

# System Analysis and Design (SAD)

Dhiraj Bashyal

# System Development Life Cycle (SDLC)

Different Approaches to Improving  
Information Systems  
Development

# Introduction

- Several different approaches have been developed in the continuous effort to improve the systems analysis and design process

# System Development Methodologies

1. Waterfall Methodology
2. Prototype Methodology
3. Joint Application Development
4. Agile Methodology
5. Spiral Model
6. Rapid Application Development
7. Extreme Programming

# Prototype Methodology

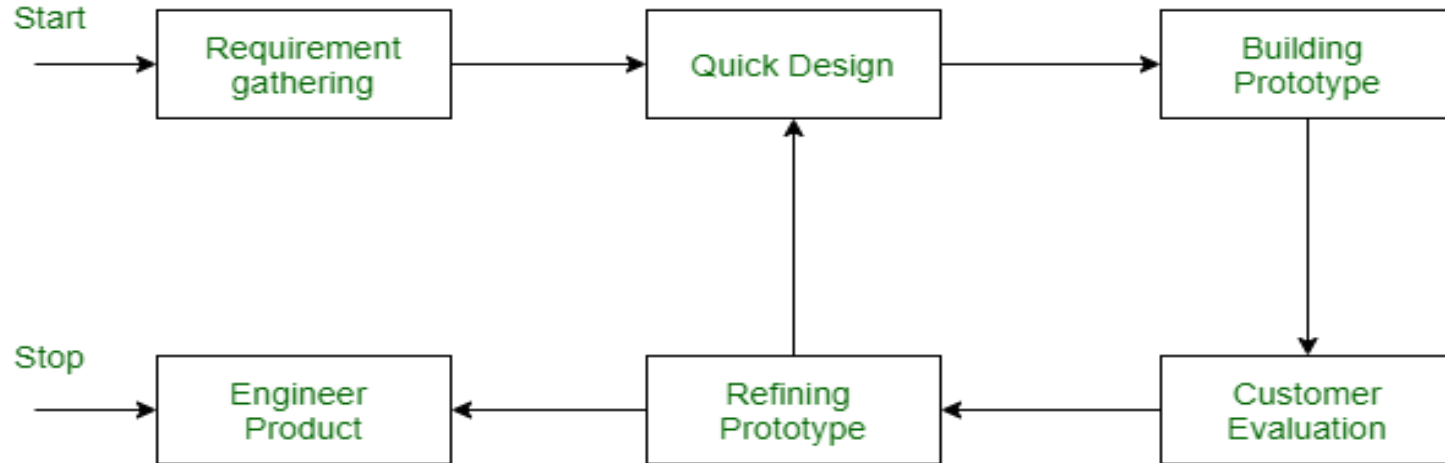


Figure - Prototype Model

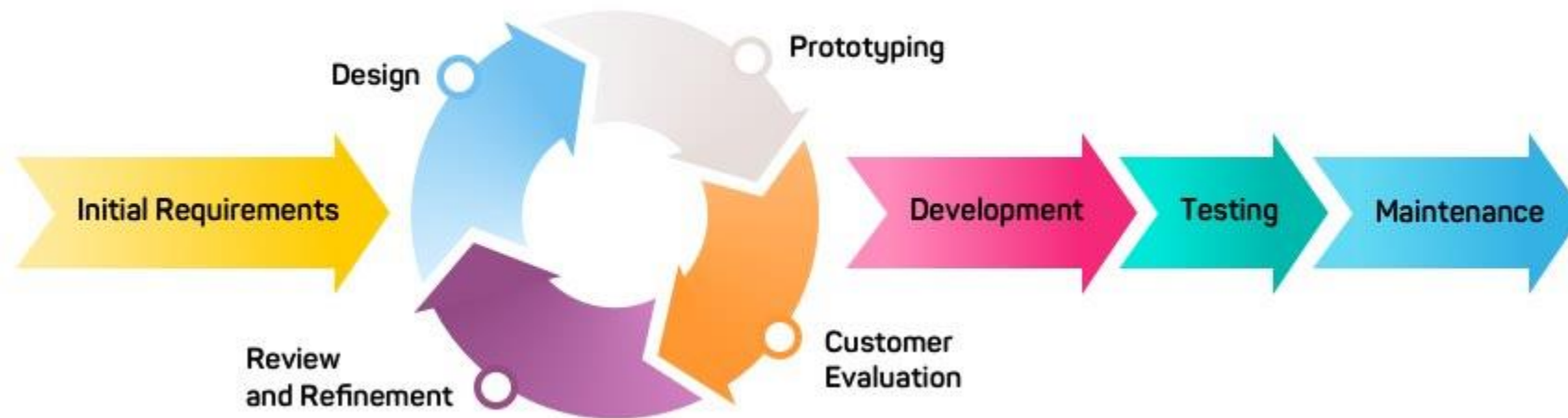
# Prototype Methodology – Introduction[1]

- Prototype Model is used when the customers do not know the exact project requirements beforehand.
- In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.
- System is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle.

# Prototype Methodology – Introduction[2]

- Prototyping is a form of rapid application development (RAD).
- Prototyping is a rapid, iterative, and incremental process of systems development in which requirements are converted to a working system that is continually revised through close work between the development team and the users.
- We can build a prototype with any computer language or development tool, but special prototyping tools have been developed to simplify the process.
- A prototype can be developed with some fourth-generation language (4GL), with the query and screen and report design tools of a database management system, and with tools called computer-aided software engineering (CASE) tools.

# Prototype Methodology – Introduction[2]





# Advantages of Prototype Model

1. Show the prototype to the client to have a **clear understanding** and complete 'feel' of the functionality developed in the software. It ensures a greater **level of customer satisfaction and comfort**.
2. **Identify the scope** of the refinement and accordingly accommodate new changes in the given requirements.
3. Significantly **reduce the risk of failure** using this method and **identify the potential risks at an early stage** and moderation steps can be taken quickly.
4. The communication between the software development team and the client **makes a very good and conducive environment** during a project.
5. It helps in **requirement gathering and requirement analysis** when there is a lack of required documents.

# Disadvantages of Prototype Model

1. Prototyping is usually done at the cost of the developer so, it is should be done using minimal resources otherwise organization's development **cost stretch too much**.
2. Customers sometimes demand **the actual product to be delivered soon after seeing an early prototype**.
3. The clients have **too much involvement** which is not always aligned with the software developer.
4. It does not appreciate too many modifications in the project as it easily disturbs the existing workflow of the entire software development process.
5. Customers may not be satisfied or interested in the product after seeing the initial prototype

# Joint Application Development Methodology

## Presentation Topic [1]

- Introduction
- Objective
- Diagram
- Advantage
- Disadvantage
- Implementation
- conclusion

# Agile Methodologies[1]

- **Agile Software Development** is an approach that is used to design a disciplined software management process which also allows some frequent alteration in the development project.
- This is a type of software development methodology that is one conceptual framework for undertaking various software engineering projects.
- It is used to minimize risk by developing software in short time boxes which are called iterations that generally last for one week to one month
- Agile Methodology meaning a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project.
- In the Agile model in software testing, both **development and testing activities are concurrent**,

# Agile Methodology[2]

- According to Fowler (2003), the Agile Methodologies share three key principles:
  1. a focus on adaptive rather than *predictive* methodologies,
  2. a focus on people rather than roles, and
  3. a focus on self-adaptive processes.
- The Agile Methodologies promote a self-adaptive software development process. As software is developed, the process used to develop it should be refined and improved.
- Development teams can do this through a review process, often associated with the completion of iterations.
- The implication is that, as processes are adapted, one would not expect to find a single monolithic methodology within a given corporation or enterprise. Instead, one would find many variations of the methodology, each of which reflects the particular talents and experience of the team using it.

# Agile Methodology[3]

- The word 'agile' means – Able to move your body quickly and easily.
- Able to think quickly and clearly.
- In business, 'agile' is used for describing ways of planning and doing work wherein it is understood that making changes as needed is an important part of the job. Business 'agility' means that a company is always in a position to take account of the market changes.
- In software development, the term 'agile' is adapted to mean **'the ability to respond to changes – changes from Requirements, Technology and People.'**

# Agile Methodology[4]



# Advantages of Agile Methodology

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Agile methodology has an adaptive approach that is able to respond to the changing requirements of the clients.
- Direct communication and constant feedback from customer representatives leave no space for any guesswork in the system.



# Disadvantages of Agile Methodology

- In the case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- This methodology focuses on working software rather than documentation, hence it may result in a lack of documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers unless combined with experienced resources.

# Agile Model Vs Waterfall Model

Agile Model	Waterfall Model
<ul style="list-style-type: none"><li>Agile methodology definition: Agile methodologies propose incremental and iterative approach to software design</li></ul>	<ul style="list-style-type: none"><li>Waterfall Model: Development of the software flows sequentially from start point to end point.</li></ul>
<ul style="list-style-type: none"><li>The Agile process in software engineering is broken into individual models that designers work on</li></ul>	<ul style="list-style-type: none"><li>The design process is not broken into an individual models</li></ul>
<ul style="list-style-type: none"><li>The customer has early and frequent opportunities to look at the product and make decision and changes to the project</li></ul>	<ul style="list-style-type: none"><li>The customer can only see the product at the end of the project</li></ul>
<ul style="list-style-type: none"><li>Agile model is considered unstructured compared to the waterfall model</li></ul>	<ul style="list-style-type: none"><li>Waterfall model are more secure because they are so plan oriented</li></ul>
<ul style="list-style-type: none"><li>Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time.</li></ul>	<ul style="list-style-type: none"><li>All sorts of project can be estimated and completed.</li></ul>
<ul style="list-style-type: none"><li>Error can be fixed in the middle of the project.</li></ul>	<ul style="list-style-type: none"><li>Only at the end, the whole product is tested. If the requirement error is found or any changes have to be made, the project has to start from the beginning</li></ul>

Source: <https://www.guru99.com/agile-scrum-extreme-testing.html>

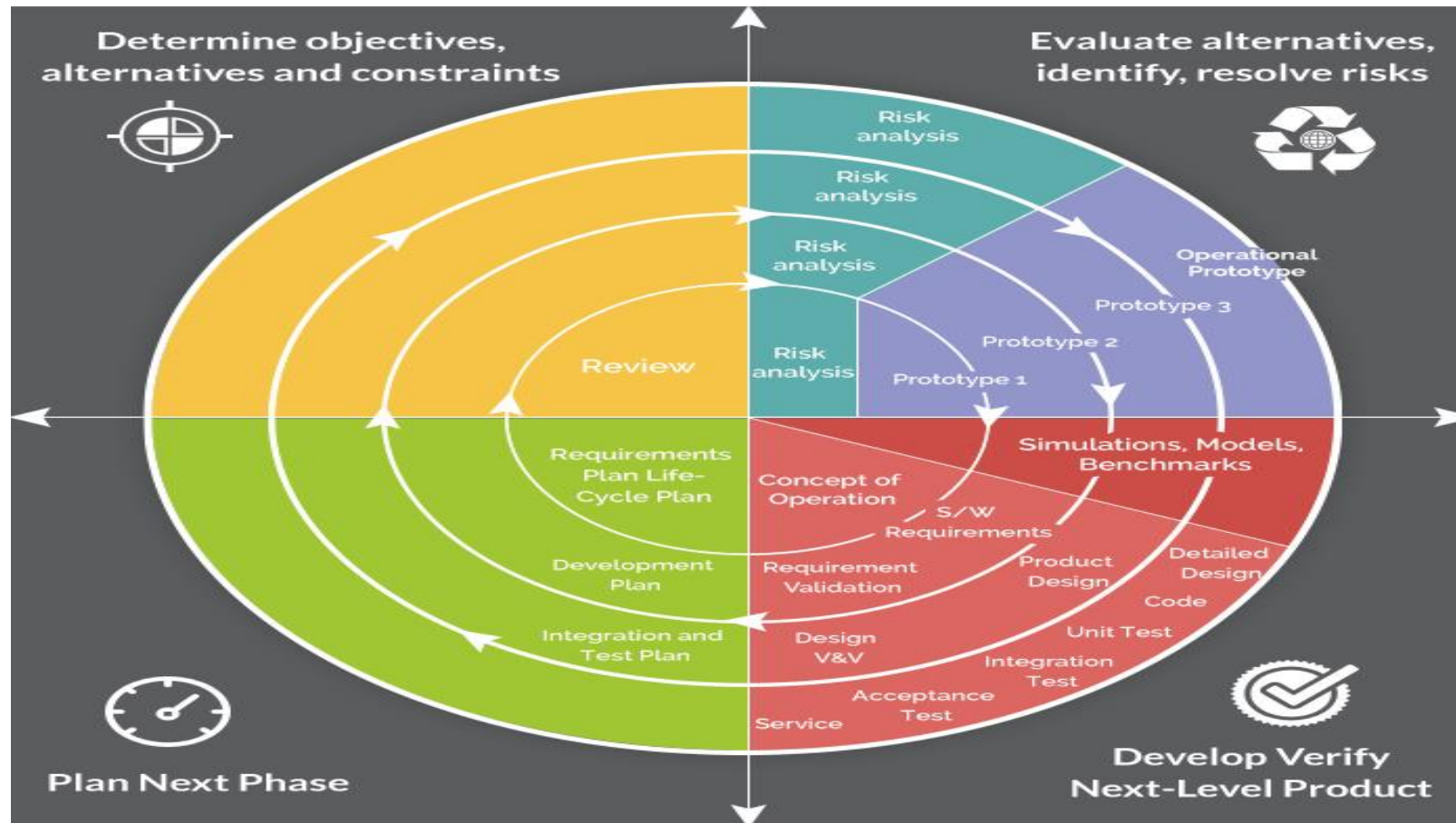
# Spiral Methodology – Introduction[1]

- The **Spiral Model** is a sophisticated model that focuses on the early identification and reduction of project risks.
- In this software development methodology, developers start on a small scale then explores the risks involved in the project, make a plan to handle the risks, and finally decides whether to take the next step of the project to do the next iteration of the spiral.
- The success of any Spiral Lifecycle Model depends on the reliable, attentive, and knowledgeable management of the project.

# Spiral Methodology – Introduction[2]

- The **Spiral Model** is similar to the incremental model, with more emphasis placed on risk analysis.
- The spiral model has four phases:
  - Planning,
  - Risk analysis,
  - Engineering and
  - Evaluation.
- A software project repeatedly passes through these phases in iterations (called spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed.

# Spiral Methodology – Introduction[3]



# Advantages of Spiral Model

- The high amount of risk analysis being done hence, avoidance of possible risk is certainly reduced with this model.
- This model is good for large size and critical projects.
- In the spiral model, additional functionality can be added at a later date.
- Development is fast and features are added systematically in this model.
- It is more suited for high-risk projects, where business needs may differ from time to time basis.

# Disadvantages of Spiral Model

- It is certainly a costly model to use in terms of development.
- The success of the entire project is dependent on the risk analysis phase thus, failure in this phase may damage the entire project.
- It is not appropriate for low-risk projects.
- The big risk of this methodology is that it may continue indefinitely and never finish.
- Documentation is more as it has intermediate phases.

# Rapid Application Development (RAD)-Introduction[1]

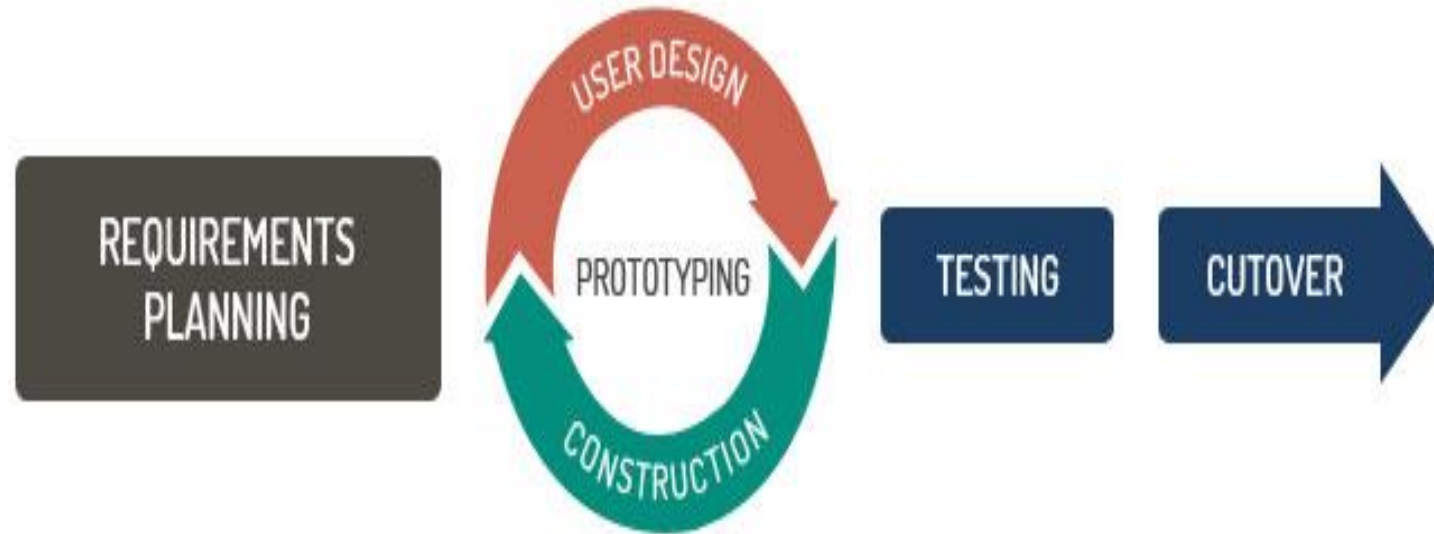
- **Rapid Application Development (RAD)** is an effective methodology that provides much quicker development and higher-quality results than those achieved with the other software development methodologies. It is designed in such a way that it easily takes the maximum advantage of the software development.
- The main objective of this methodology is to accelerate the entire software development process. The goal is easily achievable because it allows active user participation in the development process



# Rapid Application Development (RAD)-Introduction[2]

- **Rapid Application Development (RAD)** is an effective methodology that provides much quicker development and higher-quality results than those achieved with the other software development methodologies. It is designed in such a way that it easily takes the maximum advantage of the software development. The main objective of this methodology is to accelerate the entire software development process. The goal is easily achievable because it allows active user participation in the development process.

# Rapid Application Development (RAD)-Introduction[3]



# Advantages of RAD Model

- Rapid Application development model helps to reduce the risk and required efforts on the part of the software developer.
- Additionally, this model also helps the clients to take quick reviews for the project.
- This methodology encourages customer feedback which always provides improvement scope for any software development project.
- As a result of prototyping in nature, there is a possibility of lesser defects.
- Each phase in RAD delivers the highest priority functionality to the client.

# Disadvantages of RAD Model

- This model depends on the strong team and individual performances for clearly identifying the exact requirement of the business.
- It only works on systems that can be modularized can be built using this methodology.
- This approach demands highly skilled developers and a designer's team which may not be possible for every organization.
- This method is not applicable for the developer to use in small budget projects as the cost of modeling and automated code generation is very high.
- Progress and problems accustomed are hard to track as such there is no documentation to demonstrate what has been done.

# When to use RAD Model?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

# Extreme Programming Methodology

- **Extreme Programming** is distinguished by:
  - its short cycles
  - incremental planning approach
  - focus on automated tests written by programmers and customers to monitor the development process.
  - reliance on an evolutionary approach to development that lasts throughout the lifetime of the system.
  - use of two-person programming teams.
  - Planning, analysis, design, and construction are all fused into a single phase of activity.

# Extreme Programming Methodology

- **Extreme Programming** is distinguished by:
  - its short cycles
  - incremental planning approach
  - focus on automated tests written by programmers and customers to monitor the development process.
  - reliance on an evolutionary approach to development that lasts throughout the lifetime of the system.
  - use of two-person programming teams.
  - Planning, analysis, design, and construction are all fused into a single phase of activity.

# Extreme Programming Methodology

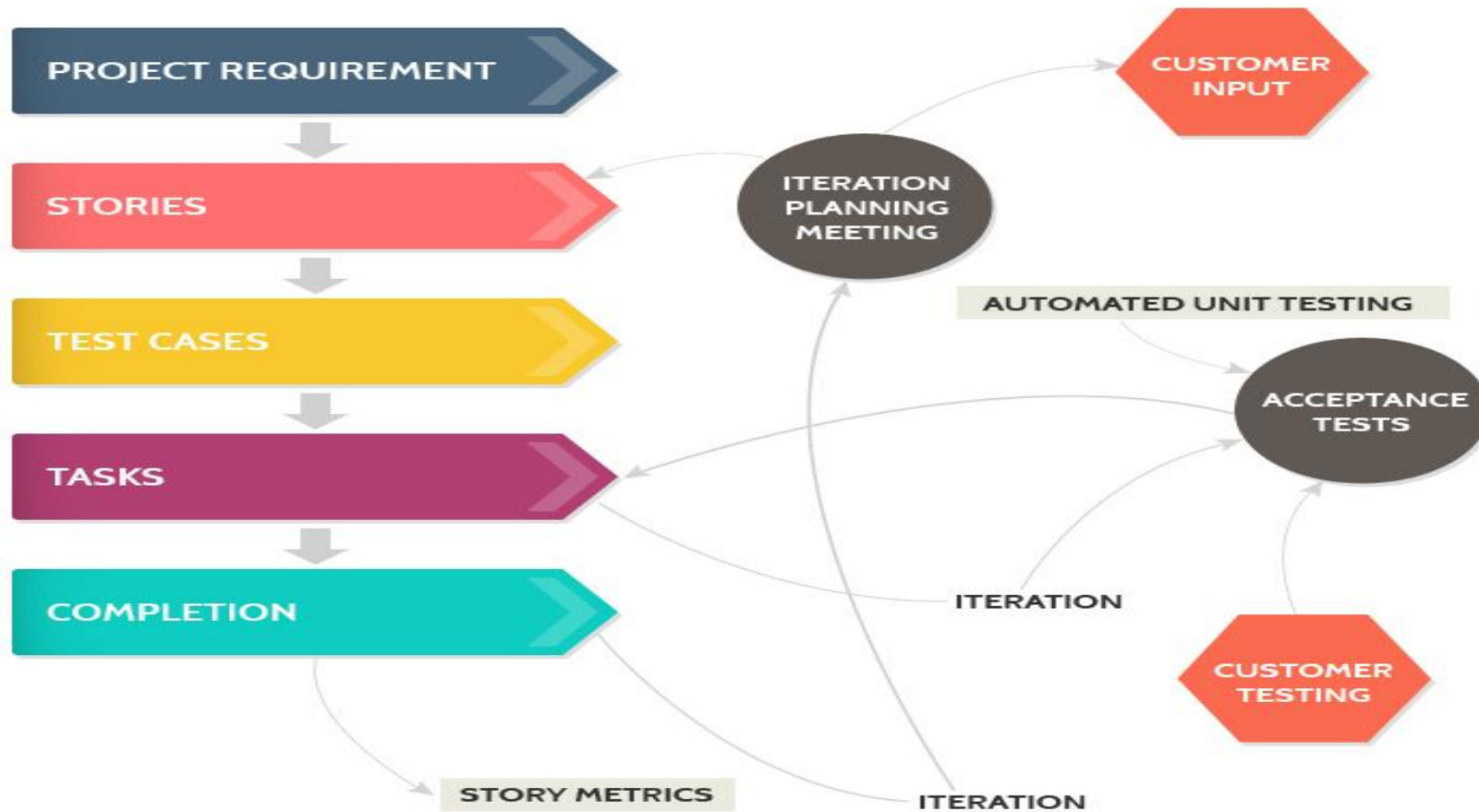
- **Extreme Programming** is distinguished by:
  - Under this approach, coding and testing are intimately related parts of the same process. The programmers who write the code also develop the tests. The emphasis is on testing those things that can break or go wrong, not on testing everything
  - Code is tested very soon after it is written. The overall philosophy behind eXtreme Programming is that the code will be integrated into the system it is being developed for and tested within a few hours after it has been written



# Why is it called ‘Extreme’?

- Extreme Programming takes the effective principles and practices to extreme levels.
- Code reviews are effective as the code is reviewed all the time.
- Testing is effective as there is continuous regression and testing
- Design is effective as everybody needs to do refactoring daily.
- Integration testing is important as integrate and test several times a day.
- Short iterations are effective as the planning game for release planning and iteration planning.

# Extreme Programming Methodology



# Advantages of Extreme Programming

- The main advantage of Extreme Programming is that this methodology allows software development companies to save costs and time required for project realization. Time savings are available because of the fact that XP focuses on the timely delivery of final products. Extreme Programming teams save lots of money because they don't use too much documentation. They usually solve problems through discussions inside of the team.
- Extreme programming methodologies emphasize customer involvement.
- This model helps to establish rational plans and schedules and to get the developers personally committed to their schedules which are surely a big advantage in the XP model.
- This model is consistent with most modern development methods so, developers are able to produce quality software.

# Disadvantages of Extreme Programming

- Some specialists say that Extreme Programming is focused on the code rather than on design. That may be a problem because good design is extremely important for software applications. It helps sell them in the software market. Additionally, in XP projects the defect documentation is not always good. Lack of defect documentation may lead to the occurrence of similar bugs in the future.
- This methodology is only as effective as the people involved, Agile does not solve this issue.
- This kind of software development model requires meetings at frequent intervals at enormous expense to customers.
- It requires too many development changes which are very difficult to adopt every time for the software developer.
- In this methodology, it tends to impossible to be known exact estimates of work effort needed to provide a quote, because at the starting of the project nobody aware of the entire scope and requirements of the project.