**Architecture in a technical context**

**Architectures enable the achievement of quality attributes:**

* If your system needs high performance, you need to pay importance to time based behavior of elements.
* If you care about the availability of the system, you need to pay importance to how components take over each other in the case of failure.
* If you care about usability, you have to isolate components of the system that deal with the user experience, from other parts of the system.

All of these points mean that if you care about a particular attribute, you have to make *these* decisions.

**Architectures and the technical environment:**

The technical environment used at the time when the architecture is designed will influence the architecture, this environment includes the standard industry practices and the software engineering techniques used in the architect’s community.

**Architecture in project life cycle context:**

**Waterfall Model:** Consists of a series of sequential activities, each with entry and exit conditions and a formalized relationship with its upstream and downstream neighbors. This is the order of the processes: Specification, design, implementation, integration, testing, installation, maintenance. Feedback paths were allowed and testing could result in reimplementation or even redesign.

**Iterative Model:** Early requirements lead to an early design which is then implemented and tested while more requirements are being listed. Then these new requirements are added onto the previous working model which is then again implemented and tested. Each of these stages is referred to as a cycle. A cycle must deliver something useful.

**Agile Model:** Popular ones include extreme programming, scrum and crystal clear. They are all incremental and iterative. Focuses on early and frequent delivery of working software, close collaborations between customer and developer, self-organizing teams, and a focus on adapting to change in requirements.

**Model Driven Development:** Based on the idea that humans should not write code but rather create models of the domain from which the code should be generated automatically. Humans create platform independent model (PIM) which is combined with platform definition model (PDM) to automatically generate a code.

All of these models require design which in turn requires an architecture. In case of need to change from one of these processes to a new process, the old architecture must be able to extract useful information from the previous process and integrate it into the new process.

**Making a business case for the system:** A business case helps you understand how a business decision will affect the organization. Initially there is a choice to whether start the project or not, after initiation the business case is then reviewed to see if the previous estimates were correct and then alternate potential decisions are looked upon. The business case also serves as the repository for storing business and marketing data.

**Understanding the architecturally significant requirements:** Appropriate techniques for eliciting requirements include use cases, finite state machine models. Requirement’s elicitation includes understanding the prior systems, which are similar to the system being built. Creation of prototype also helps to understand requirements because it provides to the stakeholders a model of what the system could potentially look like.

**Creating or selecting the architecture:** Conceptual integrity is the key which can only be achieved by a small number of minds working together to create the architecture.

**Documenting or communicating the architecture:** The architecture document should be informative, unambiguous, minimal, readable and aimed for the various stakeholders who will use it. Developers must understand the work assignments, the testers must understand the task structure and the management must understand the scheduling constraints.

**Analyzing or evaluating the architecture:** It must satisfy the needs of the stakeholders by achieving the quality attributes requirements. Scenario-based techniques provide the most considerable way to analyze this.

**Implementing and testing the system based on the architecture:** Architectural conformance is important. The architecture must be explicit and well communicated, the infrastructure to support the developers in building that architecture must be there. There is no point in constructing a conceptually sound architecture if the developers just ignore it for some reason.

**Ensuring that the implementation conforms to the architecture:** After an architecture is created, it is checked if the actual architecture (in the documents) and its implementation are the same and if they are different then effort must be made to either fix the implementation or change the document.

**Architecture in a business context**

**Architectures and business goals:** The architect must understand the goals of the vested organizations. Business goals are manifested as quality attribute requirements, that is every quality attribute requirement must be derived from some business-oriented purpose. However not all the business goals will show up in the architecture, an example would be “keeping staff gainfully employed”, asking employees to work from home to cut costs.

**Architectures and the development Organization:** An organization usually has assets such as a team skilled in a particular field, existing architectures, similar systems. The organization might accept a project which is next in line to an existing project and a high degree of asset reuse. An organization might also decide on a project based on its strategic value by considering a long-term business investment, for example it can choose a cloud computing project if it wants to build a reputation in cloud computing. Organizations are organized around technology and related concepts so there are networking groups, database groups, business groups etc so the subsystems of a software architecture is given to a team of the same name or is related to that name.

**Architecture in a professional context:**

Architects need good diplomatic, negotiation and communication skills to be able to effectively communicate to stakeholders the chosen priorities and why some of there expectations are not being fulfilled. They need to have up to date knowledge for example about database platforms. They need to be able to support the management and dealing with the customers, they need to know business considerations.

Architectural choices may come from experience; for example if you have had good results from using a particular architectural approach, you are likely to use it again but conversely if you had bad results from a particular architectural approach then you are less likely to do so. Architectural choices can also come from education and training; for example: architectures that you might have studied in a book, exposure to system or architectural patterns that have been successful or unsuccessful.

**Stakeholders:**

A stakeholder is anyone who has stake in the success of the system (customers, end users, developers, project manager etc). All of these stakeholders despite have a shared stake in the system, have different concerns that they wish to be guaranteed or optimized. Architects need to understand the nature and constraints of the project early on and for that they need to engage the stakeholders to explicitly state their needs; this allows architect to make trade-offs by negotiating with the stakeholders about any contradictory goal or concern and then coming to an agreement early on. Some qualities such as performance, reliability, availability, memory utilization, network usage, modifiability, interoperability is desirable by more than one stakeholder.

**How is architecture influenced:**

A software architecture is influenced by business and social groups, existence of similar architecture, deadlines, technologies being used at present. All five of the sections (Business, technical, project, professional, stakeholders) influence the architecture.

**What do architectures influence:**

Conversely architecture has an influence on business, technical, project and professional contexts.

**Technical Context:** The architecture can affect stakeholder requirements by giving the customers the opportunity to receive a system that is based on a similar architecture rather than being built from scratch; the advantage of doing so could be less cost, reliability, timeliness etc and because of this the stakeholder might be willing to remove a few of his requirements or rather modify them differently. Shrink-wrapped software provides solutions that are inexpensive, of high quality and are not expensive to an individual’s needs because of which it has made an impact on requirements.

**Project Context:** An architecture decomposes a system into units which must be implemented and integrated to form a system. These units form the basis for the formation of development teams; for example, teams are formed for development, testing, integration etc. If a company feels the need to continue onwards with similar projects, it is highly likely that the categorization of teams will be more or less the same and thus the company will invest in each team in trying to nurture their expertise. This can be considered as a feedback from the architecture to the development organization.

**Business Context:** A successful system built using a particular architecture might enable a company to assert its dominance in a particular market segment causing the organization to even change its business goals owing to its newfound position in the market. That architecture can then provide opportunities for efficient production and deployment of similar systems.

**Professional Context:** A successful system built using specific architectural techniques will encourage the architect to use those techniques for subsequent systems while an unsuccessful system will discourage the architect in using similar architectures or architectural techniques.

These and other feedback mechanisms included form what is called the Architecture Influence Cycle.