Software Engineering for Embedded Systems

Chapter 5

Renesas Electronics America Inc.
Embedded Systems using the RX63N

Rev. 1.0

Topics

- Need for a structured development process
- Overview
- Software development stages
- Software development lifecycle models

Need for a structured development process

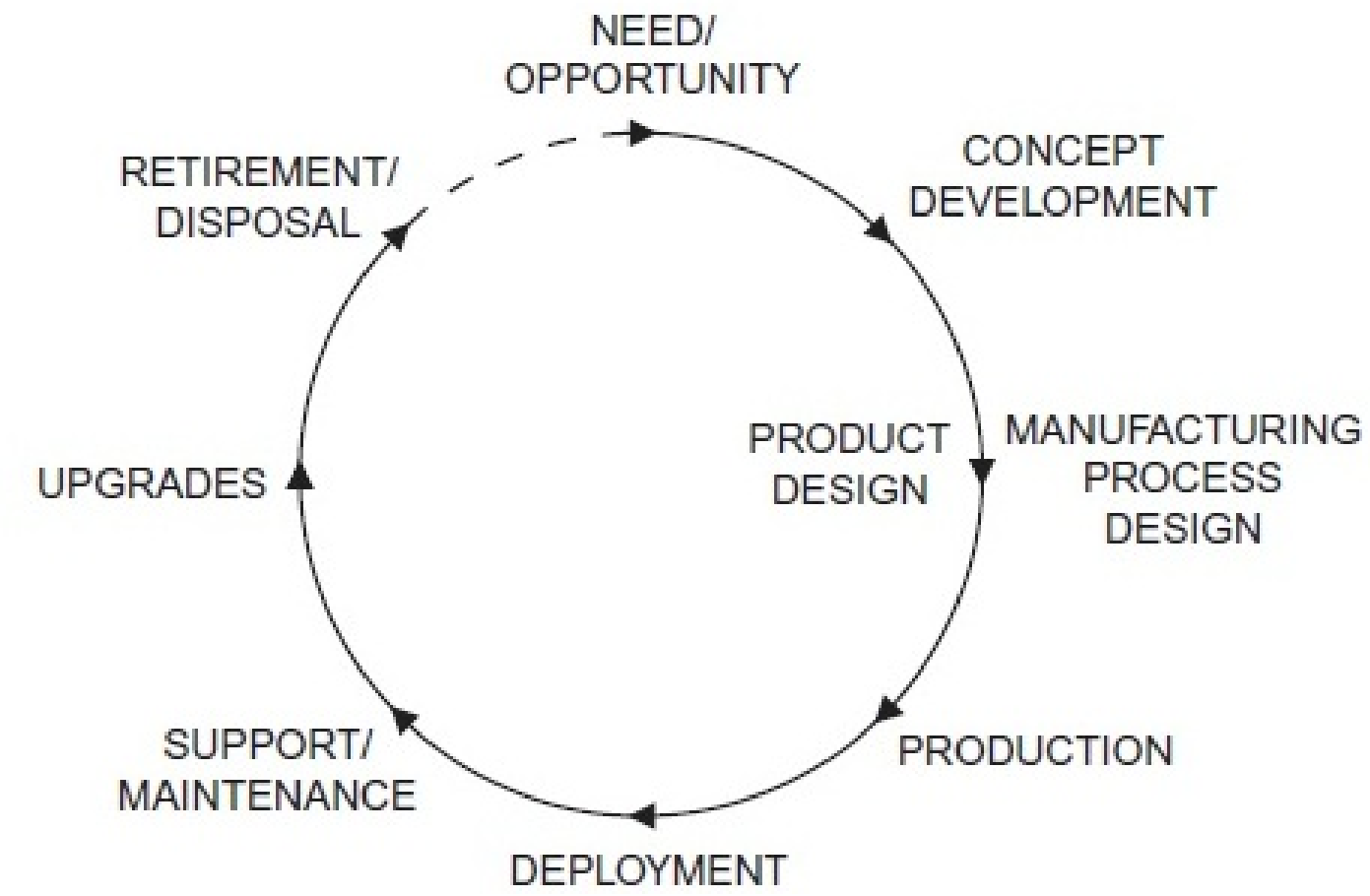
Consider the following common software system occurrences:

- Unpredictability of software
- Buggy code
- Unexpected behavior of peripherals
- Incompetency of developer
- Budget constraints

We need a structured process to reduce risks involved in software development process.

Overview

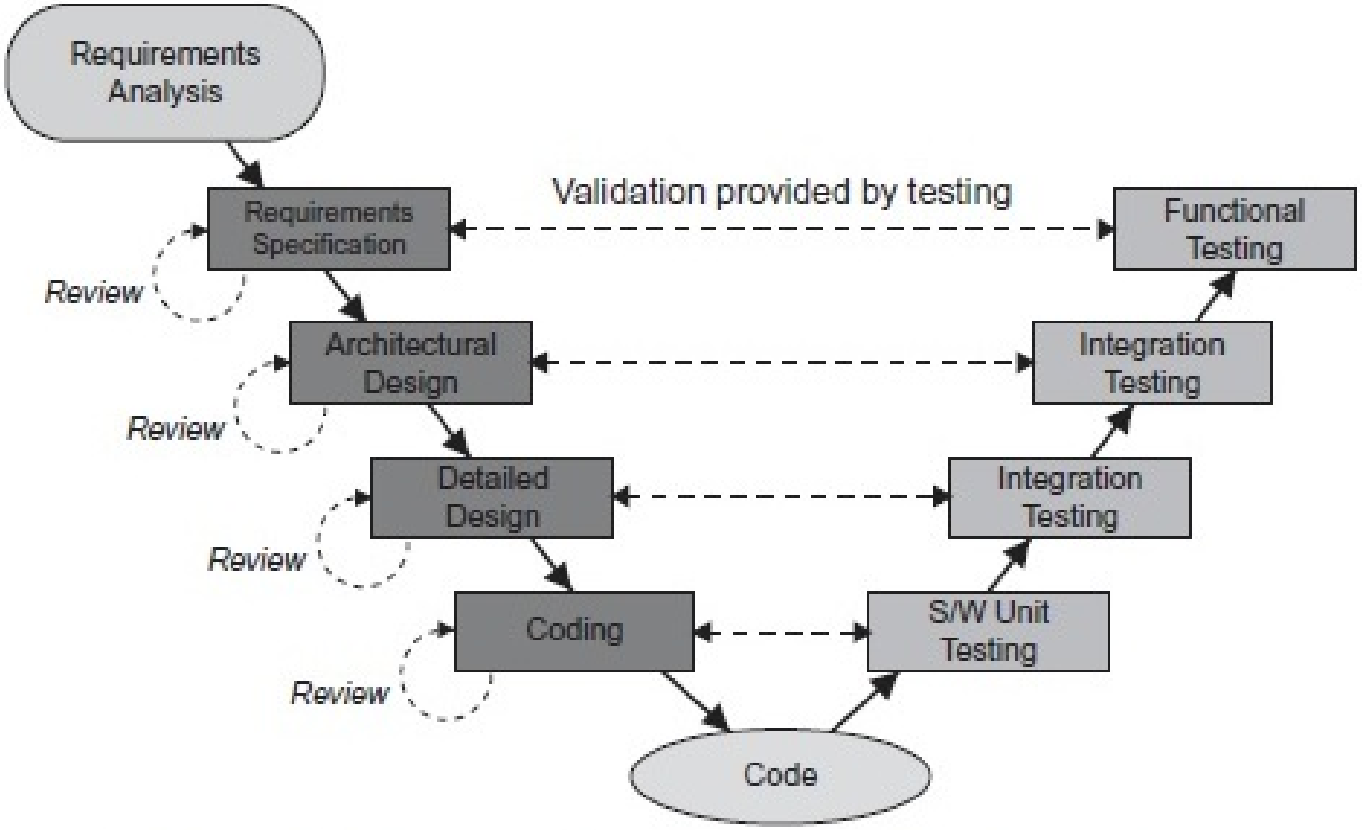
- The following figure simply explains various stages of an embedded systems lifecycle.



[2]

Example model

- The “V” model emphasizes testing at each level of the software development process.



Software Developments stages

1. Requirements
2. Design before coding
3. Peer review
4. System architecture and design approach
5. Detailed design
6. Implementation
7. Software testing

1. Requirements

- They have to be written down so that every team member has a clear idea of what is expected. Requirements can be expresses in text or graph.

- There are three types:

1. Functional requirements: state the expected functionality of the system
2. Non-functional requirements: describe system behavior like response time, energy efficiency
3. Constraints: define the limits on the system, like cost

2. Design before coding

- All the parts of the system should be understood to know what parts are going to be complex to code.
- It only saves time working on the code.
- Graphical representations like flowcharts make coding easier.
- Coding too soon or before designing may not be the most efficient way of implementation.

3. Peer review

- Getting a peer review helps in broadening the perspective on the project implementation.
- It helps detect oversights and misconceptions or mistakes.
- Also knowing that peers will be reviewing your software makes you careful and professional.

4. System architecture and Design approach

- The system architecture defines what processor and peripherals will be used.
- It should also divide the software into major parts.
- The documentation should include which processing is time-critical and safety-critical, and how the system is designed to take care of it.

5. Detailed Design

- It involves defining and designing the subsystems of the architecture.
- The software can be divided into functions and ISRs.
- It is better if the software design is defined in steps of algorithm or flowchart of every function.
- Designing should take into consideration all the requirements and constraints.

6. Implementation

- The detailed design should be actually coded and programmed into the processor.
- C is the most dominant programming language for embedded systems.
- The code should be simple, generic, and clear.
- The code should follow the team or company's coding standards.

7. Software testing

- It is impossible to create error-free code, for a system that will handle all possible input conditions, in a single try.
- Software testing verifies the completeness of the system and identifies the software's weaknesses.
- Black box testing tests the functionality of the software. How the software implements it is unimportant to the test.
- White box testing relies on the knowledge of the software. The tester makes sure he/she goes through every condition and function of the software and checks for desired results.
- Bugs occurred in the past are repeatedly checked. This is called regression testing.

Software Development Lifecycle Models

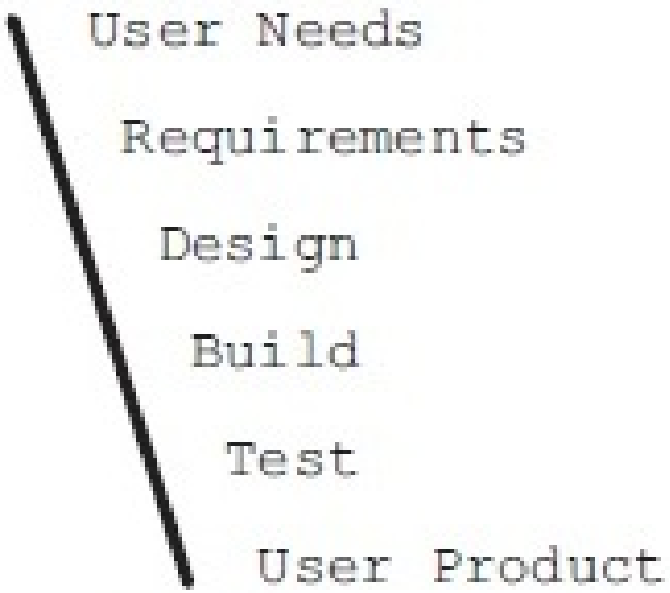
- The model which lists the steps in the lifecycle is the waterfall model. It is an idealized model.

- The models like iterative process, spiral process, agile development, and prototyping are actually used in software development.

Waterfall model

- It is an ideal model and there is no reviewing process involved between the stages.

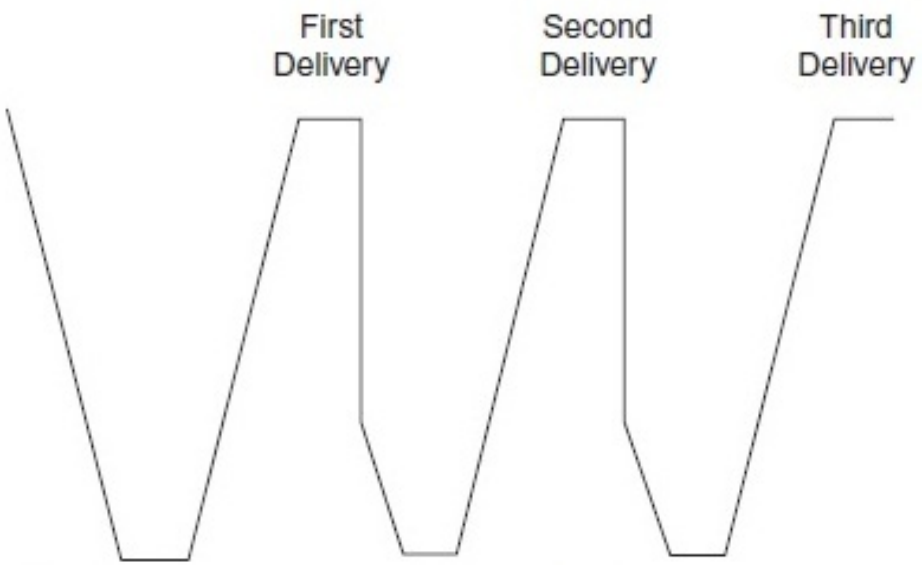
- The model is appropriate to follow when the problem is well understood and there is little risk.



Iterative process

- This process is used when the customer wants the product to be developed based on customer's feedback.

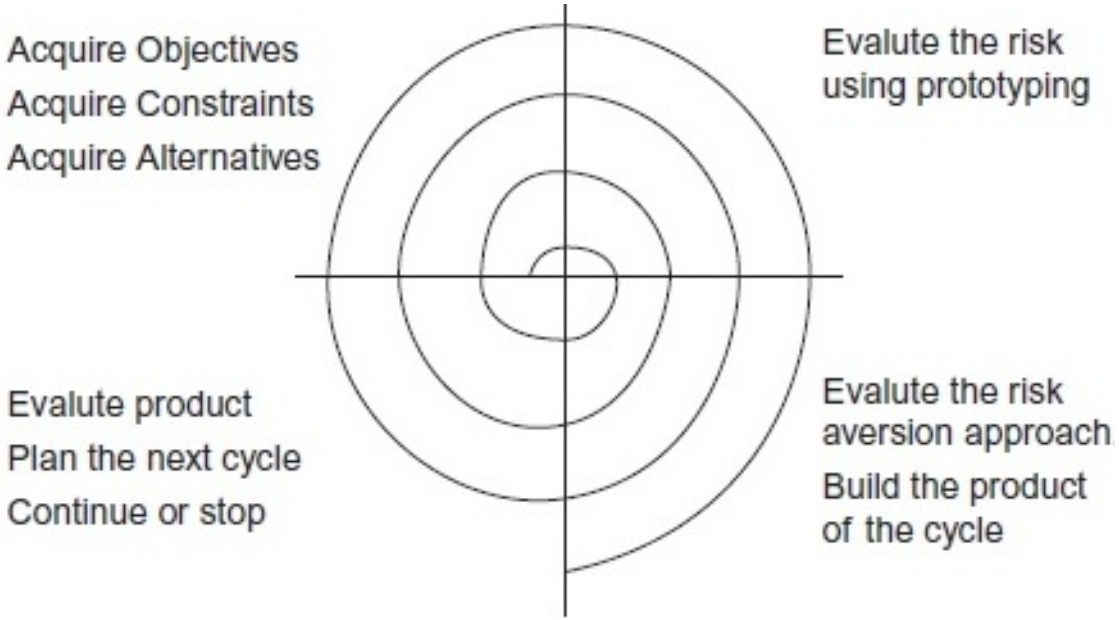
- This process is successful when the development time is short.



Spiral process

- This is a process which involves constant development over previous cycles, often used in high risk projects like research in new technology.

- Its stages are requirements, designing a prototype, building the product, and evaluation of the product, which leads to further development.



Agile Development process

- The most common agile development methods are scrum, extreme programming, and lean development.

- These methods seek to deliver the product to the customer as early as possible.

- With scrum, the development is divided into ‘sprints.’ Each sprint has specific goals to be achieved. The scrum master solves any problems faced by a team member and allows speedy completion of the sprint.

Prototyping

- In this process, the developer presents a prototype of the product to the customer.

- This works best when performed as an iterative or spiral process.

There are two basic types:

1. Throw away prototype: Developer uses demonstration tool or language to build a prototype and throws away the prototype to build the actual product with real programming language.
2. Evolutionary prototype: Developer uses code from a prototype to build the actual product.

Conclusion

The software development stages are:

1. Requirements
2. Design before coding
3. Peer review
4. System architecture and design approach
5. Detailed design
6. Implementation
7. Software testing

The various software development models that can be used are waterfall, iterative process, spiral process, agile development, and prototyping.

References

Unless specified all images taken from :

[1] Renesas Electronics, Inc. (February, 2013). *RX63N Group, RX631 Group User's Manual: Hardware, Rev 1.60.*

[2] Koopman, Philip J. (2010). *Better Embedded System Software.* New Castle, PA.: Drumnadrochit Education.

Renesas Electronics America Inc.