Tribhuvan University

(Faculty of Humanities and Social Science)

Bachelor in Computer Application (BCA)

Project: Presentation on Software Development Methodologies. (RAD, Spiral, Prototype)

Submitted to:

Sunil Sharma
Department of Software Engineering.
Asst. Director @ BCA Program, MMC, Bhadrapur, Jhapa.

© Group – D

BCA Fourth 2019 (2076)
Mechi Multiple Campus
Bhadrapur, Jhapa.

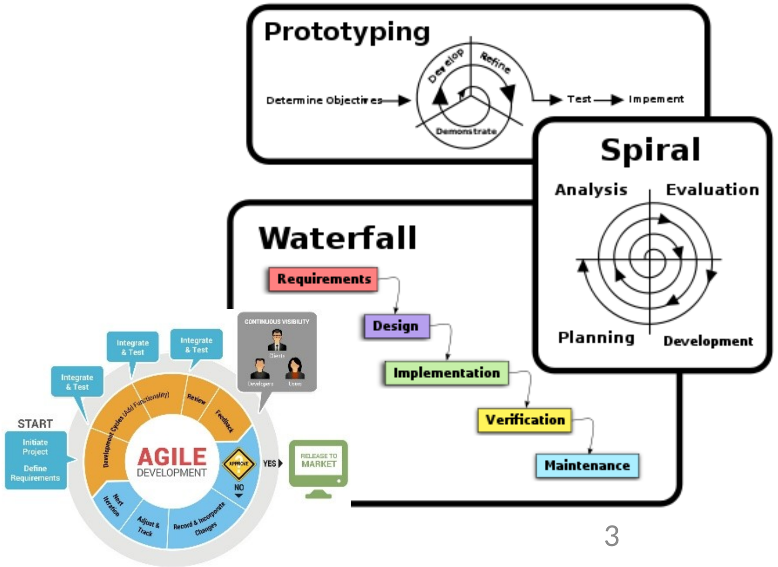
Submitted on: 06/25/2019

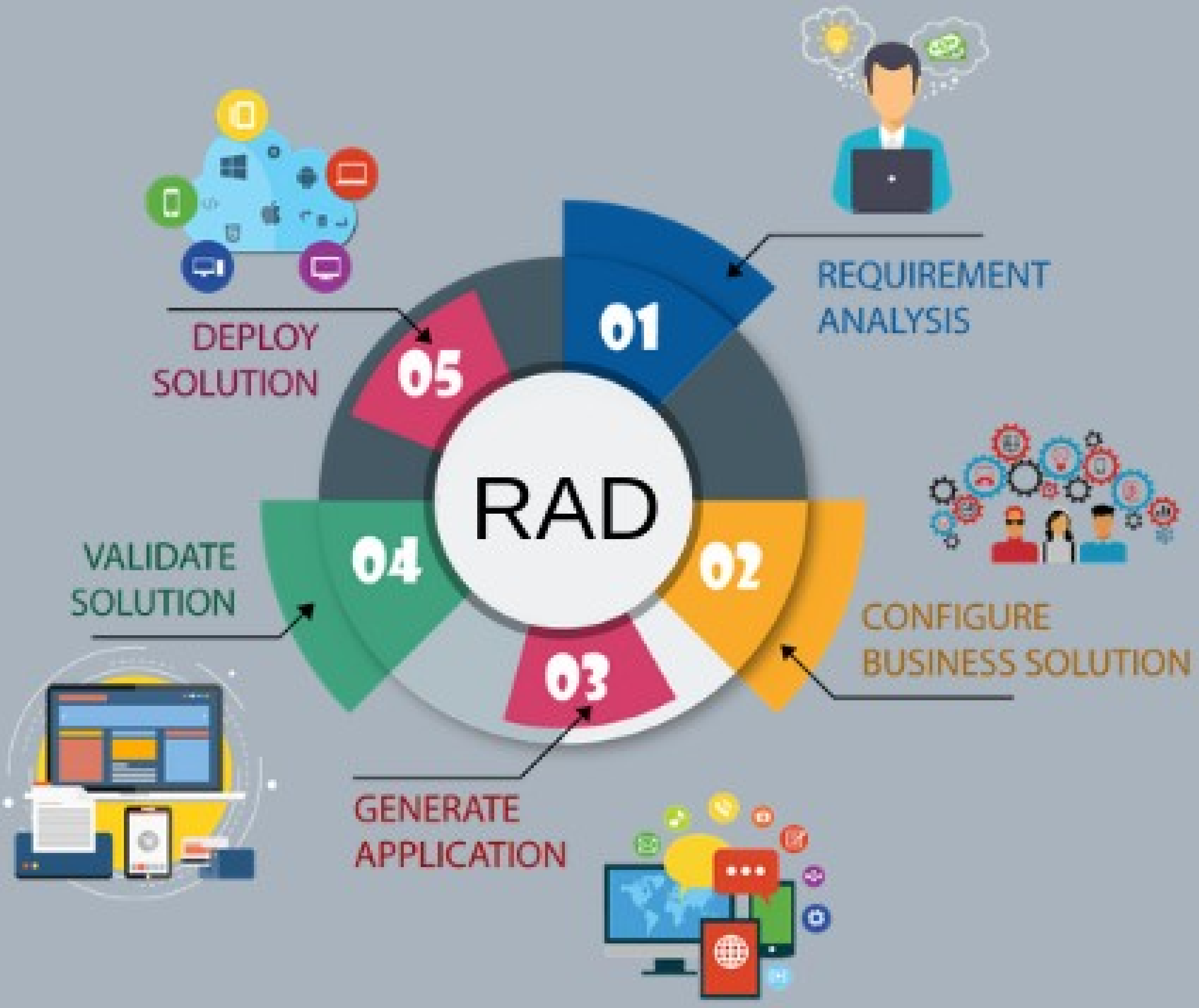
Group Members:
(alphabetically)

- Arpan Ojha
- Gaurav Nyaupane (Team Leader)
- Girija Prasad Bhattarai
- Nischal Khatiwada

Software Development Methodology:

- Formal Set Of Steps To Follow For Creating Software.
- Set Of Rules Or Practices Developer Have To Follow In Order To Create Quality Software.
- A Set Of Conventions The Software Industries Decides To follow.
- A Systematic Approach For Engineering Or Organizing Software Projects.
- Guidelines Or Procedure Pattern, Which Is Supposed To Be Followed In Entire Software Project To Achieve High Quality Outcome.

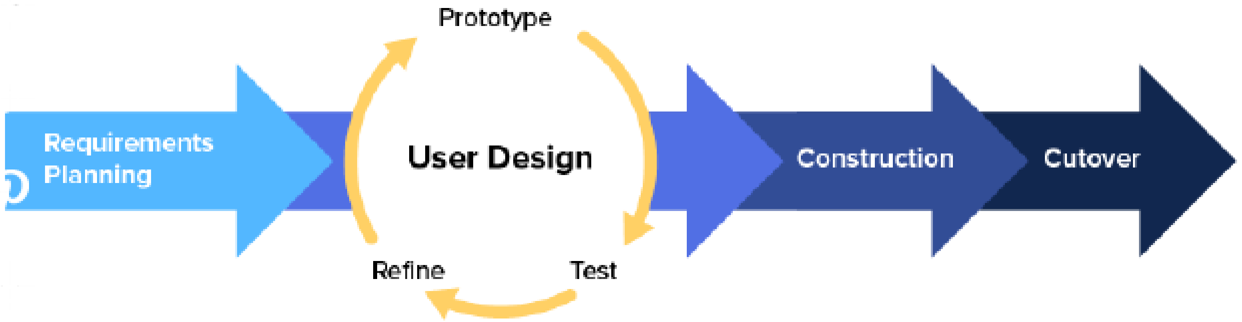




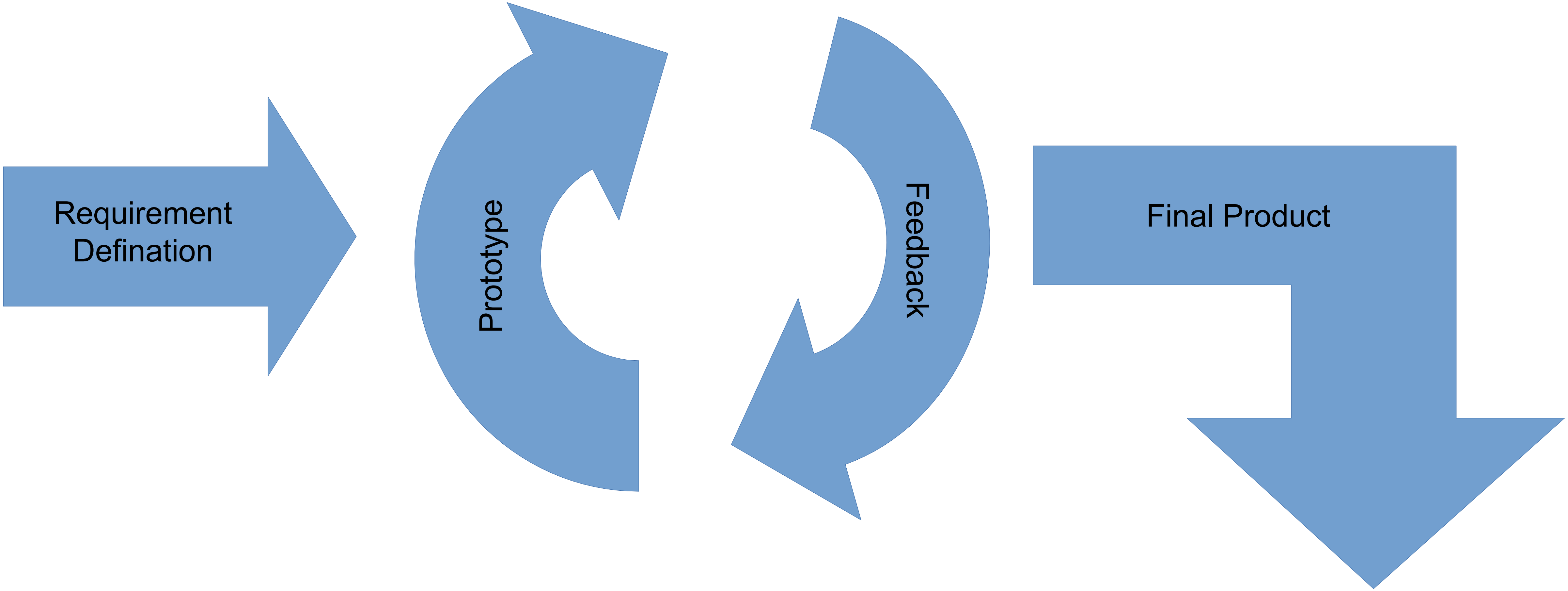
Rapid Application Development Model (RAD)

Rapid Application Development:

- **History:** RAD was response to traditional plan driven waterfall model which was based on traditional engineering approach of bridge and building construction. Developed in the 1970s and 1980s such as Structured System Analysis And Design Method (SSADM).
- **Base Principle:** RAD recognize that the software development is a knowledge intensive process and it provides flexible processes that help taking advantage of knowledge gained during the project to improve or adapt the solution.
- **Development of RAD:**
  - First RAD model was developed by Barry Boehm and was known as Spiral Model.
  - Barry Boehm, James Martin and other subsequent RAD approaches emphasized developing prototype.

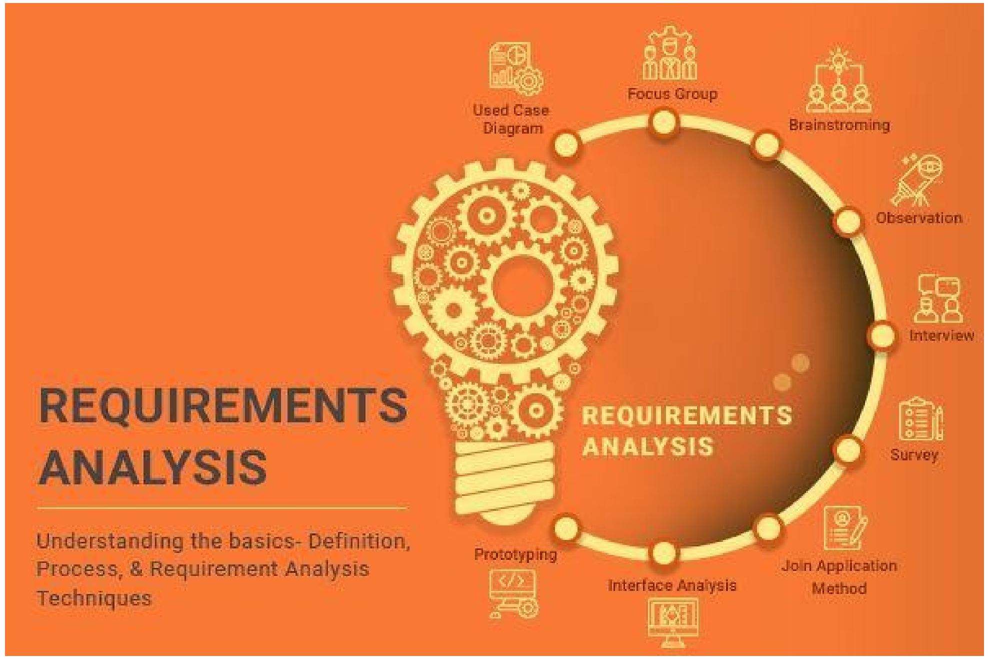


Steps in RAD: (Although RAD has changed over the years below are The Basic Four Steps.)



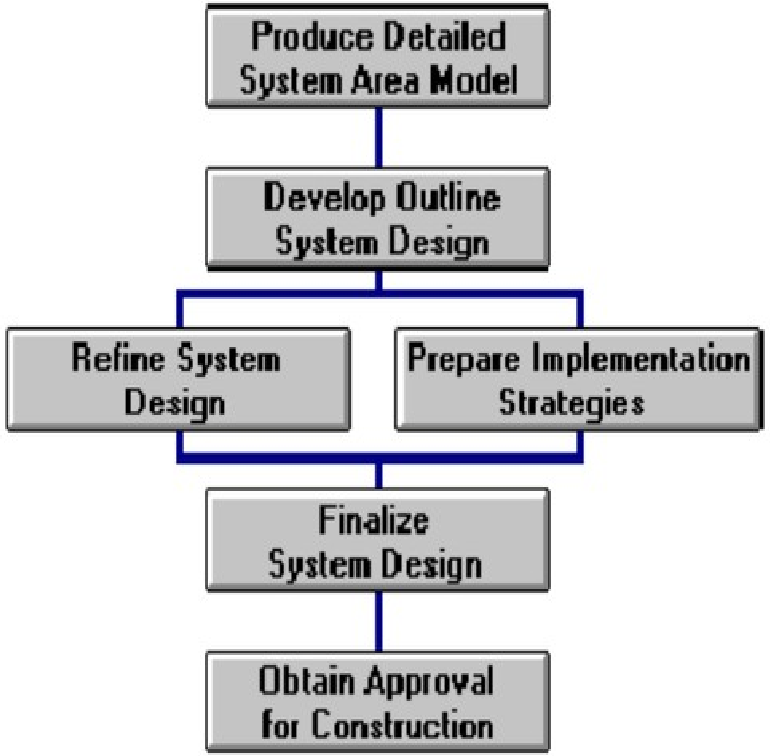
Define The Requirement:

- Users, Managers and IT staff agree upon business needs.
- Project’s scope and system requirements are gathered.
- RAD doesn’t require developers to sit with end users and get detailed list of specification.
- **Gathering of a broad requirement is enough.**
- Broad nature of the requirements helps developer to give specific requirement at different point of the development cycle.
- Takes one or four weeks to complete.
- Some tasks done in this phase are:
  - Research Current Situation
  - Define Requirements(Group Elicitation)
  - Finalize Requirements



Prototype:

- This is where the actual development take place.
- Developers are not forced to follow strict set of requirements.
- Developers create prototypes with different features and functions **as fast as they can**.
- Prototypes are then shown to the clients who decide what they like and what they don’t.
- **RAD Design Workshop** is conducted.
- Programming, application development, coding, unit integration and system testing is done in this phase.


Produce Detailed System Area Model
Develop Outline System Design
Refine System Design
Prepare Implementation Strategies
Finalize System Design
Obtain Approval for Construction

Receive Feedback:

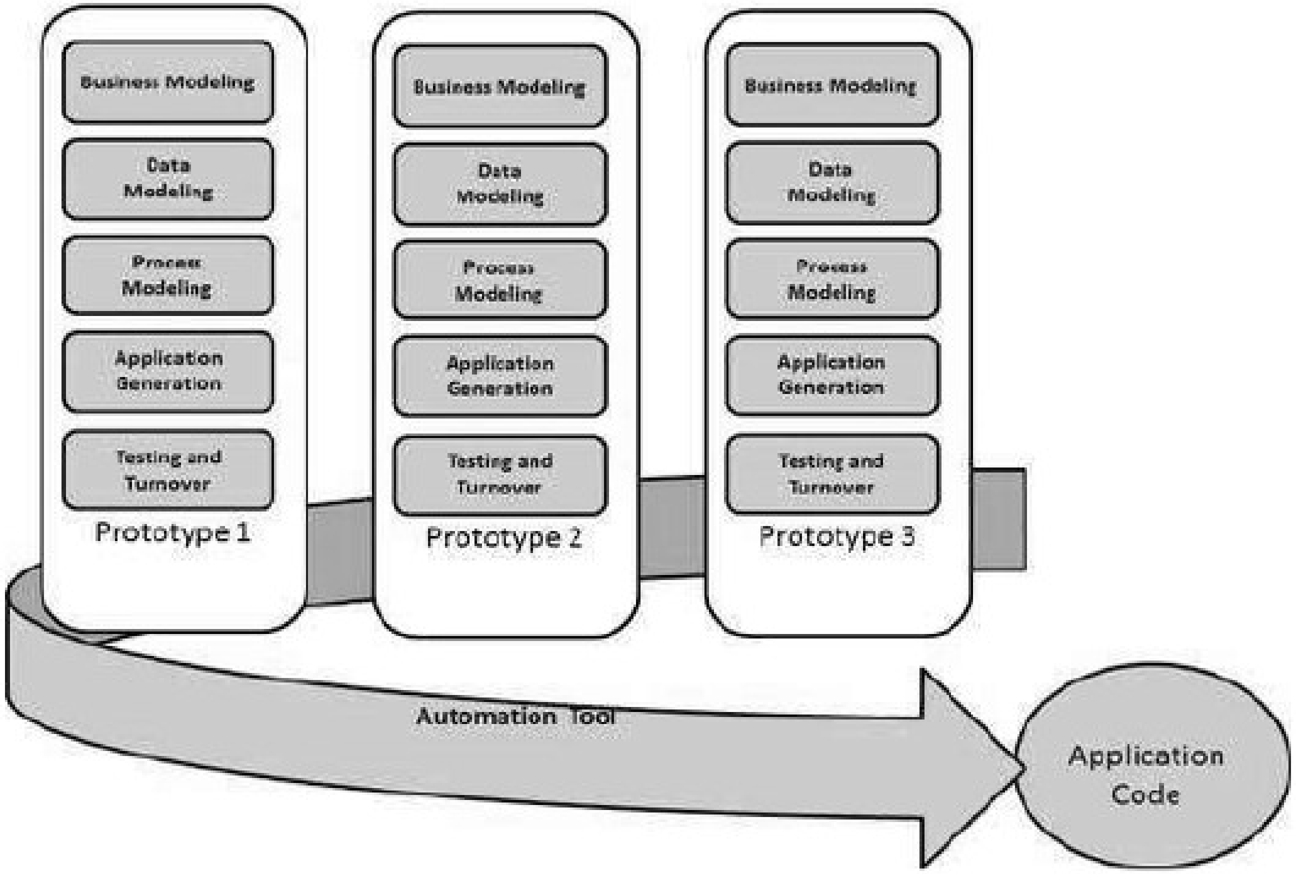
- In this stage feedback on what's good, what's not, what works and what doesn't is shared.
- Feedback about functionality, visuals and interfaces are gathered.
- With this feedback in mind prototyping continues.
- **Prototyping and Feedback are repeated until a final product which fit both developer's and client's requirement is realized.**
- **RAD Design Workshop** is conducted, Where users respond to actual working prototypes and analysts refine designed modules.
- JAD session can also be conducted.

Finalize Software:

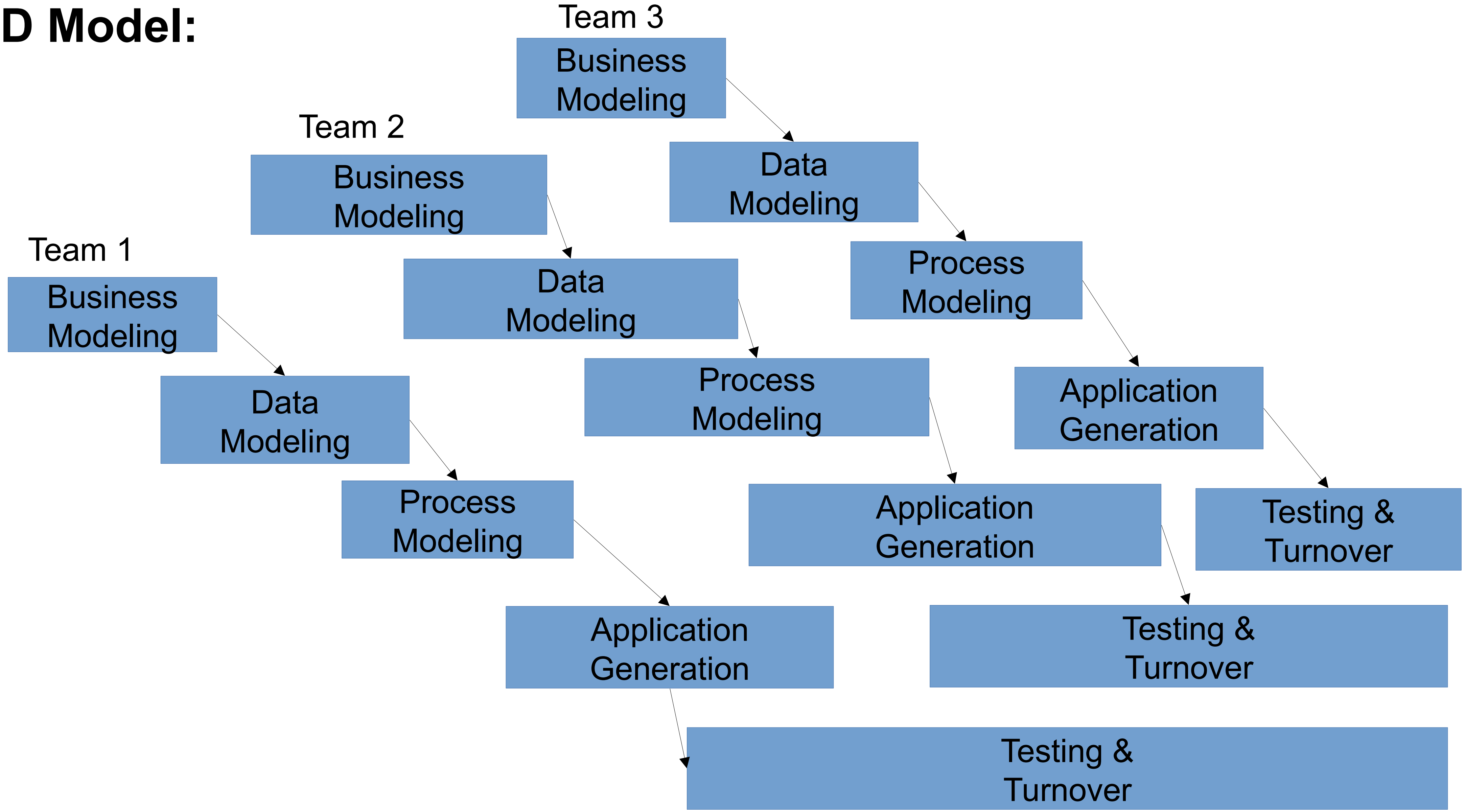
- Features, functions and interface of the software are finalized with the client.
- Stability, usability and maintainability are verified.
- Contract is formally ended here.
- Product is delivered to client.

What Is Rapid Application Development?

- RAD is a form of Agile Software Development Methodology.
- RAD is a S/W Development Methodology, that uses minimum planning in favour of rapid prototyping.
- Since there is no detailed pre-planning, it makes it easier to incorporate.
- RAD follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working on prototype.
- The prototype developed in RAD are reusable, which is main reason for success of this model.



RAD Model:

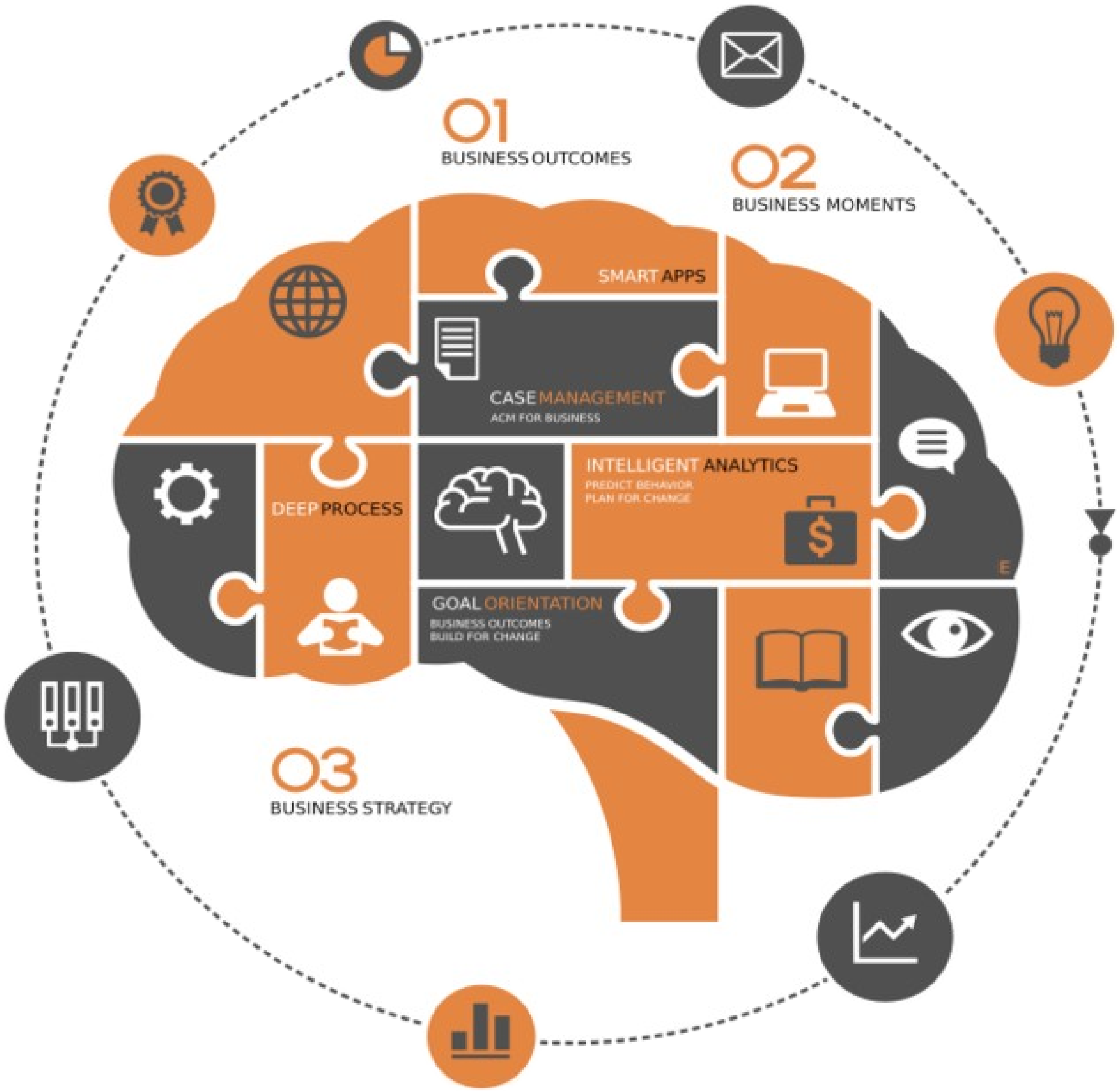

Team 3
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing & Turnover
Team 2
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing & Turnover
Team 1
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing & Turnover

Models in RAD.

1. **<u>Business Modeling:</u>**

- The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels.

- A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.



Models in RAD.

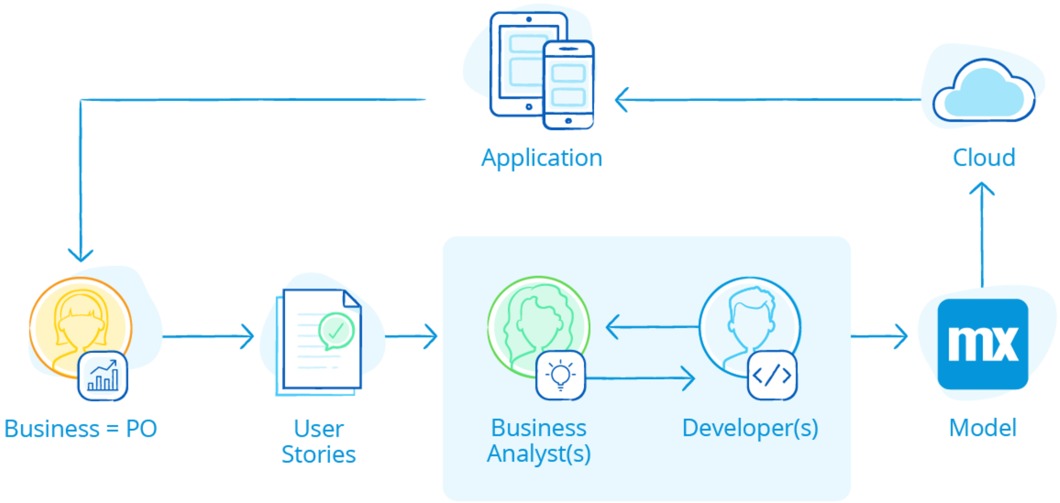
2. **Data Modeling:**

- The information gathered in the Business Modeling phase is reviewed and analyzed to form sets of data objects vital for the business.
- The attributes of all data sets is identified and defined.
- The relation between these data objects are established and defined in detail in relevance to the business model.

Models in RAD.

3. Process Modeling:

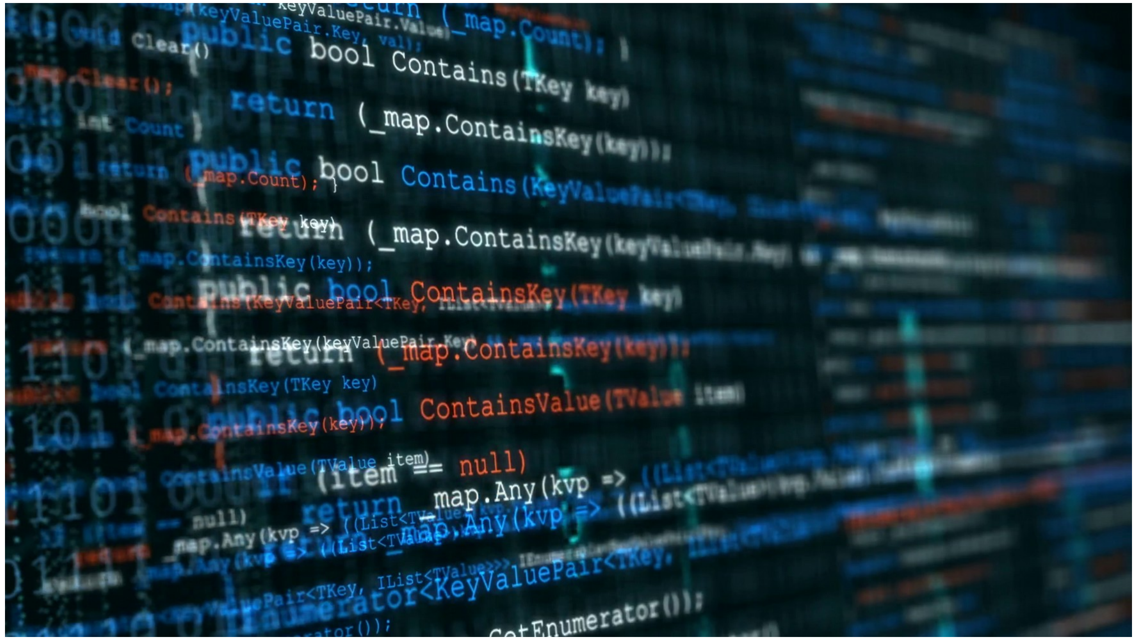
- The data object sets defined in the Data Modeling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model.
- The process model for any changes or enhancements to the data object sets is defined in this phase.
- Process descriptions for adding, deleting, retrieving or modifying a data object are given.



Models in RAD.

4. <u>Application Generation:</u>

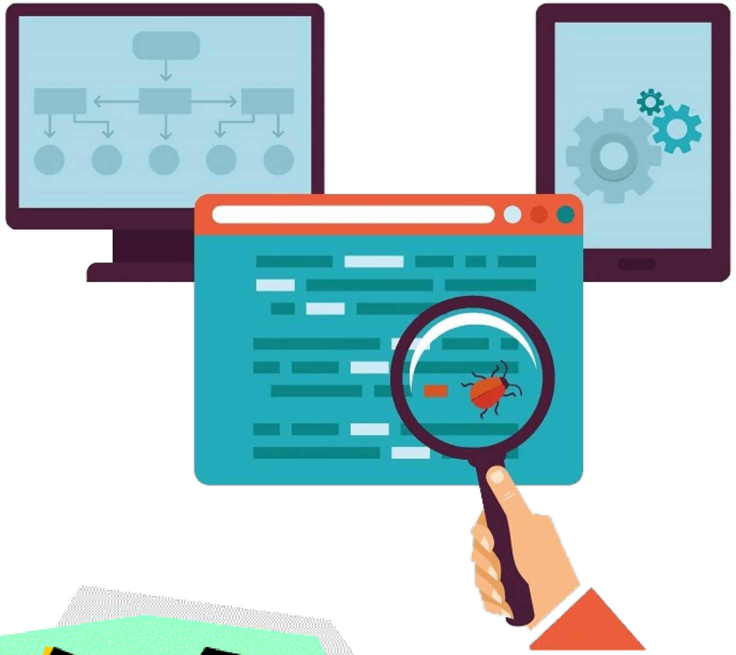
- The actual system is built in this phase.
- Actual coding is done.
- Development is done by using automation tools to convert process and data models into actual prototypes.
- First working prototype is developed(if it is first iteration) or additional features will be added to previous one.
- Once ready, the prototype is further processed to testing and after turnover.

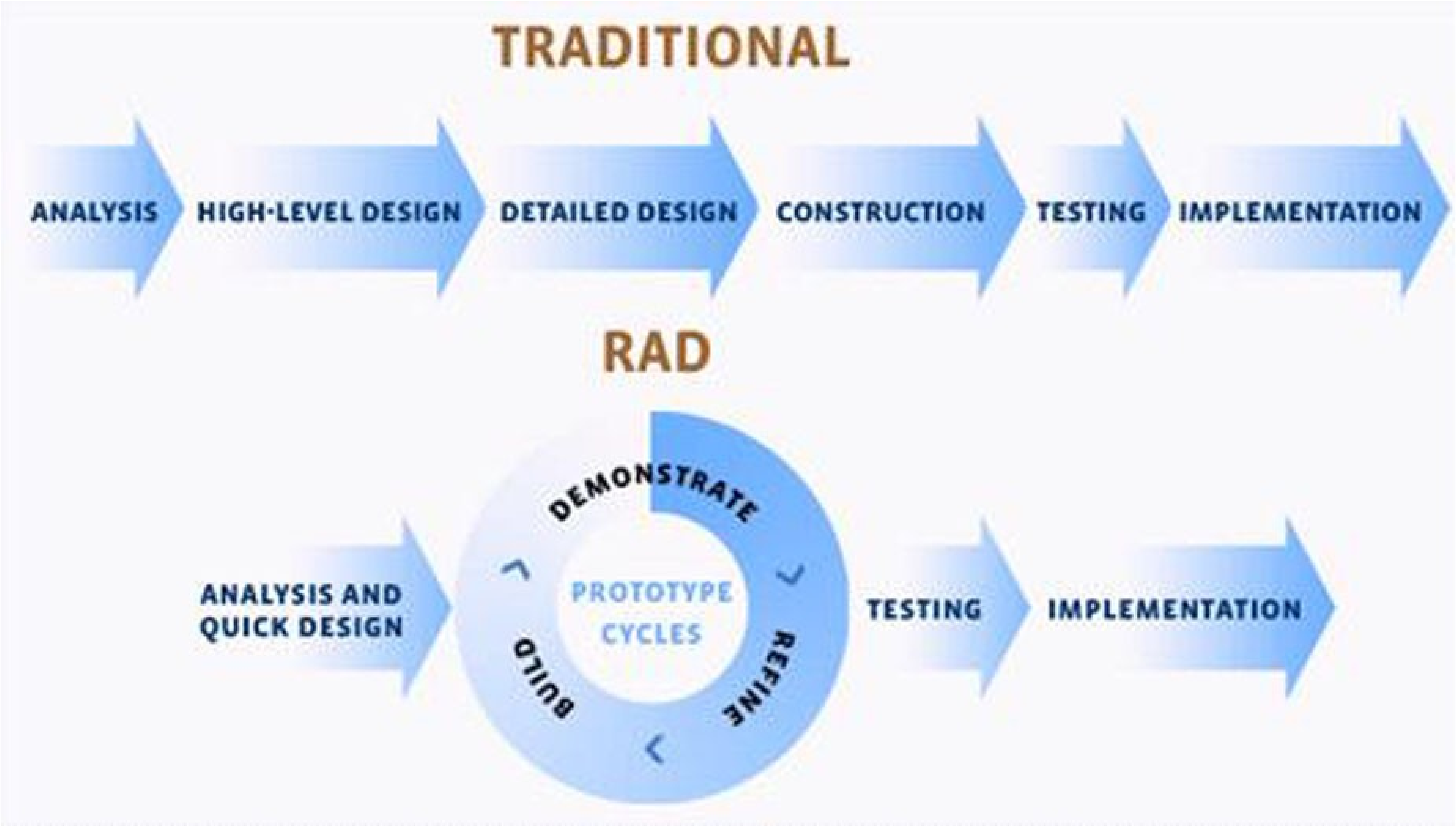


Models in RAD.

5. Testing and Turnover:

- The overall testing time is reduced in the RAD model as the prototypes are independently tested during every iteration.
- The data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage.
- After this Phase the final system is delivered to the client.
- Since most of the components are independently tested, chances of failure tends to ZERO.




TRADITIONAL
ANALYSIS
HIGH-LEVEL DESIGN
DETAILED DESIGN
CONSTRUCTION
TESTING
IMPLEMENTATION
RAD
ANALYSIS AND QUICK DESIGN
DEMONSTRATE
REFINE
BUILD
PROTOTYPE CYCLES
TESTING
IMPLEMENTATION

RAD Model vs Traditional SDLC:

Base	RAD	Traditional
1). Requirement Specification	No need of detail specification.	Detailed Specification needed.
2). Changes in Requirement	Acknowledged.	Ignored.
3). Plan	Flexible.	Strict
4). Involvement of Customer	All the time.	In planning & requirement gathering phase only.
5). Reusability of components	Increased.	Almost impossible.
6). Role of Feedback	Critical.	Just fine, Ignored most of the time.

Application of RAD Model:

- RAD model can be applied successfully to the projects in which clear modularization is possible. If the project cannot be broken into modules, RAD may fail.

- RAD should be used only when a system can be modularized to be delivered in an incremental manner.

- It should be used if there is a high availability of designers for modeling.

- It should be used only if the budget permits use of automated code generating tools.

- RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.

- Should be used where the requirements change during the project and working prototypes are to be presented to customer in small iterations of 2-3 months.

- RAD is perfect for the system which have changing requirement and needs to be delivered consistently, But it may take forever to develop fully satisfying system if requirement change are unstable and continuous in nature.

Advantage of RAD:

- ✓ 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 a 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.
- ✓ Better quality. By having users interact with evolving prototypes the business functionality from a RAD project have higher success.

Disadvantages of RAD:

- Dependency on technically strong team members for identifying business requirement.
- Only system that can be modularized can be built using RAD.
- High dependency on modeling skills and needs expert for modularization.
- Inapplicable to cheaper projects as cost of modeling and automated code Inapplicable to cheaper projects as cost of modeling and automated code.
- Management complexity is more, So is suitable for systems that are component based and scalable.
- Requires user involvement throughout the life cycle.
- Suitable for project requiring shorter development times.
- Highly trained professionals are hard to find.
- It can not be used for High Performance project or in project with high risk.

When to use RAD:

- ➲ If you are working on a tight deadline.
- ➲ If you wish to get more reusable component in project but need to finish in short time period.
- ➲ If user does not have detailed knowledge on their requirement.(‘I will know what I need, when I'll see it’ concept of prototyping.)
- ➲ If you have expert team and your project can be modularized.
- ➲ If client party is capable enough to fund the project and development team have enough fund to start with.
- ➲ If user involvement during the project is possible.
- ➲ If project is not straight forward in sense of limited and fixed feature.(as all agile methods avoid such project.)

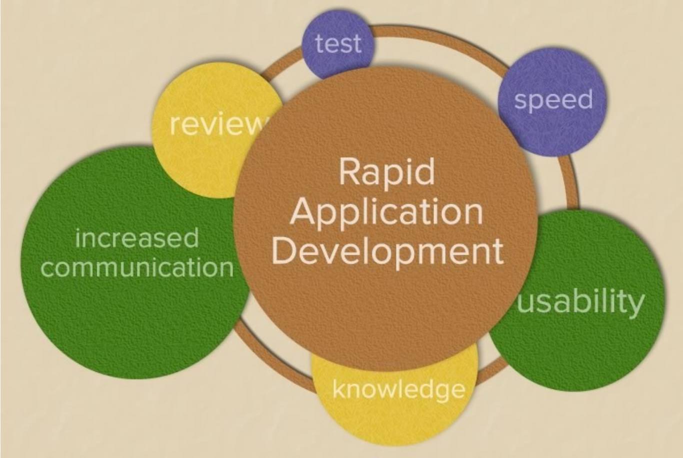
Before Choosing RAD:

Project manager should be positive on these questions before choosing RAD:

- Do you have a knowledgeable and experienced team of developers, coders, and engineers who can dedicate themselves to this ongoing development process?
- Is your client open and committed to following this approach?
- Is your client willing to commit to the determined schedule for the model completion?
- Do you have the right tools and software to effectively apply this methodology?
- Can your client and team afford this model economically?

Benefits of using RAD:

- Delivers fully functional system within 60 to 90 days. ie. Rapid Delivery of Product.
- Change with Advancing technology and Customer's Expectation.
- Built the ideal business structure.
- Minimum waste cycle.
- Reusable Components.
- User's review and low and almost ZERO chance of failure in satisfying requirements.
- End user will be able to see how the product will look like and how it would function before any of functions are actually implemented.
- Improvement in development team's skill.



Alternatives to RAD:

If RAD seems like “Not Structured Enough” for your project, you can choose Traditional Waterfall as alternative. Because RAD was created in response to the inadequacies of traditional software process model.

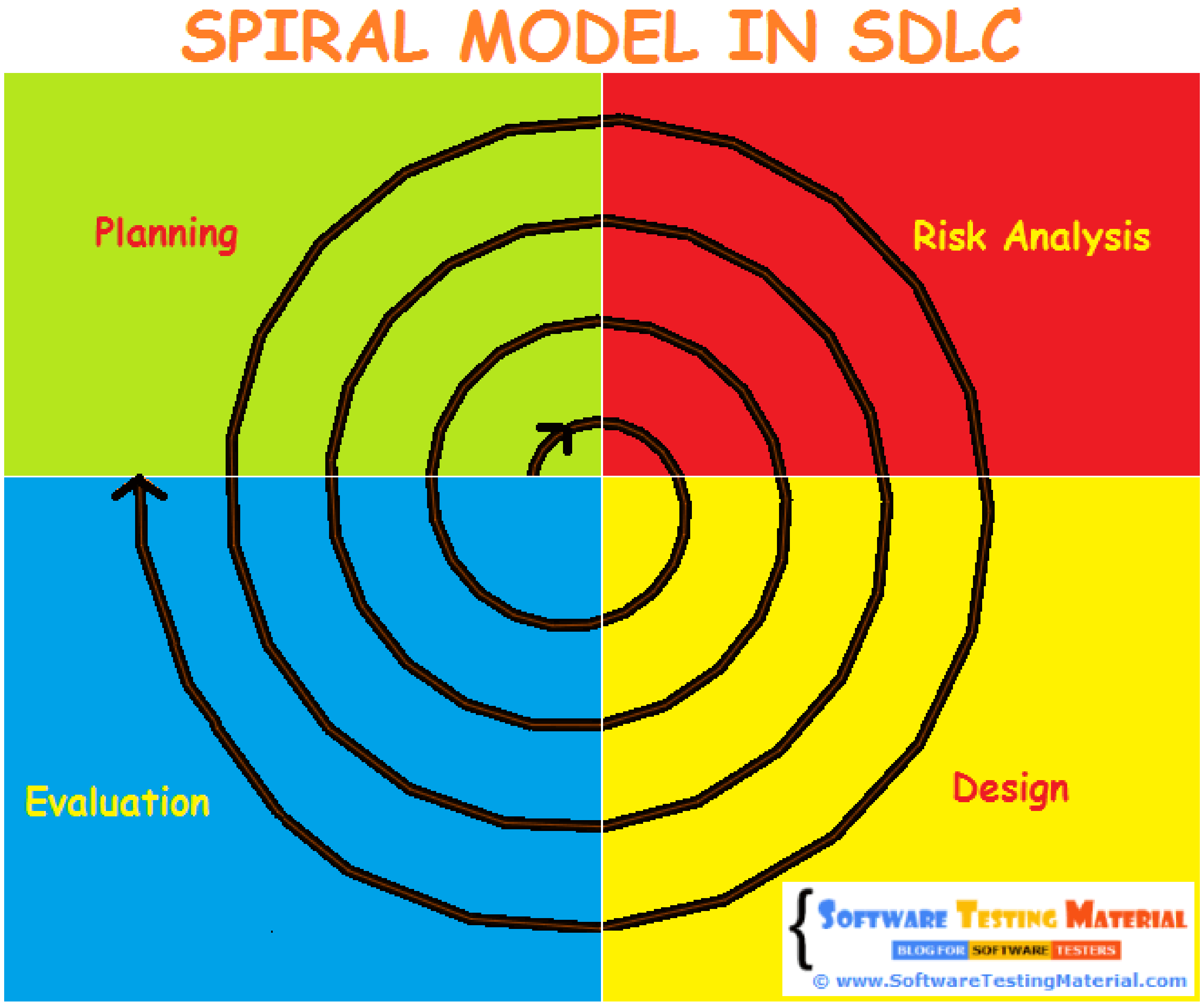
If RAD is too expensive for your project, you can choose Lean Software Development, Lean Software development methodology Focus specifically on reducing waste Throughout the project.(by scrapping Unneeded Features, developer’s autonomy over client etc.)

Sources:

https://www.w3computing.com/systemsanalysis/rapid-application-development/
https://tutorialspoint.com/sdlc/sdlc_rad_model.htm
https://kissflow.com/rad/rapid-application-development/
https://en.wikipedia.org/wiki/Rapid_application_development

Photos:

- hacker.io
- shutterstock
- tutorialspoint
- carrerBuilder.com
- Pinterest.com
- hokuapps.com


SPIRAL MODEL IN SDLC
Planning
Risk Analysis
Evaluation
Design
Software Testing Material
BLOG FOR SOFTWARE TESTERS
© www.SoftwareTestingMaterial.com

Introduction:

- In this system development method, we combine the features of both, prototype model and waterfall model.

- It’s a risk-driven model which means that the overall success of a project highly depends on the risks analysis phase. Risk analysis requires specific expertise on every iteration. Thus, to review and analyze the project from time to time, special skills are needed.

- In spiral model we can arrange all the activities in the form of a spiral

- A spiral model is divided into number of framework activities, called task regions

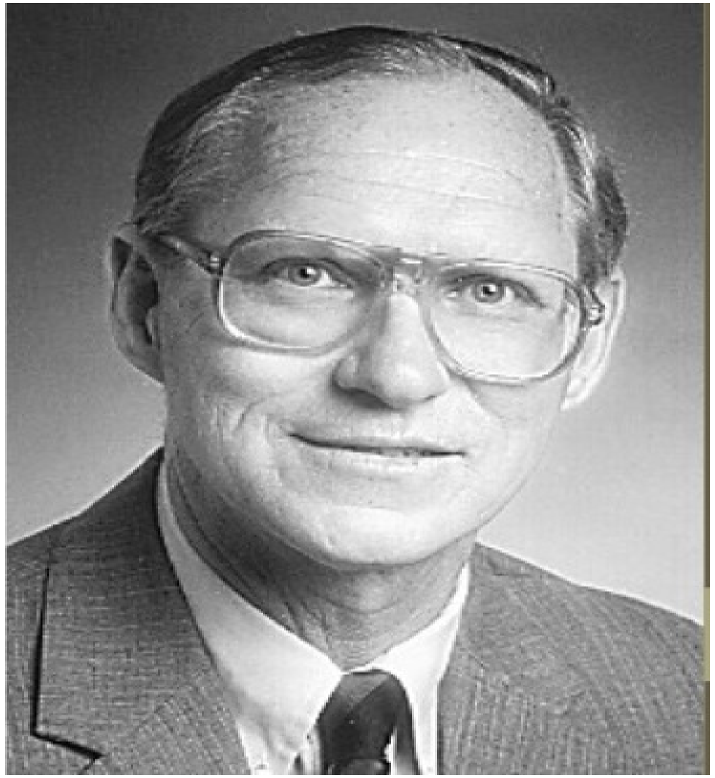
- Typically, there are between three and six task regions.

- **“If there’s a possibility to add some additional functionality at the last stages of software product development. Since risk monitoring and regular expertise are core characteristics of this approach, the overall project becomes more transparent.”**

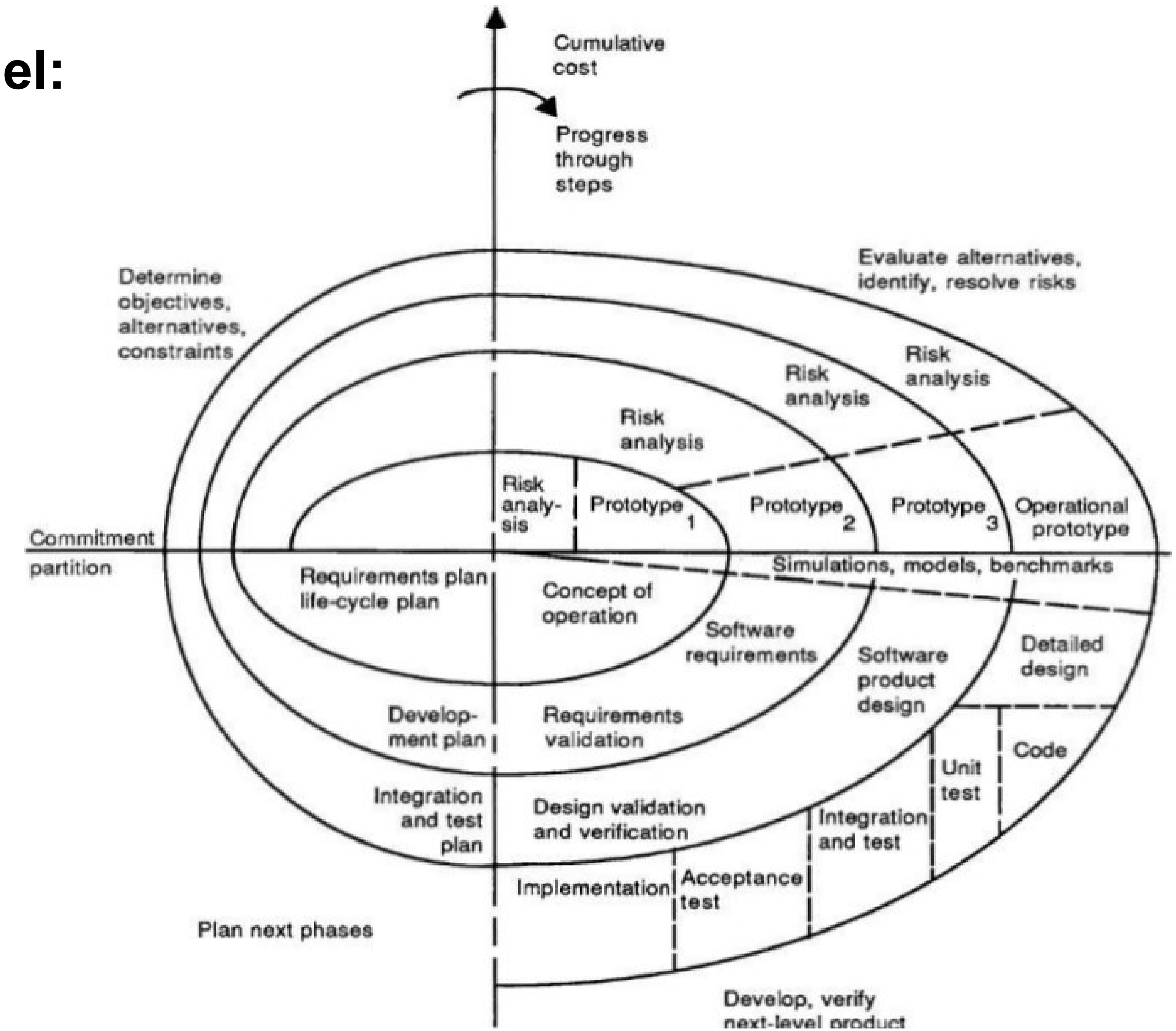
History:

- The spiral model was defined by Barry Boehm in his 1988 article.

- This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration matters.



Spiral Model:



Phases:

- Planning & Customer Communication
- Risk Analysis
- Engineering
- Customer Evaluation

Planning & Customer Communication:

- Includes understanding the system requirements by continuous communication between the customer and the system analyst.
- required to establish effective communication between developer and customer.
- Task required to define resources, Timelines and other project related Information.

Risk Analysis:

- 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.
- It required to assess both technical and management risks.

Engineering:

- In this phase software is developed, along with testing at the end of the phase.
- It includes testing, coding and deploying software at the customer site
- Hence in this phase the development(coding and testing) is done.

Customer Evaluation:

- This phase allows the customer to evaluate the output of the project before the project continues to the next spiral

- It required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

Advantages:

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for mission critical project.
- Strong approval and documentation control.
- Additional functionality can be added at a later date.

Disadvantages:

- Can be 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.

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)

Strength:

- Provides early indication of insurmountable risks, without much cost
- Users see the system early because of rapid prototyping tools
- Critical high-risk functions are developed first
- The design does not have to be perfect
- Users can be closely tied to all lifecycle steps
- Early and frequent feedback from users

Moreover:

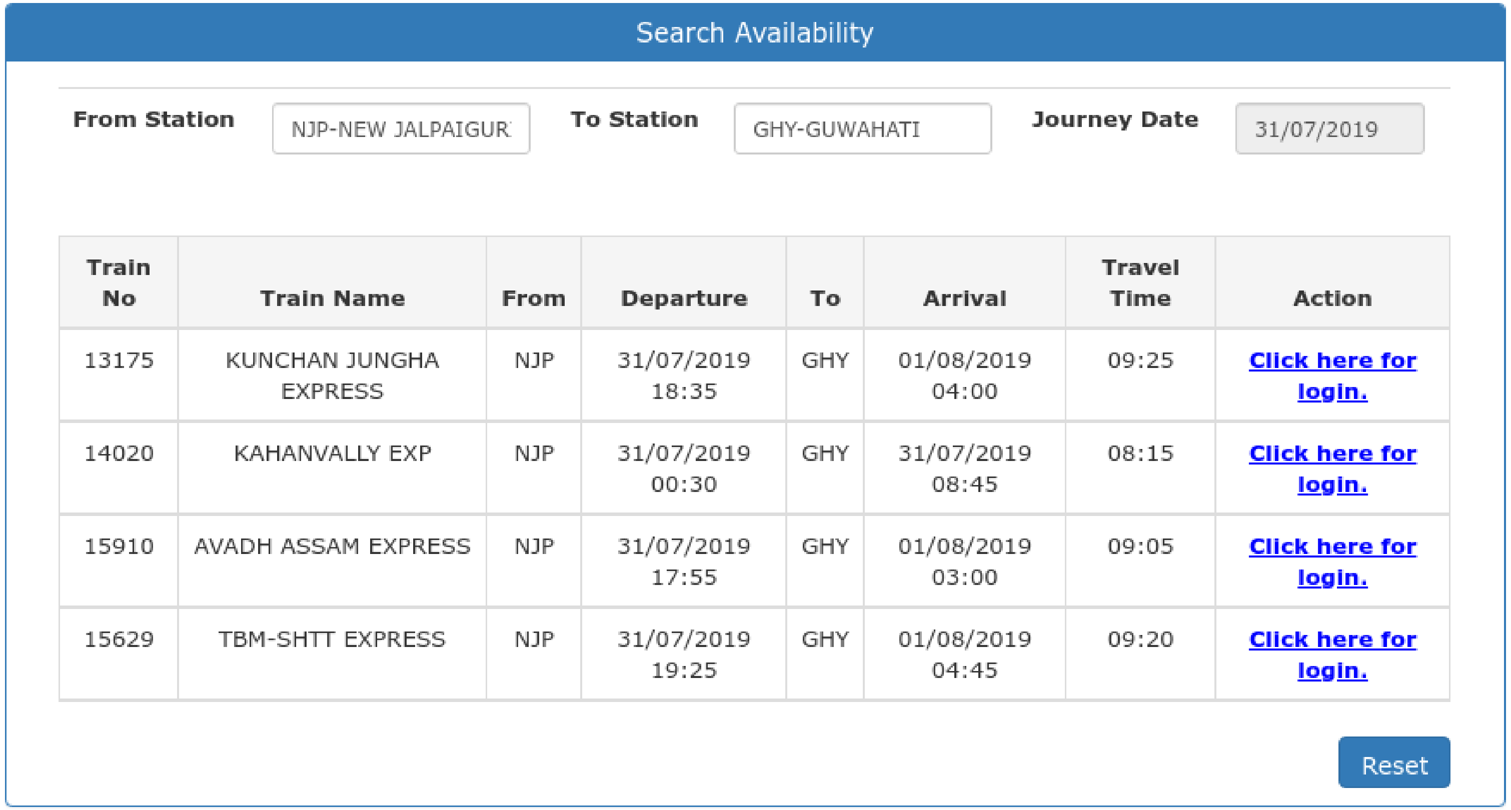
Another important thing to remember is that Spiral Model should be used in that kind of projects it was initially designed for. It can be a good option if you face a medium, or high-risk project and costs are highly important, a customer is unsure completely about his needs and requirements are complex, and significant changes are expected.

Sources:

https://www.softwaretestinghelp.com/spiral-model-what-is-sdlc-spiral-model/

https://www.guru99.com/what-is-spiral-model-when-to-use-advantages-disadvantages.html
https://www.geeksforgeeks.org/software-engineering-spiral-model/
https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
https://en.wikipedia.org/wiki/Spiral_model

An interactive system for railway passengers that might find train times from terminals installed in stations.



Search Availability

From Station: NJP-NEW JALPAIGUR

To Station: GHY-GUWAHATI

Journey Date: 31/07/2019

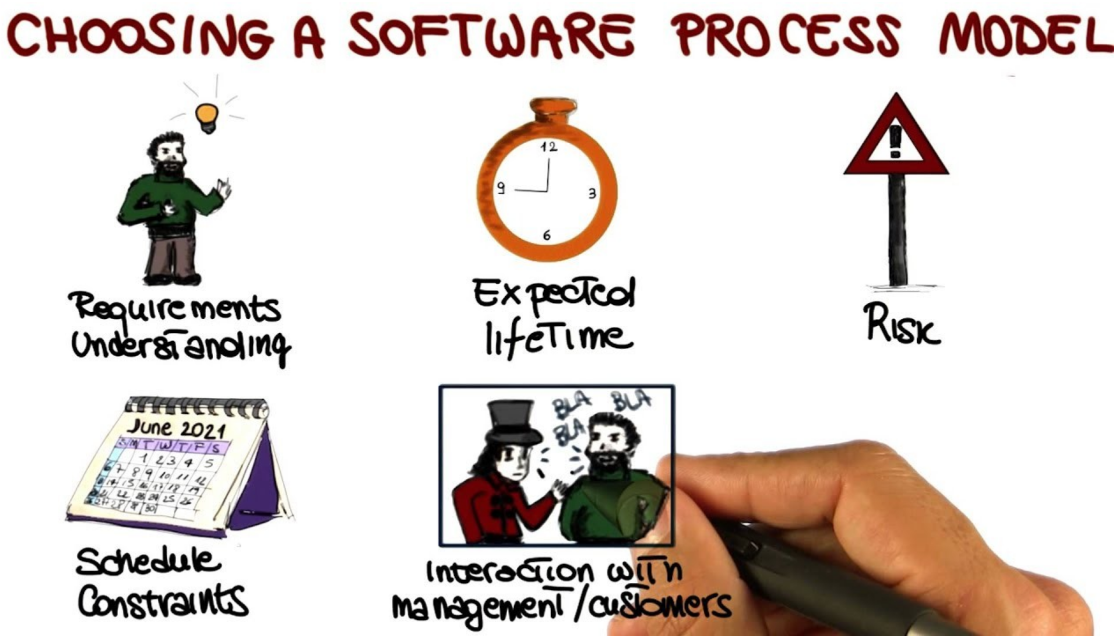
Train No	Train Name	From	Departure	To	Arrival	Travel Time	Action
13175	KUNCHAN JUNGHA EXPRESS	NJP	31/07/2019 18:35	GHY	01/08/2019 04:00	09:25	Click here for login.
14020	KAHANVALLY EXP	NJP	31/07/2019 00:30	GHY	31/07/2019 08:45	08:15	Click here for login.
15910	AVADH ASSAM EXPRESS	NJP	31/07/2019 17:55	GHY	01/08/2019 03:00	09:05	Click here for login.
15629	TBM-SHTT EXPRESS	NJP	31/07/2019 19:25	GHY	01/08/2019 04:45	09:20	Click here for login.

Problem Definition:

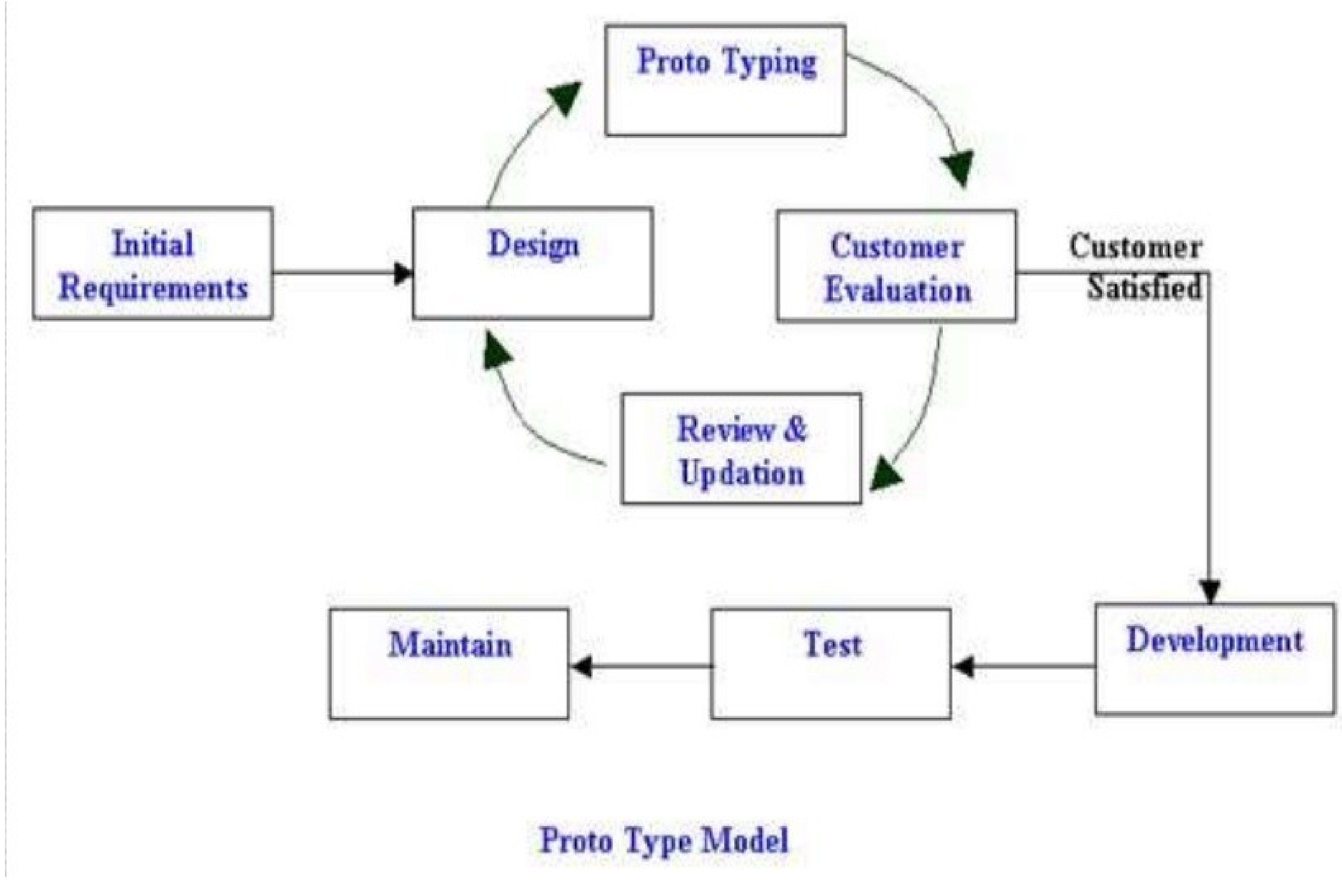
- The system needs to show accurate time and schedules of the train.
- Interactions with users is major concern.
- Interface is the key changing requirement.
- Frequent change in policies.

Choosing Correct SW Process Model

- Installing system in train terminals may be bit challenging in terms of cost and time.
- The requirements are initially not well-understood.
- This system can be modularized to deliver in prototype manner.
- Success of system is mandatory.


CHOOSING A SOFTWARE PROCESS MODEL
Requirements Understanding
Expected lifeTime
Risk
June 2021
Schedule Constraints
BLA BLA BLA
Interaction with management/customers

Prototyping Model:



Why Prototyping Model?

- Low cost and time.
- Requirement Determination.
- Policy Changes.
- Missing functionality identification.
- Good guarantees of success.

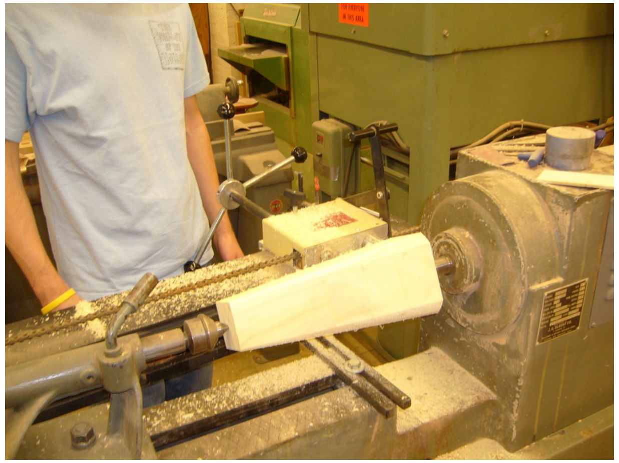
What is Prototype Model In fact?

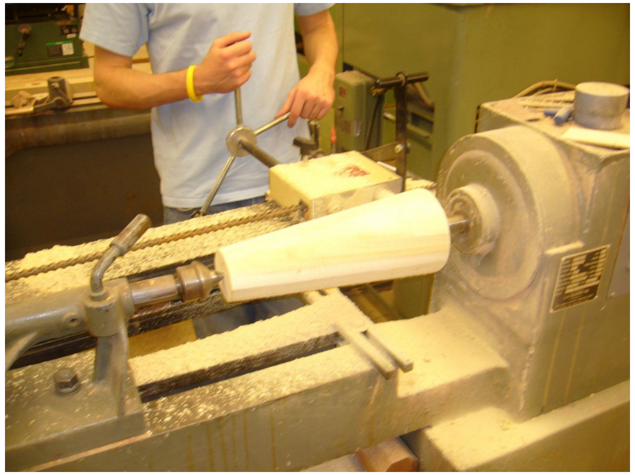
- Replaces some of the parts of the SDLC with and evolutionary and iterative process.
- Software prototypes are **repeatedly** **provided** to customer for evaluation and feedback.
  - Primarily gives design and missing functionality.
  - Development team are provided new requirements.
- Ultimately, the product reaches a satisfactory completion

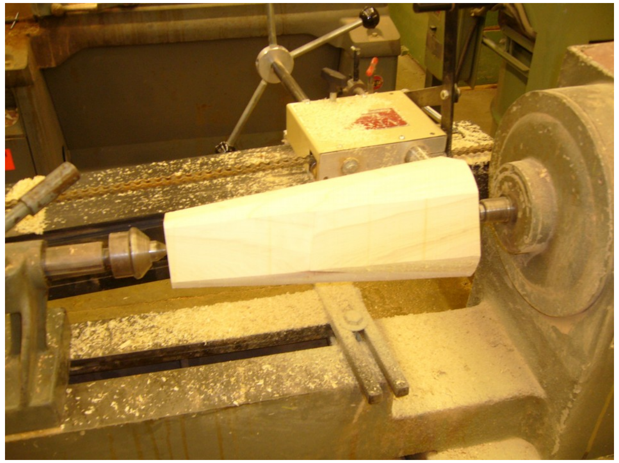
- Development of fast prototypes and customer availability for feedback.
- Developers respond to feedback and add additional parts as application evolves into an acceptable product.

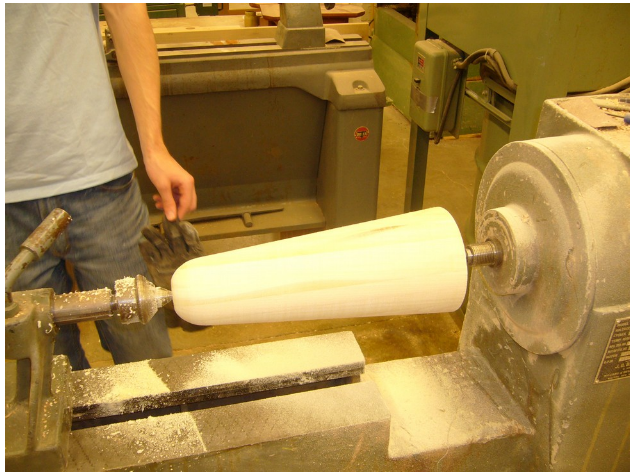
- **Types of Prototype:**
  - Throw away Prototype
  - Evolutionary Prototype

Practical Example: (wood hobby shop)

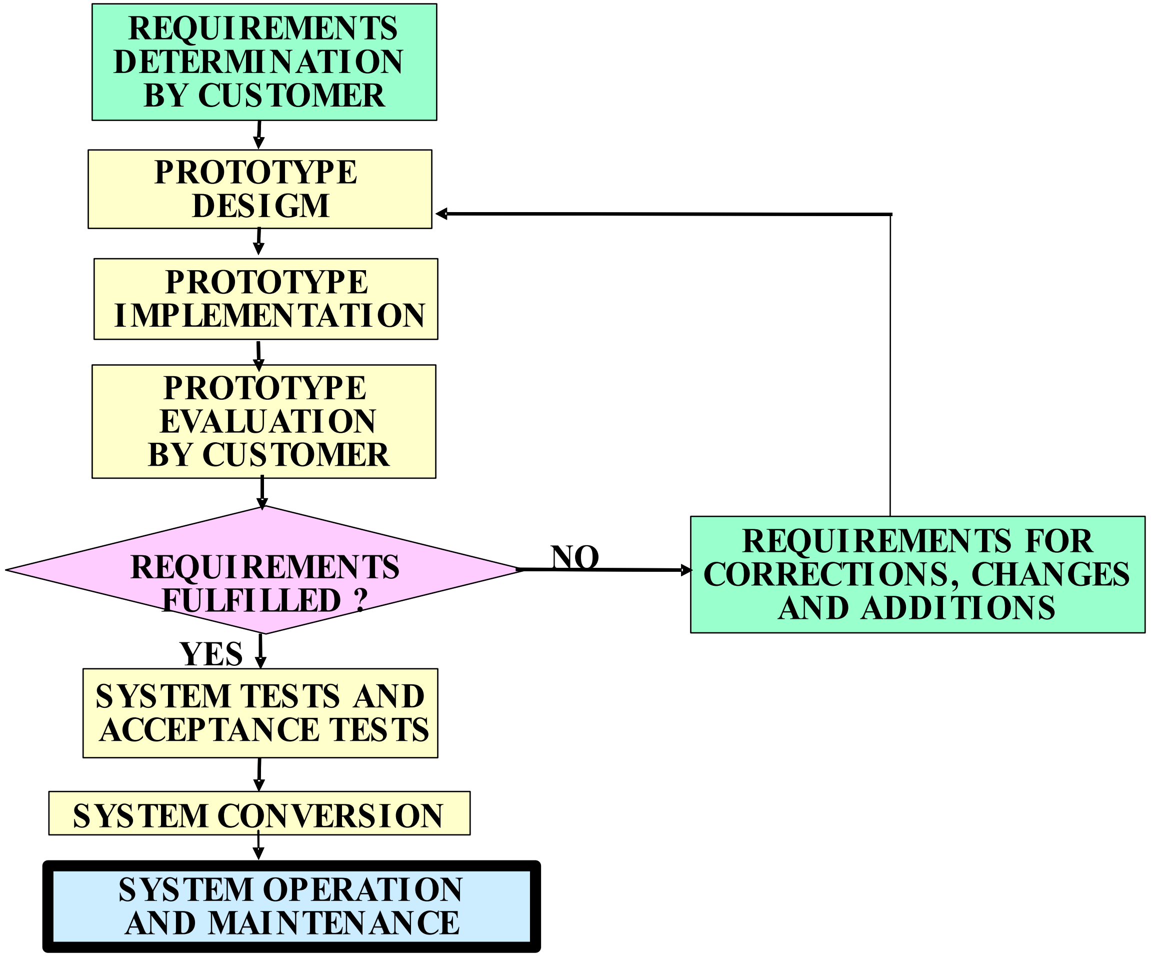








Prototyping Flow Chart:



REQUIREMENTS DETERMINATION BY CUSTOMER

PROTOTYPE DESIGM

PROTOTYPE IMPLEMENTATION

PROTOTYPE EVALUATION BY CUSTOMER

REQUIREMENTS FULFILLED ?

SYSTEM TESTS AND ACCEPTANCE TESTS

SYSTEM CONVERSION

SYSTEM OPERATION AND MAINTENANCE

REQUIREMENTS FOR CORRECTIONS, CHANGES AND ADDITIONS

- **Advantages**
  - Better fit to customer requirements and reduced risk of project failure
  - Users are actively involved in the development.
  - Customer can see the steady progress.
  - Quicker user feedback is available leading to better solutions.
  - This is useful when requirements are changing rapidly.
  - Missing functionality and error can be identified easily
  - Quick implementation of, incomplete, but functional, application.

- **Disadvantages**
  - It is impossible to know how long it will take.
  - There is no way to know the number of iteration required.
  - 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.

Points to be remembered regarding Prototype model:

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- System having a very high amount of interaction with end users, are best suited for Prototype model.
- 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 usable system.
- ***“End user are actually excellent for designing good human computer interface systems.”***

Sources:

- https://www.irctc.co.in (for train terminal system information)
- http://tryqa.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/
- Pearson Education Limited 2004
- https://searchcio.techtarget.com/definition/Prototyping-Model
- https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm
- http://ecomputernotes.com/software-engineering/explain-prototyping-model

Images:

- http://www.piedpiper.com/
- https://www.freepik.com/

Thank You !

Any Confusion? Okay, Lets Clear it. ;)

EO
F