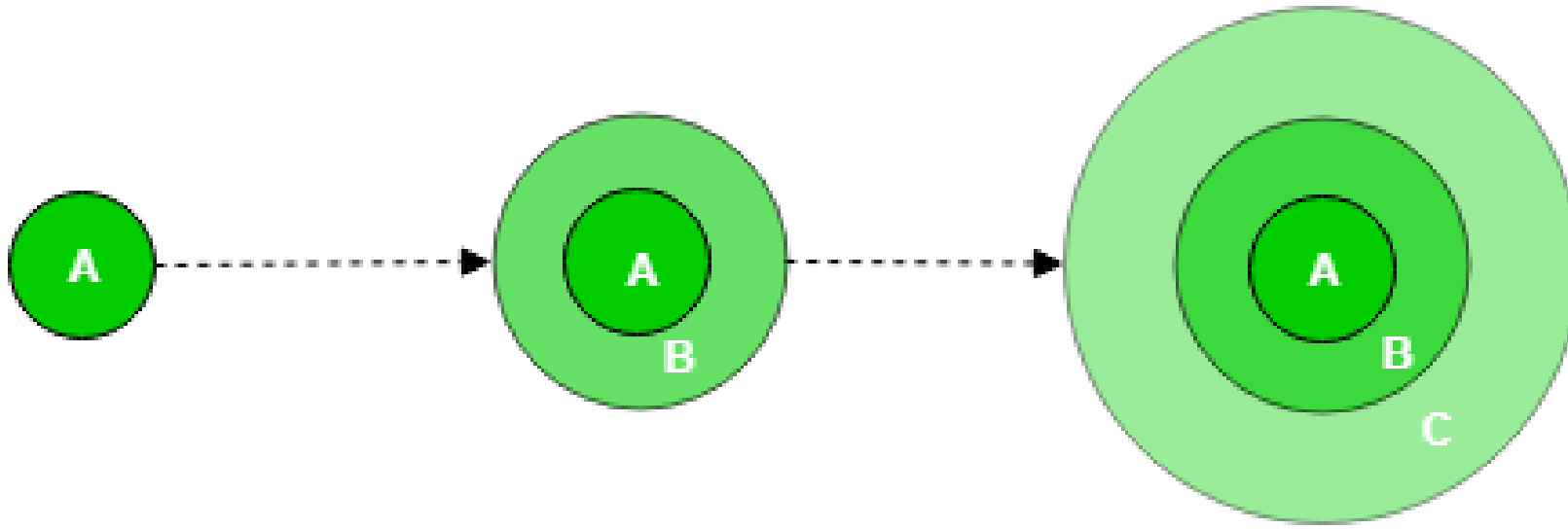


Software Process Models

Incremental Model

Incremental process model is also known as the Successive version model.

First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.

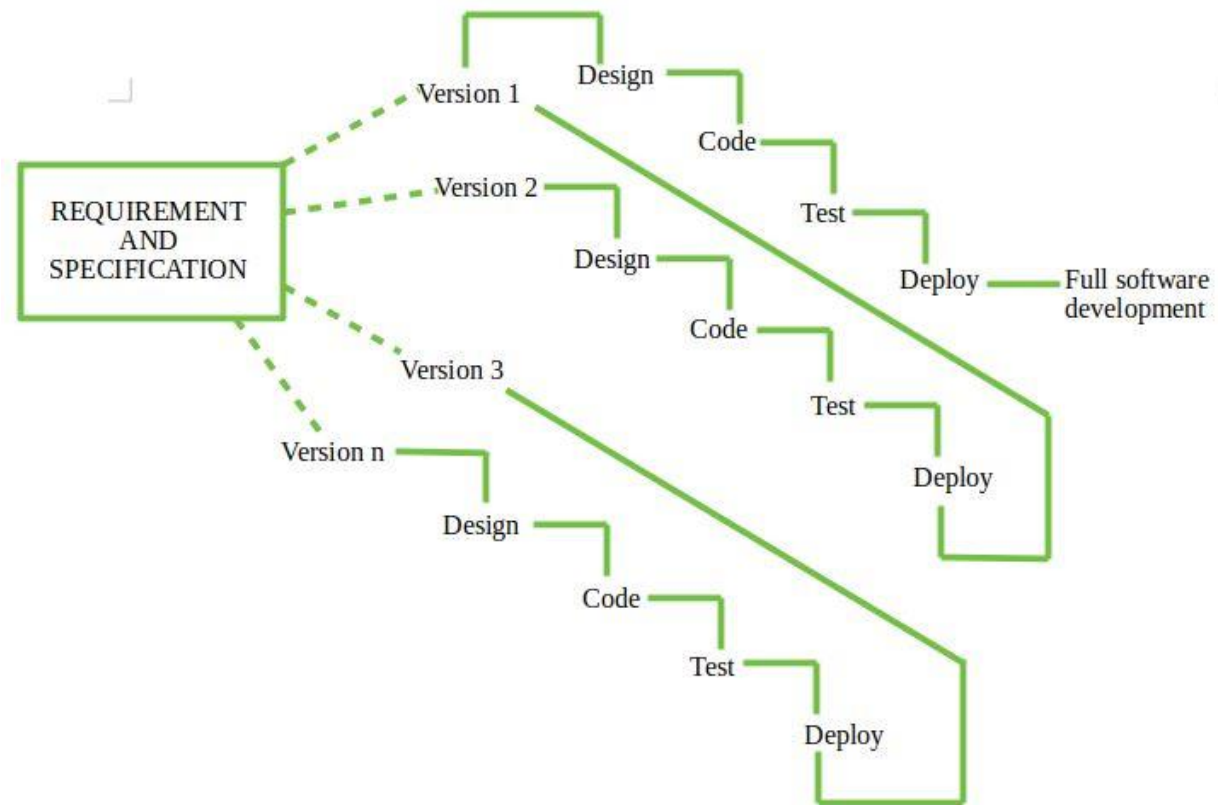


Life cycle activities

Requirements of Software are first broken down into several modules that can be incrementally constructed and delivered. At any time, the plan is made just for the next increment and not for any kind of long term plan. Therefore, it is easier to modify the version as per the need of the customer. The development Team first undertakes to develop core features (these do not need services from other features) of the system.

Once the core features are fully developed, then these are refined to increase levels of capabilities by adding new functions in Successive versions. Each incremental version is usually developed using an iterative waterfall model of development.

As each successive version of the software is constructed and delivered, now the feedback of the Customer is to be taken and these were then incorporated into the next version. Each version of the software has more additional features than the previous ones.



Phases of Incremental Model

Incremental Phases	Activities performed in incremental phases
Requirement Analysis	<ul style="list-style-type: none">•Requirement and specification of the software are collected
Design	<ul style="list-style-type: none">•Some high-end function are designed during this stage
Code	<ul style="list-style-type: none">•Coding of software is done during this stage
Test	<ul style="list-style-type: none">•Once the system is deployed, it goes through the testing phase

When to use

The Requirements should be known clearly and understood, when there is a demand for the early release of the product is there, when there are high-risk features and requirement goals are present in the objective of the software.

This kind of methodologies are mainly followed by-product based companies as the defects risk in the developed software are quite minimum and also used in developing software in web applications.

This model is also preferred when the project has lengthy development schedules.

Also if the development is adopting new technology in the software development then also this method is preferred as the developers are new to the technology

Advantages of the Incremental Model

Since the object will be divided into incremental stages, it will be made sure that 100% requirements are achieved and 100% objective of the software.

Since there is testing at each incremental phase there will be multiple testing for the software and more the testing better the result and fewer defects.

By adopting this approach we can lower the initial delivery cost.

This model is flexible and incurs least cost when there is a change in the requirement or the scope.

The user or the customer can provide feedback on each stage so work effort will be valued and sudden changes in the requirement can be prevented.

Compared to the other model this model is tend to be cheaper on the pockets of the user.

By following this models errors can be identified quiet easily

Disadvantages

- Needs good planning and design.
- Need a good Analysis.
- Problems might be caused due to system's architecture.
- Every iteration phase is rigid and it doesn't overlap with each other.
- Solving a problem in one unit would require correction in all the units and would consume lot of time.
- Modules should be clearly defined.
- Well-defined module interfaces are required.
- Total cost is much higher than the Waterfall model.
- Need a complete and clear understanding of the whole system before it broken down and built incrementally.

Evolutionary Models

Software system evolves over time as requirements often change as development proceeds. Thus, a straight line to a complete end product is not possible. However, a limited version must be delivered to meet competitive pressure.

Usually a set of core product or system requirements is well understood, but the details and extension have yet to be defined.

Cont...

You need a process model that has been explicitly designed to accommodate a product that evolved over time.

It is iterative that enables you to develop increasingly more complete version of the software.

Two types are introduced, namely

- Prototype Process Model
- Spiral Process Models.

Evolutionary Models: Prototyping

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements.

By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.

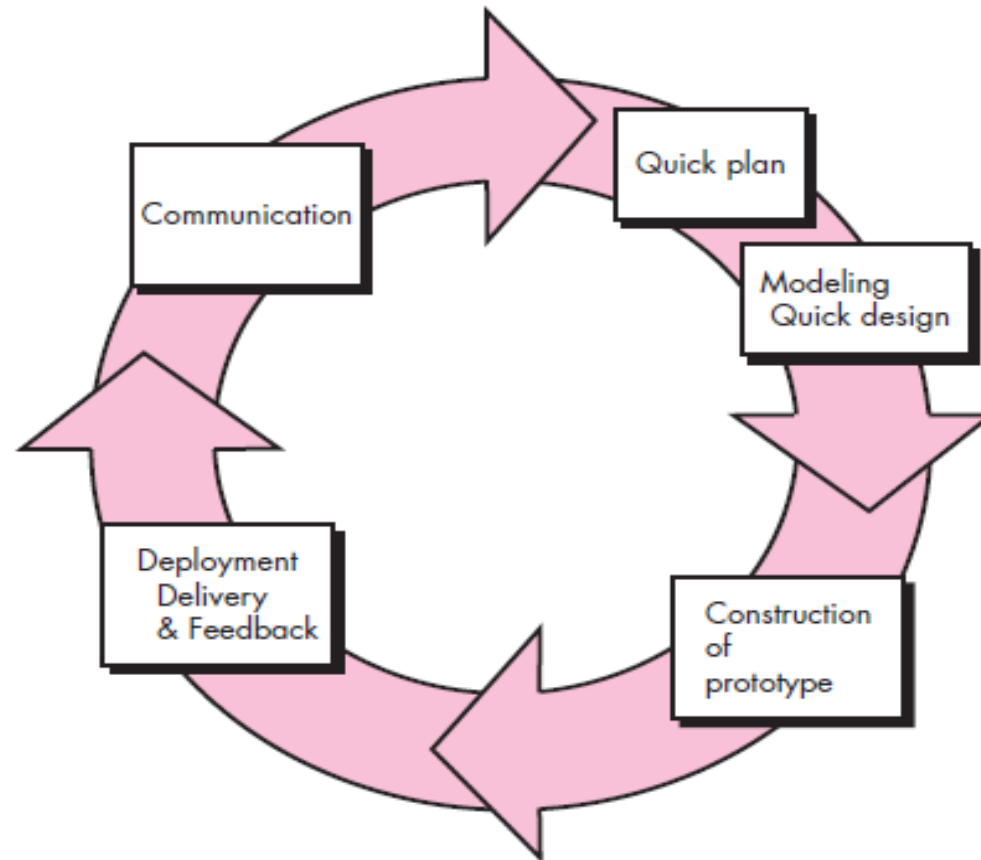
Prototype

The prototype are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

It is just the Dummy product, or a sample product just to show the customer that how actual product will look like.

Customer evaluate prototype to refine requirements

Prototype Model



When to use Prototype Model

Prototype model should be used when the desired system needs to have a lot of interaction with the end users.

Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

Advantages

Users are actively involved in the development

Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.

Errors can be detected much earlier.

Quicker user feedback is available leading to better solutions.

Missing functionality can be identified easily

Confusing or difficult functions can be identified

Requirements validation, Quick implementation of, incomplete, but functional, application.

Disadvantages

Leads to implementing and then repairing way of building systems.

Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

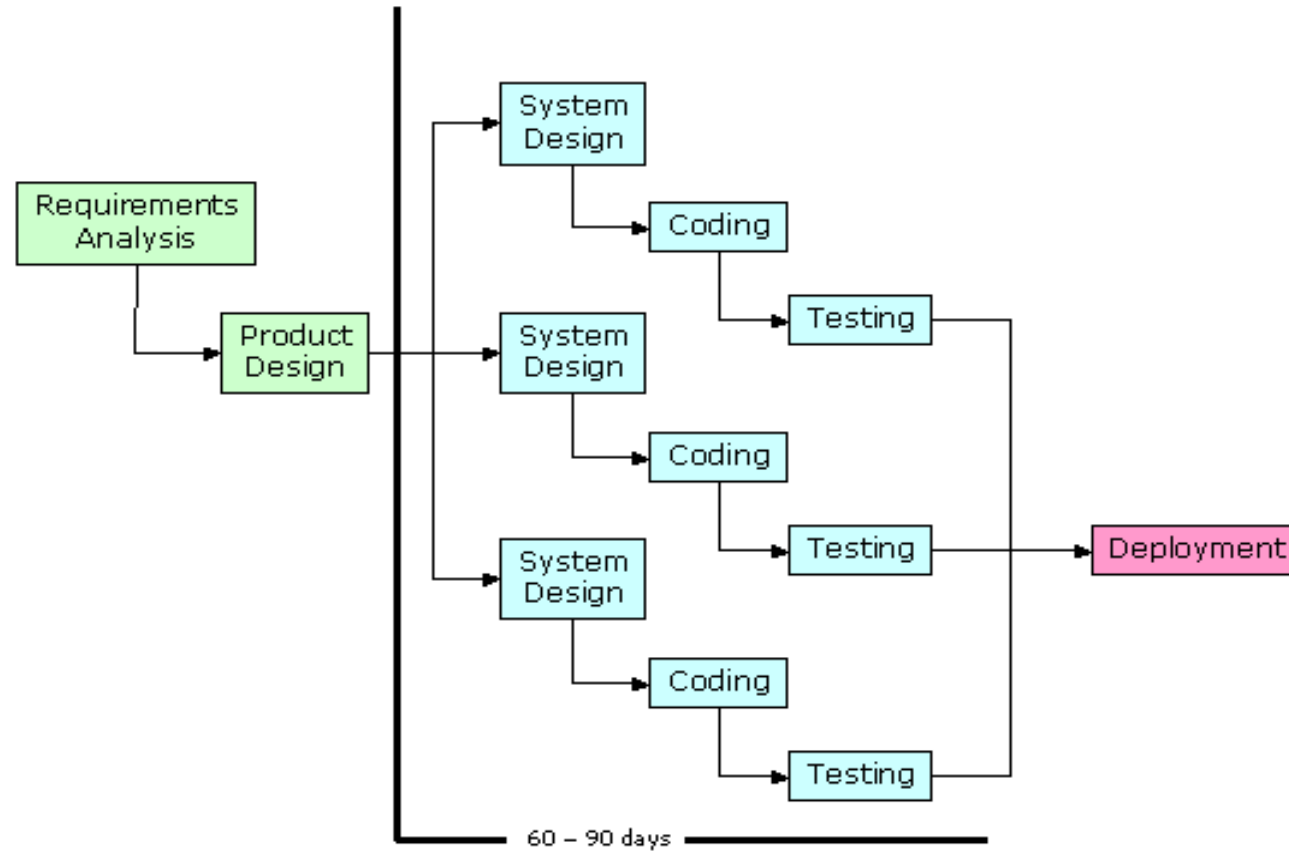
Incomplete application may cause application not to be used as the full system was designed

Incomplete or inadequate problem analysis.

Rapid Application Development Model

In RAD model the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. The functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery.

RAD Model



When to use RAD Model

RAD should be used when there is a need to develop fully functional system within very short time(60 to 90 days)

Whole System can be divided in to small modules

User want complete working software as soon as possible

RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

User is willing to pay more

Advantages

Changing requirements can be accommodated.

Progress can be measured.

Iteration time can be short with use of powerful RAD tools.

Productivity with fewer people in short time.

Reduced development time.

Increases reusability of components

Quick initial reviews occur

Encourages customer feedback

Integration from very beginning solves a lot of integration issues.

Disadvantages

Dependency on technically strong team members for identifying business requirements.

Only system that can be modularized can be built using RAD.

Requires highly skilled developers/designers.

High dependency on modeling skills.

Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

Management complexity is more.

Suitable for systems that are component based and scalable.

Requires user involvement throughout the life cycle.

Suitable for project requiring shorter development times.

Spiral Model

A heavy-weight, plan-driven, highly-structured approach for large projects.

Especially designed for those with higher chances of failure.

Combines iterative model, emphasizes risk assessment, customer participation, prototyping, and more.

Definitely an iterative process.

Spiral Model

Spiral model has different phases depending upon the choice that team make.

Communication

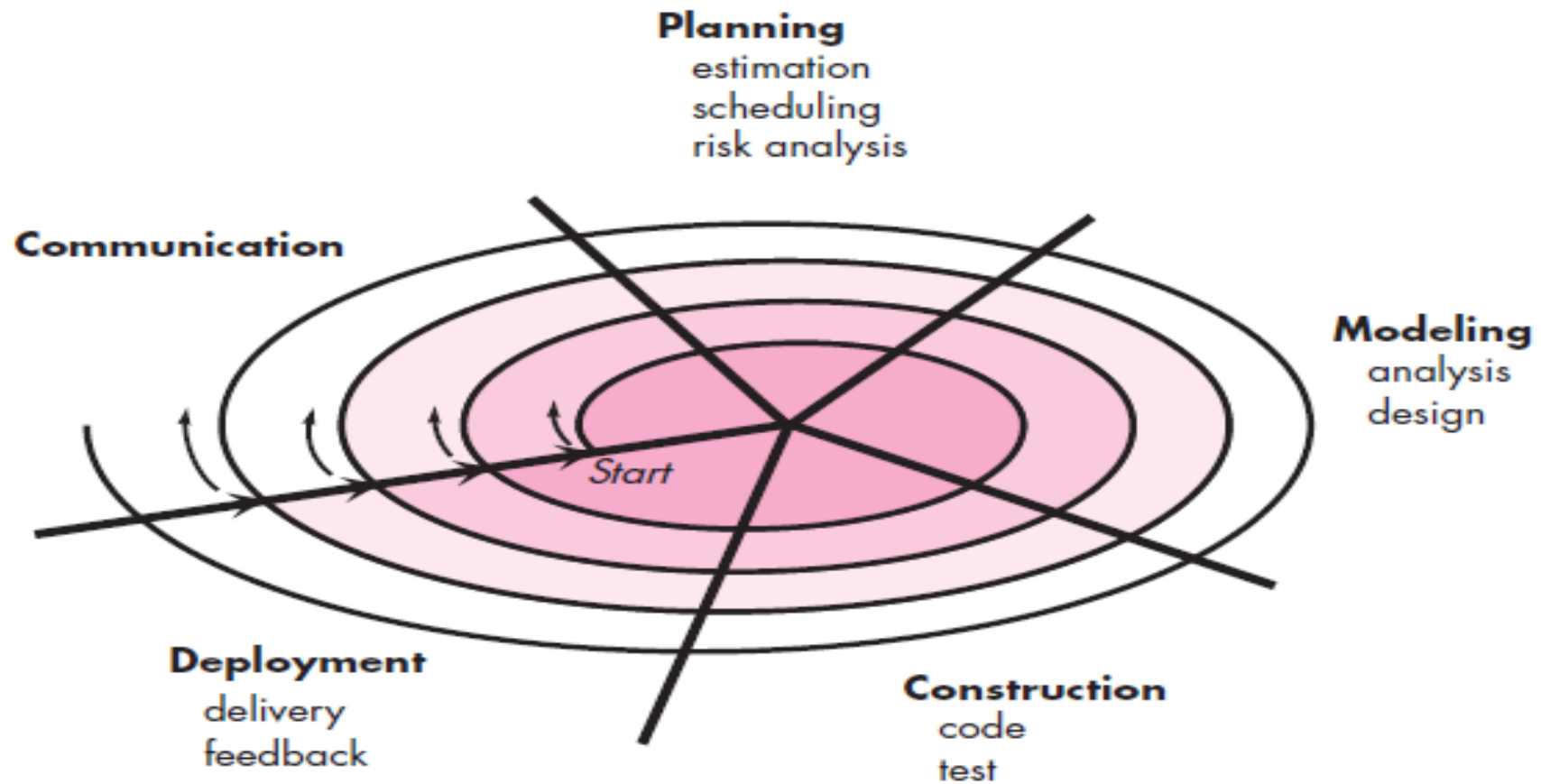
Planning

Modeling

Construction

Deployment

Spiral Model



Spiral Model

Planning Phase: Requirements are gathered during the planning phase. Requirements like 'BRS' that is 'Business Requirement Specifications' and 'SRS' that is 'System Requirement specifications'.

Risk Analysis: In this phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.

Spiral Model

Construction Phase: In this phase software is developed, along with testing at the end of the phase. Hence in this phase the development and testing is done.

Deployment phase: This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

When to use Spiral Model

When costs and risk evaluation is important

For medium to high-risk projects

Long-term project commitment unwise because of potential changes to economic priorities

Users are unsure of their needs

Requirements are complex

New product line

Significant changes are expected (research and exploration)

Advantages of Spiral Model

High amount of risk analysis hence, avoidance of Risk is enhanced.

Good for large and mission-critical projects.

Strong approval and documentation control.

Additional Functionality can be added at a later date.

Software is produced early in the software life cycle.

Disadvantages Spiral Model

Can be a costly model to use.

Risk analysis requires highly specific expertise.

Project's success is highly dependent on the risk analysis phase.

Doesn't work well for smaller projects.