The three problems from the observation are: determine if the architecture and design is in use, explain how it is being used and judge how useful it is in general. I think these questions can be answered by following a practical process:

* **Create or retrieve the system list:** To answer the question if the documented architecture and design is used, we need to list all the systems used before and currently being used in the company, and then create a list of all systems documented in the architecture and design documentation. Once we have these two lists, we can cross comparison and find out the answer to the question. The final result should only include names of the systems and the status because it is for CIO to review which architecture and design documentation is used and which are not: no detailed or technical description or diagram needed. For example, the list can be:

|  |  |  |
| --- | --- | --- |
| System Name | Documentation | Status |
| System A | Architecture and design documentation A | Used/Retired |
|  | Architecture and design documentation B | Not used |
| System C | Architecture and design documentation C | Currently using |

* **Create or retrieve the context diagram:** To answer the question how architecture and design documentation is used, firstly we need to know what it does. To serve this specific purpose, the context diagram should only include high-level non-technical description of the system, the introduction of user classes and the relationships between them. It is because the main purpose of the context diagram is an initial understanding of the documentation : “What does it do? Who can use it ? What can I do with it as different user classes?”. Once we have the context diagram, the CIO can easily understand, decide and evaluate what the documentation wants to do, where and how the system can be used. For example, the documentation can be a medical record system, the user classes are doctors, nurses and patients. The context diagram shows the medical record system stores medical records and all user classes can visit the medical record system, and that is all the information.
* **Create or retrieve the container diagram:** To answer the question how architecture and design documentation is used, the second step is to know the structure. The container diagram should include high-level technologies to explain the structure of the system such as database, web application, mobile application, etc. Once we have the container diagram, the CIO knows the structure so he can plan, evaluate, improve and make decisions based on his understanding of the documentation. For example, if the documentation is a medical record system and it has a web application and a database. Based on the container diagram, the CIO can examine the working situation of the database, or understand that he or she can access the system through the web.
* **Create or retrieve the component diagram:** My understanding about “useful in general” is to see if something can be used in another place, to avoid or reduce time and money cost. Therefore, to answer the question how architecture and design documentation is useful in general, a component diagram is necessary for determination. For example, if there is a search component in documentation A, and the CIO validates the search component can be merged to documentation B, C, D as well. Obviously the search component is very useful in general because it saves money and time. Therefore, since the components diagram contains information about high-level technology decisions, it can help the CIO understand how useful the documentation is.
* **Create or retrieve the class diagram:** The class diagram is high-level implementation, therefore even a component may not be eligible for other documentation, some classes in the component may be useful. For example, the patient class in the medical record system can also be used in a hospital system, as long as the class is contained in the documentation. Therefore the class diagram can also answer the CIO’s question about determining if documentation is useful.
* **Use the same notations and technical language:** The stakeholders of the architecture and design documentation include developers, therefore language used in the diagrams and documentation should be technical: use the same and professional standard such as terminology, code, etc. Although inconsistent notations happen very often, the CIO should make a notation standard for the company, enforce everyone to comply so there will be no misunderstanding or conflicts about it.