
CSPC61, EMBEDDED SYSTEMS AND ARCHITECTURE
CHAPTER-11: DEFINING THE SYSTEM – CREATING THE ARCHITECTURE AND DOCUMENTING THE DESIGN

---

1. Draw & describe the 4 phases of the Embedded System Design and Development Lifecycle Model.

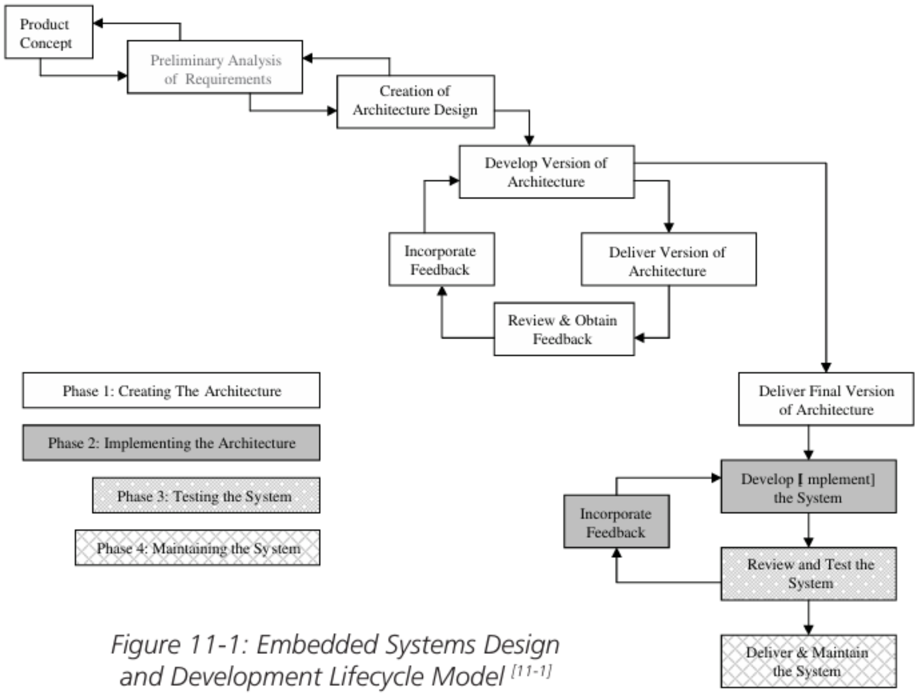

Figure 11-1: Embedded Systems Design and Development Lifecycle Model [11-1]

This model indicates that the process of designing an embedded system and taking that design to market has four phases:

- *Phase-1*: Creating the Architecture, which is the process of planning the design of the embedded system.
- *Phase-2*: Implementing the Architecture, which is the process of developing the embedded system.
- *Phase-3*: Testing the System, which is the process of testing the embedded system for problems, and then solving those problems.
- *Phase-4*: Maintaining the System, which is the process of deploying the embedded system into the field and providing technical support for users of that device for the duration of the device's lifetime.

---

2. Of the four phases, which phase is considered the most difficult and important? Why?

The most important time is spent in phase 1, creating the architecture. At this phase of the process, no board is touched, and no software is coded. It is about putting full attention, concentration and investigative skills into gathering information about the device to be developed, understanding what options exist, and documenting those findings.

---

3. What are the six stages in creating an architecture?

- *Stage 1*: Having a solid technical base;
- *Stage 2*: Understanding the architecture business cycle;
- *Stage 3*: Defining the architectural patterns and reference models;
- *State 4*: Creating the architectural structures;
- *Stage 5*: Documenting the architecture;
- *Stage 6*: Analysing and evaluating the architecture.

---

4. What are the ABCs of embedded systems? Draw and describe the cycle.

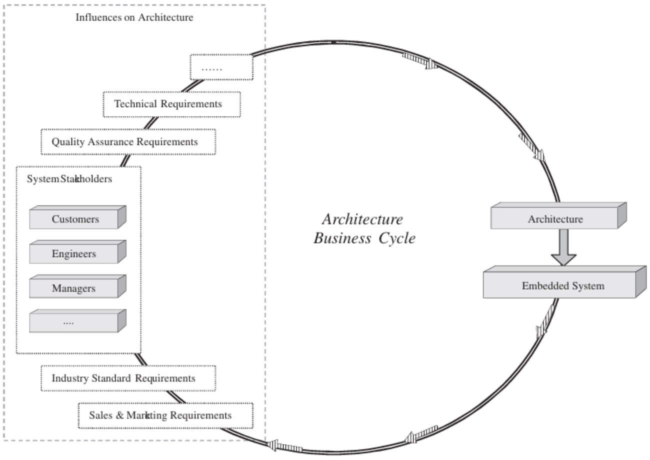

Figure 11-3: Architecture Business Cycle [11-2]

The ABCs of embedded systems are the many different types of influences that generate the requirements of the system, the requirements in turn generate the architecture, the architecture then produces the system, and the resulting system in turn provides requirements and capabilities back to the organization for future embedded designs.

---

5. List and define the four steps of Stage 2 of creating the architecture.
- *Step 1*: Understanding that ABC influences drive the requirements of an embedded system, and that these influences are not limited to technical ones.
- *Step 2*: Specifically identifying all the ABC influences on the design, whether technical, business, political and/or social.
- *Step 3*: Engaging the various influences as early as possible in the design and development lifecycle and gathering the requirements of the system.
- *Step 4*: Determining the possible hardware and/or software elements that would meet the gathered requirements.

---

6. Name four types of influences on the design process of an embedded system.
- *Business*: Sellability, Time-to-Time Market, Costs
- *Technical*: Performance, Reliability, Scalability, Portability, Availability
- *Industry*: Standards
- *Quality Assurance*: Testability, Availability, Schedule
- *Customers*: Cost, Performance

---

7. Which method is least recommended for gathering information from ABC influences? A. finite-state machine models / B. scenarios / C. by phone / D. in an e-mail / E. All of the above. Answer **D) in an email**

---

8. Name and describe four examples of general ABC features from five different influences.

Refer Q6

---

9. What is a prototype? How can a prototype be useful?

A useful tool in understanding, capturing, and modelling system requirements is through utilizing a *system prototype*, a physically running model containing some combination of system requirements. A prototype can be used to define hardware and software elements that could be implemented in a design, and indicate any risks involved in using these elements. Using a prototype in

---

---

conjunction with the general ABC features allows you to accurately determine early in the project what hardware and software solutions would be the most feasible for the device to be designed.

---

10. What is the difference between a scenario and a tactic?

In the context of Embedded Systems, a scenario often describes a potential usage or operational environment in which the embedded system will function. It could involve factors such as user interactions, environmental conditions, or system inputs. Conversely, a tactic in Embedded Systems refers to a specific implementation or approach used to accomplish a particular function or task within the embedded system's operation. While a scenario outlines the broader context, a tactic focuses on the detailed methods or strategies employed within that context to achieve desired system behaviour or functionality.

---

11. []

---

12. [T/F] A requirement can have multiple tactics. **True**.

Requirements specify what the system must do or accomplish, while tactics outline how those requirements will be met. Depending on the complexity of the requirement or the system itself, multiple tactics may be employed to fulfil it effectively. Each tactic represents a different approach or strategy for satisfying the requirement, offering flexibility in design and implementation.

---

13. What is the difference between an architectural pattern and a reference model?

An architectural pattern is a high-level blueprint or profile of an embedded system, outlining the arrangement and interaction of software and hardware components. It describes the types of elements present in the system, their functions, their interrelationships, and external interfaces.

On the other hand, a reference model represents a specific topological layout of the hardware and software elements within the embedded system. It serves as a concrete representation of the system's architecture, illustrating how different components are structured and connected.

---

14. What is the "4+1" model? Why is it useful? List and define structures that correspond to the "4+1" model.

The "4+1" model states that a system architect should create five concurrent structures per architecture at the very least, and each structure should represent a different viewpoint of the system. What is literally meant by "4+1" is that four of the structures are responsible for capturing the various requirements of the system. The fifth structure is used to validate the other four, ensuring that there are no contentions between the structures and that all structures describe the exact same embedded device, from their various viewpoints.

- *Structure 1:* A logical modular structure showcases key functional hardware and software elements, outlining their interrelationships to meet system requirements, aiding in system building and integration.
- *Structure 2:* A process structure reflects concurrency and synchronization, demonstrating how an OS meets nonfunctional requirements, such as performance and resource availability.
- *Structure 3:* A development structure maps hardware and software into the development environment, aiding buildability and considering constraints like IDEs and programming complexity.
- *Structure 4:* A deployment/physical structure illustrates how software maps onto hardware, ensuring nonfunctional requirements like hardware resource availability and reliability are met, defining hardware requirements based on software needs.

---

15. What is the process for documenting an architecture? How can a particular structure be documented?

- *Step 1:* Create a comprehensive table of contents outlining the architecture documentation, including an overview of the system, supported requirements, structure definitions, interrelationships, documentation layout, and modelling techniques.
- *Step 2:* Develop individual documents for each structure detailing supported requirements, design rationale, constraints, and graphical representations of structural elements, incorporating both graphical and nongraphical representations and outlining any external interfaces or protocols.
- *Step 3:* An architecture glossary. This document lists and defines all the technical terms used in all of the architectural documentation.

---

16. List and define two common approaches for analysing and evaluating an architecture? Give at least five real-world examples of either.

    1. *Architecture Level Prediction of Software Maintenance (ALPSM):* This approach evaluates maintainability through scenarios, assessing how well the architecture supports future maintenance activities.
    2. *The Architecture Tradeoff Analysis Method (ATAM):* A quantitative quality attribute approach that identifies problem areas and technical implications of an architecture through questions and measurements. It can be applied to evaluate various quality attributes.

Real-world examples:

1. *Maintainability of Operating Systems:* ALPSM can predict the ease of maintaining an operating system architecture over its lifecycle.
2. *Evaluation of Web Application Architecture:* ATAM can assess the scalability and performance implications of a web application architecture, helping to optimize its design.
3. *Cost-Benefit Analysis for Cloud Architectures:* CBAM, an extension of ATAM, can analyze the economic implications of adopting specific cloud architectures.
4. *Evaluation of IoT Device Architecture:* ISO/IEC 9126-1 thru 4 standards can be used to assess the reliability, usability, and efficiency of an IoT device architecture.
5. *Real-time Embedded System Design:* Rate Monotonic Analysis (RMA) evaluates the real-time behaviour of an embedded system architecture, ensuring timely task execution and system reliability.

---

17. What is the difference between a qualitative and quantitative quality attribute approach?

In embedded systems, a qualitative quality attribute approach involves evaluating system characteristics based on subjective judgments or descriptions, providing an understanding of attributes like reliability or power consumption without assigning numerical values.

Conversely, a quantitative quality attribute approach entails measuring attributes using numerical metrics, facilitating precise analysis and comparison of aspects such as response time or memory usage, thereby enabling objective assessment and optimization of embedded systems.

---

18. What are the five steps introduced in the text as a method by which to review an architecture?

- *Step 1:* The evaluation team receives architecture documentation, briefed on the evaluation process and content.
- *Step 2:* Architectural approaches and patterns are listed based on team feedback after analyzing the documentation.
- *Step 3:* The team and architects agree on scenarios derived from system requirements, prioritizing based on importance and implementation difficulty.
- *Step 4:* Emphasis is placed on evaluating high-risk and crucial scenarios.
- *Step 5:* Evaluation results encompass agreed requirements, benefits, risks, strengths, problems, and recommended design changes.
