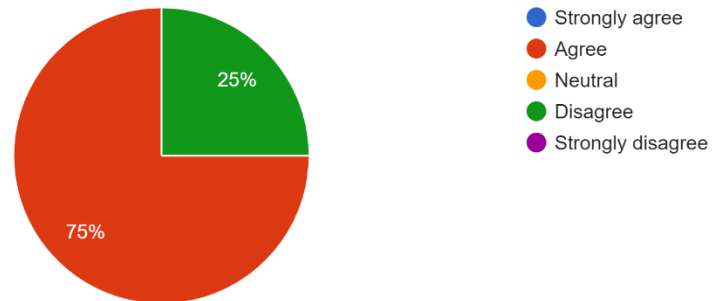
In my particular case, debug tools, unit testing and other important tools are new to me.

highlighting the functionality of the tool "alternative 03"

Conclusion

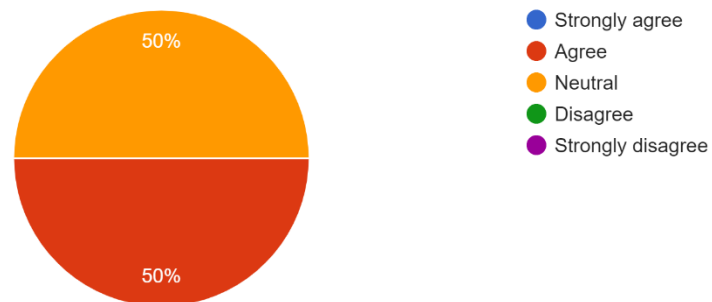
The proposed solutions (at least one alternative, any combination of diagrams) is capable of performing the specifications that were given (functional requirements are met)?

4 responses



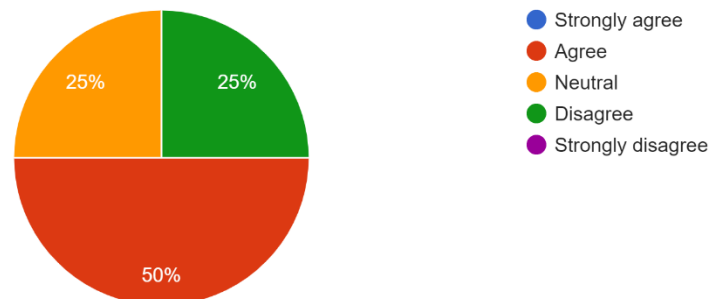
The proposed solutions (at least one alternative, any combination of diagrams) adress non-functional and quality requirements for the problem at hand?

4 responses



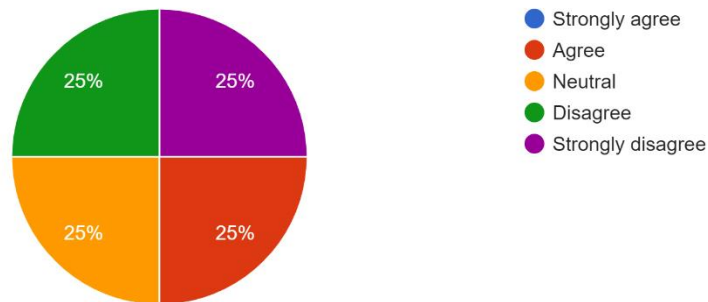
Considering all alternatives, the decomposition of components is adequate enough?

4 responses



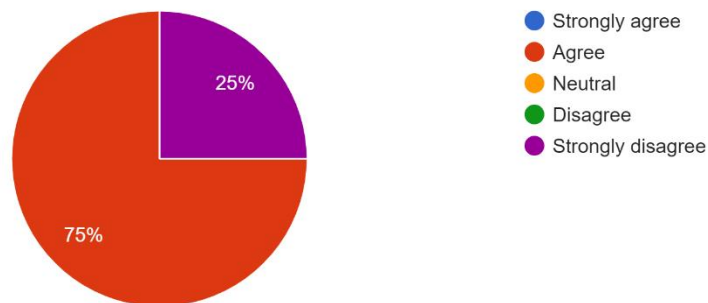
The graphical representation of architecture is adequate?

4 responses



Providing these alternatives upfront would enable guidance towards the solution?

4 responses



What is missing (in general, on these diagrams)?

2 responses

Clarity and Hierarchy between components.

to highlight the main feature of the original solution, that is the extended feedback loop within the development process, showing more users along with their interactions at different diagrams

What is improper or misleading (in general, on these diagrams) ?

2 responses

Sometimes data flow.

mixing different abstraction levels on the same diagram

What is redundant and unnecessary (in general, on these diagrams)?

2 responses

Colour

each alternative provides an edge over the others but lacks on aspects other alternatives offer, these could be consolidated to have more attractive alternatives.

What is new and useful (anything that you saw in any alternative)?

2 responses

Narrowing the scope of possible solutions, bringing focus. It is a useful aid.

the breakdown of the process on different levels of abstraction along with the details provided on these levels.