Basic Concepts of Software Architecture

1. Architects focus on system qualities

Architects are the artifact that use different architecture than the others that have the same feature and focus on arised qualities.

2. Tradeoffs

Tradeoff is the feature or value that is disadvantageous in order to satisfy other features focused in that method. In other words it is a cost to pay to achieve what is focused.

3. The Differences between Architecture, Architecting, and Architects

Architecture is a role of people. (job title)
Architecting is a process or action of designing something.
Architect is an artifact that architecture design/try to build.

4. Views

View is a projection of a model, or a relationship between 2 models. (helps picture the flow/model)

5. Quality Attributes

Quality Attributes are the abilities of the software other than the functional part (functional - what the software can do). It used to evaluate the system. For example performance, scalability, usability, security, throught.

6. Analysis

Analysis is the process to conclude/understand something from the data you have. It easier to analysis data if you have view, since it will easier for you to see the overview image of the you want.

7. Standard Notations

Standard Notation is a format of writing that is clear and consistent, to make people understand the same thing correctly.

8. Guide Rails/Constraints

Guide Rail is an obstacle that limits or slows the process of working. For example when trying to find something in a database with bad design, it’s harder or sometimes slower than the one with good design.

Or when the web scalability is low, when too many people use it at the same time the server might fail. However, at the same time, it can also help ensure that your software quality attributes are met.

9. Architectural Styles (aka Architectural Patterns)

Architectural Styles are styles of system/design/process. There are many different styles. Each of them can help you to know its elements, constraints, and tradeoffs. To choose the suitest method for your project, you compare its possible styles.

10. Conceptual Model

Conceptual Model is a set of concepts that can be imposed to provide meaning and structure. It is a rule or a theory that explains the fact of something. In software architecture, it can be the meaning of concepts about model relationship, canonical model structure, quality attributes, and other.

11. Engineering with Models

Engineering with Models is a way to develop something with patterns/models. It can make the process smoother, faster, and cheaper. It doesn't mean it is more correct, but it helps you reduce your resource use.

12. Canonical Model Structure

Canonical Model Structure is a structure of models with different expressions/formats. (Contains domain, boundary, internals, code models)

13. Models and Code

The difference between architecture (model) and code is that for people or even for programers code is hard to tell what it does, but architecture in model is easier to understand. It provides hints that humans can understand rather than no meaning variables. So it benefits developing and documentation.

14. Process and Risks

In the process of building/developing something there are risks that something could go wrong. It mostly happens when you do not worry about what you are doing. But there are many techniques to mitigate the risks – protocol analysis, queuing theory, threat modeling, etc.

15. Risk-Driven Approach for Software Architecture

Risk-Driven Model is a model to first identify and prioritize the risks, second relevant architecture, and third to re-evaluate the risks (looping it until it works out for you). But you need to balance wasting time on low-impact techniques or ignoring project-threatening risks.