3.2 Spiral model

- The *spiral model*, originally proposed by Boehm, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model.
- It provides the potential for rapid development of incremental versions of the software.
- Using the spiral model, software is developed in a series of incremental releases.
- During early iterations, the incremental release might be a paper model or prototype.
- During later iterations, increasingly more complete versions of the engineered system are produced.
- A spiral model is divided into a number of framework activities, also called ***task regions***.
- Typically, there are between three and six task regions.
- Following figure depicts a spiral model that contains six task regions:

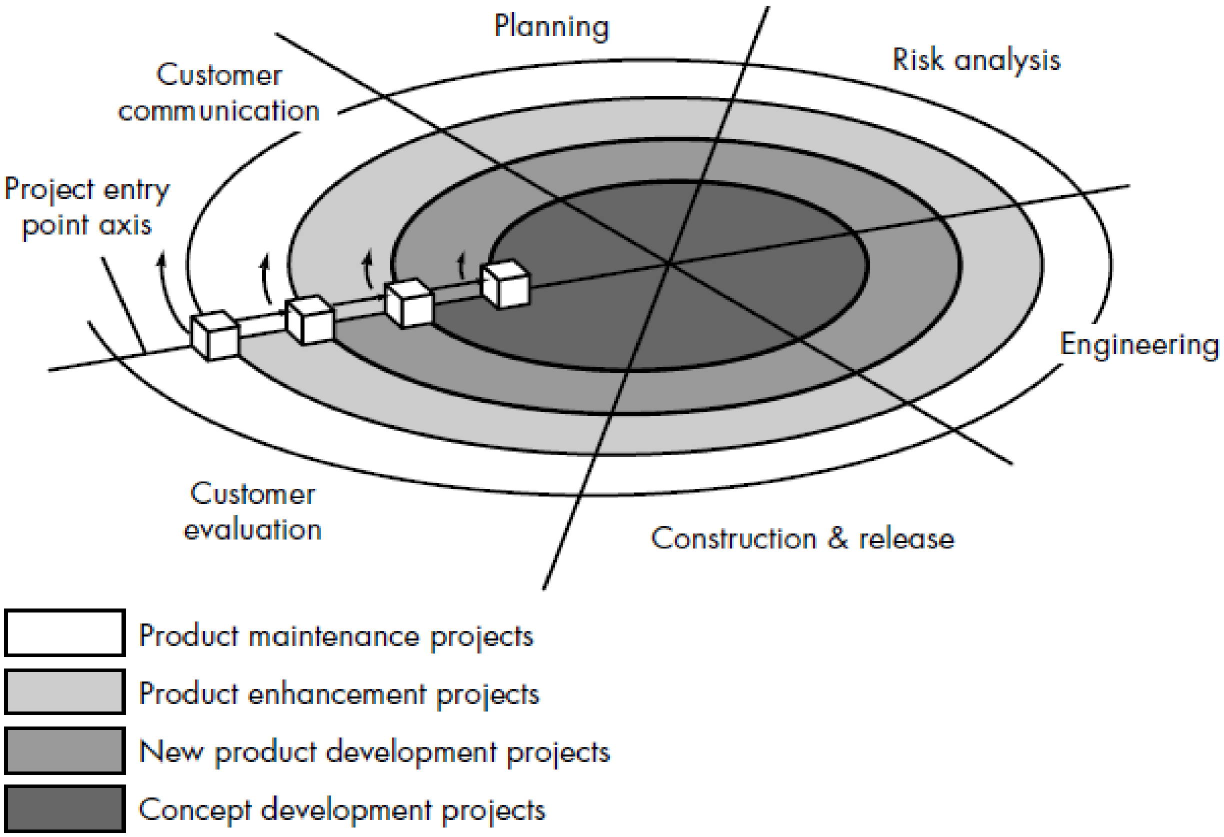


Figure: Spiral Model

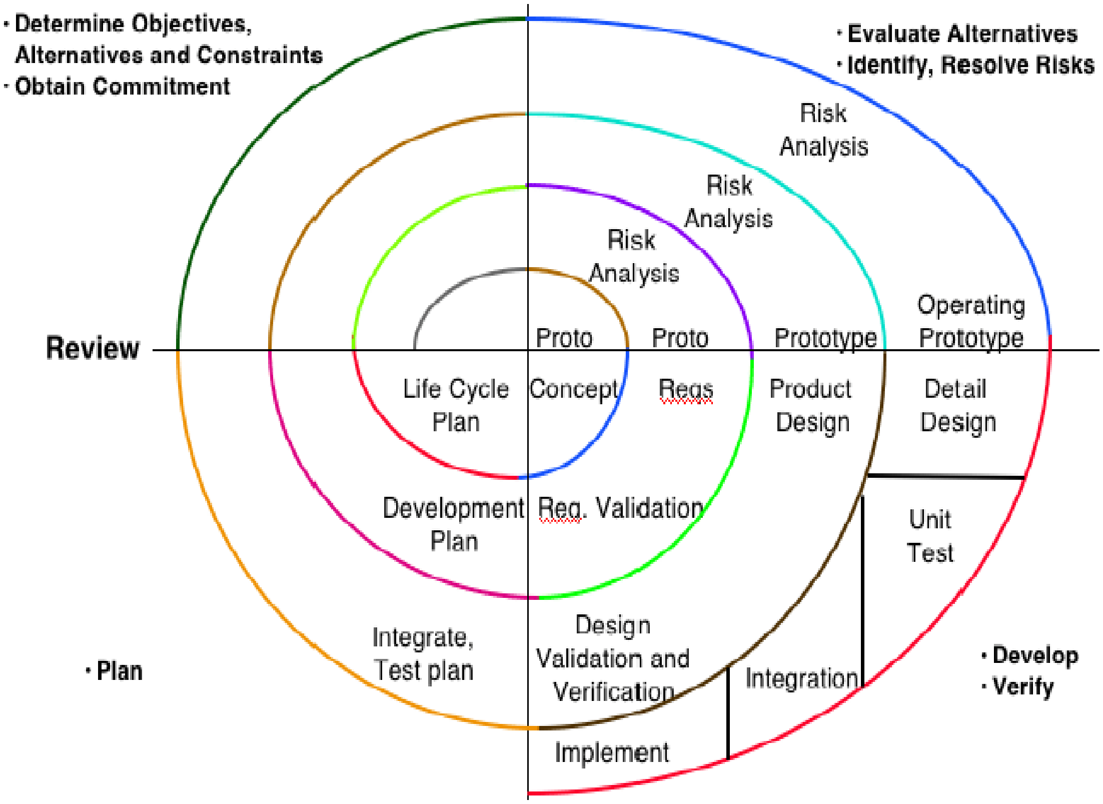
1. **Customer communication**—tasks required to establish effective communication between developer and customer.
2. **Planning**—tasks required to define resources, timelines, and other project related information.
3. **Risk analysis**—tasks required to assess both technical and management risks.
4. **Engineering**—tasks required to build one or more representations of the application.
5. **Construction and release**—tasks required to construct, test, install, and provide user support (e.g., documentation and training).
6. **Customer evaluation**—tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

- Each of the regions is populated by a set of work tasks, called a ***task set***.

- For small projects, the number of work tasks and their formality is low.
- For larger, more critical projects, each task region contains more work tasks that are defined to achieve a higher level of formality.
- As this evolutionary process begins, the software engineering team moves around the spiral in a clockwise direction, beginning at the center.
- The first circuit around the spiral might result in the development of a product specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software.
- Each pass through the planning region results in adjustments to the project plan.
- Cost and schedule are adjusted based on feedback derived from customer evaluation.
- In addition, the project manager adjusts the planned number of iterations required to complete the software.
- The spiral model is a realistic approach to the development of large-scale systems and software.
- Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level.
- The spiral model uses prototyping as a risk reduction mechanism but, more important, enables the developer to apply the prototyping approach at any stage in the evolution of the product.
- It maintains the systematic stepwise approach suggested by the classic life cycle but incorporates it into an iterative framework that more realistically reflects the real world.
- The spiral model demands a direct consideration of technical risks at all stages of the project and, if properly applied, should reduce risks before they become problematic.

The development spiral consists of four quadrants as shown in the figure below:

Quadrant 1: Determine objectives, alternatives, and constraints.
Quadrant 2: Evaluate alternatives, identify, resolve risks.
Quadrant 3: Develop, verify, next-level product.
Quadrant 4: Plan next phases.



Although the spiral, as depicted, is oriented toward software development, the concept is equally applicable to systems, hardware, and training.

To better understand the scope of each spiral development quadrant, let’s briefly address each one.

Quadrant 1: Determine Objectives, Alternatives, and constraints

Activities performed in this quadrant include:

- Establish an understanding of the system or product objectives—namely performance, functionality, and ability to accommodate change.
- Investigate implementation alternatives—namely design, reuse, procure, and procure/ modify.
- Investigate constraints imposed on the alternatives—
  - technology,
  - cost,
  - schedule,
  - support, and
  - risk.
- Once the system or product’s objectives, alternatives, and constraints are understood, Quadrant 2 (Evaluate alternatives, identify, and resolve risks) is performed.

Quadrant 2: Evaluate Alternatives, Identify, Resolve Risks

- Engineering activities performed in this quadrant select an alternative approach that best satisfies technical, technology, cost, schedule, support, and risk constraints.
- The focus here is on **risk mitigation**.
- Each alternative is investigated and prototyped to reduce the risk associated with the development decisions. Boehm describes these activities as follows:
  - Prototyping,
  - simulation,
  - benchmarking,
  - reference checking,
  - administering use questionnaires,
  - analytic modelling, or
  - combinations of these and
  - other risk resolution techniques.
- The outcome of the evaluation determines the next course of action.
- If critical operational and/or technical issues such as performance and interoperability (i.e., external and internal) risks remain, more detailed prototyping may need to be added before progressing to the next quadrant.

Quadrant 3: Develop, Verify, Next-Level Product

- If a determination is made that the previous prototyping efforts have resolved the issues, activities to develop, verify, next-level product are performed.
- As a result, the basic “waterfall” approach may be employed—meaning concept of operations, design, development, integration, and test of the next system or product iteration.
- If appropriate, incremental development approaches may also be applicable.

Quadrant 4: Plan Next Phases

- The spiral development model has one characteristic that is common to all models—the need for advanced technical planning and multidisciplinary reviews at critical staging or control points.
- Each cycle of the model culminates with a technical review that assesses the status, progress, maturity, merits, risk, of development efforts to date; resolves critical operational and/or technical issues; and reviews plans and identifies critical issues to be resolved for the next iteration of the spiral.

- Subsequent implementations of the spiral may involve lower level spirals that follow the same quadrant paths and decision considerations.

Spiral Model Advantages:

- Repeated or continuous development helps in risk management.
- The developers or programmers describe the characteristics with high priority first and then develop a prototype based on these.
- This prototype is tested and desired changes are made in the new system.
- This continual and steady approach minimizes the risks or failure associated with the change in the system.
- Adaptability in the design of spiral model in software engineering accommodates any number of changes that may happen, during any phase of the project.
- Since the prototype building is done in small fragments or bits, cost estimation becomes easy and the customer can gain control on administration of the new system.
- As the model continues towards final phase, the customer's expertise on new system grows, enabling smooth development of the product meeting client's needs.

Spiral Model Disadvantages:

The following can be summarized as the disadvantages of the spiral model.

- Spiral models work best for large projects only, where the costs involved are much higher and system pre requisites involves higher level of complexity.
- Spiral model needs extensive skill in evaluating uncertainties or risks associated with the project and their abatement.
- Spiral models work on a protocol, which needs to be followed strictly for its smooth operation. Sometimes it becomes difficult to follow this protocol.
- Evaluating the risks involved in the project can shoot up the cost and it may be higher than the cost for building the system.
- There is a requirement for further explanation of the steps involved in the project such as breakthrough, blueprint, checkpoints and standard procedure.