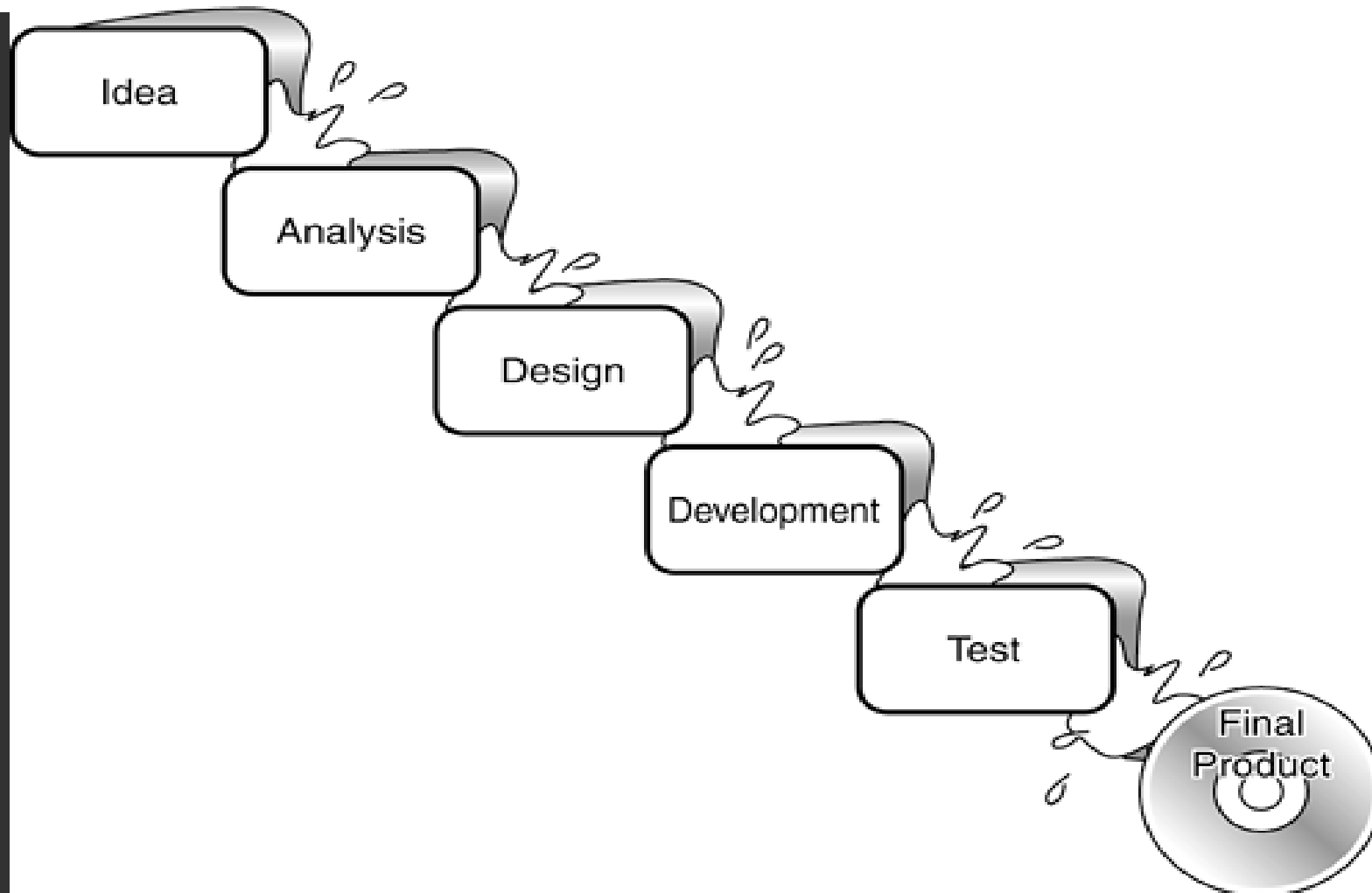


SOFTWARE PROCESS MODEL

WATERFALL MODEL

- It is also referred to as a **linear-sequential life cycle model**.
- Each phase must be completed fully before the next phase can begin.
- Phases do not overlap
- Primitive software development methodology
- Each phase must be completed properly before moving to another phase.
- After a number of iterations, phases of the life cycle (such as specification and design) are “frozen”

WATERFALL MODEL



FEATURES OF WATERFALL MODEL

- A waterfall model is easy to follow.
- It can be implemented for any size project.
- Every stage has to be done separately at the right time so you cannot jump stages.
- Documentation is produced at every stage of a waterfall model allowing people to understand what has been done.
- Testing is done at every stage.

ADVANTAGES OF A WATERFALL MODEL

- A waterfall model helps find problems earlier on which can cost a business less than if it was found later on.
- Requirements will be set and these wouldn't be changed.
- As everything is documented a new team member can easily understand what's to be done.
- Implementers have to follow the design accurately

DISADVANTAGES OF A WATERFALL MODEL

- If requirements may change the Waterfall model may not work.
- Many believe it is impossible to make one stage of the projects life cycle perfect.
- Difficult to estimate time and cost for each stage of the development process.
- Constant testing of the design is needed.

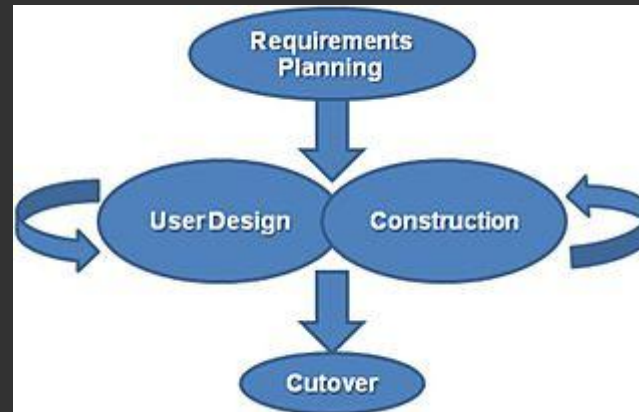
APPLICABILITY

- When requirements are well known and few changes are likely to be needed
- Can be used also for parts of larger software systems

RAPID APPLICATION DEVELOPMENT

- It is a software development methodology that uses minimal planning in favor of rapid prototyping.
- Generally allows software to be written much faster, and makes it easier to change requirements.
- Designed to produce useful software quickly
- It is a type of incremental model.
- The components or functions are developed in parallel as if they were mini projects.
- It is an object-oriented approach to systems development that includes a method of development as well as software tools

RAPID APPLICATION DEVELOPMENT



PHASES

- **Requirements Planning phase**

discuss and agree on business needs, project scope, constraints, and system requirements.

- **User design phase**

users interact with systems analysts and develop models and prototypes that represent all system processes, inputs, and outputs.

- **Construction phase**

focuses on program and application development task similar to the SDLC

- **Cutover phase**

resembles the final tasks in the SDLC implementation

ADVANTAGE

- Flexible and adaptable to changes
- Can handle large projects without a doubt
- RAD realizes an overall reduction in project risk
- RAD realizes an overall reduction in project risk.
- Generally RAD incorporates short development cycles

DISADVANTAGE

- Requires highly skilled developers/designers.
- Only system that can be modularized can be built using RAD
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.
- High dependency on modeling skills.
- Can't use for small projects
- Requires more resources and money to implement RAD
- All applications are not compatible with RAD
- Technical risks are high
- Need both customer and developer commitments to complete a project. Otherwise RAD will fail

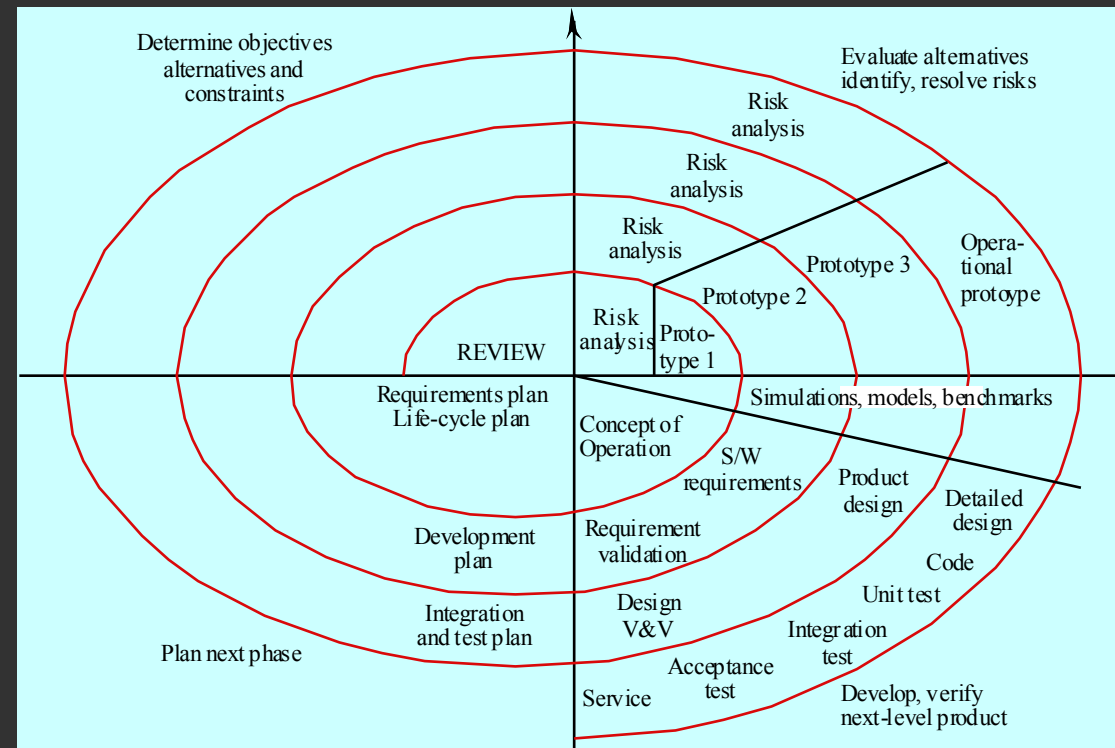
SPIRAL MODEL

- It is a type of iterative software development model which is generally implemented in high risk projects.
- In this system development method, we combine the features of both, waterfall model and prototype model.
- In Spiral model we can arrange all the activities in the form of a spiral.
- It is similar to the incremental model, with more emphasis placed on risk analysis.
- Each loop represent a development phase

SPIRAL MODEL

- There are four phases in Spiral Model :
 1. Planning
 2. Evaluation
 3. Risk Analysis &
 4. Engineering.

SPIRAL MODEL



ADVANTAGES

- 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

- 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.

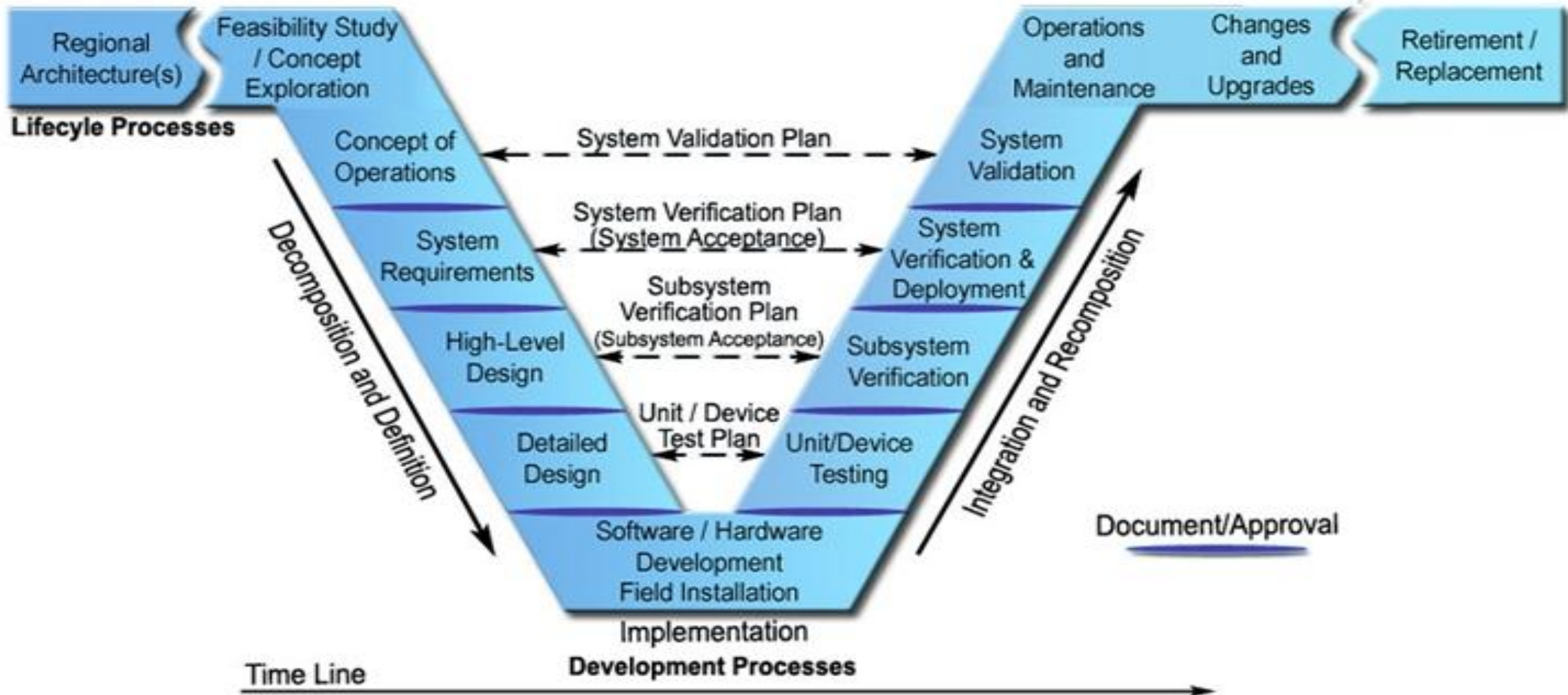
APPLICABILITY

- 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)

V- PROCESS MODEL

- V- model means Verification and Validation model.
- Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes.
- Each phase must be completed before the next phase begins.
- Testing of the product is planned in parallel with a corresponding phase of development.

V-PROCESS MODEL



PHASE OF V-PROCESS MODEL

The various phases of the V-model are as follows:

- **Requirements** : In this model before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in the requirements gathering.
- **The high-level design (HLD)** phase focuses on system architecture and design. It provide overview of solution, platform, system, product and service/process. An integration test plan is created in this phase as well in order to test the pieces of the software systems ability to work together.
- **The low-level design (LLD)** phase is where the actual software components are designed. It defines the actual logic for each and every component of the system. Class diagram with all the methods and relation between classes comes under LLD. Component tests are created in this phase as well.
- **The implementation** phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.
- **Coding**: This is at the bottom of the V-Shape model. Module design is converted into code by developers.

ADVANTAGES

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

DISADVANTAGES

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

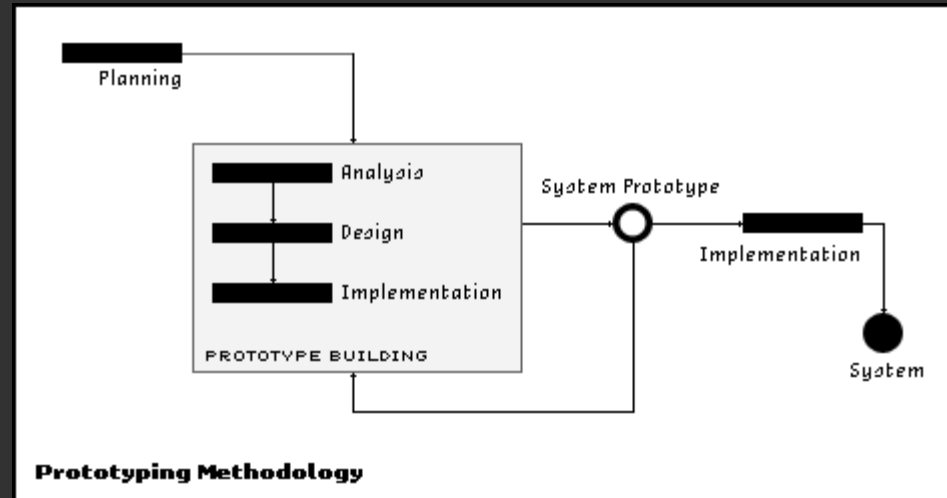
APPLICATION

- used for small to medium sized projects where requirements are clearly defined and fixed.
- should be chosen when ample technical resources are available with needed technical expertise.

PROTOTYPING MODEL

- Process of building a model of a actual system
- Are developed to understand the actual behavior of a proposed system
- Are a way to minimize the risk of project failure
- Prototypes are useful in seeking user reactions, suggestions, innovations, and revision plans
- Is an information-gathering technique

PROTOTYPING MODEL



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

- Insufficient analysis
- User confusion of prototype and finished system
- Developer misunderstanding of user objectives
- Excessive development time of the prototype
- Expense of implementing prototyping

APPLICATION AREAS

- very effective in the analysis and design of on-line systems.
- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- 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.

ANY QUESTIONS???

THANK YOU...