

Software Development Process Models

03-04-2018

Software Development Process

- Today we have reached a situation where you have control over everything at the tip of your fingers. Technology has developed beyond imagination. All thanks to the software development industry!
- The world of [software development](#) is something that is limitless.
- Technology is a perfect blend of innovation and ideation that conceptualizes to form a platform, which is suitable for operating various software developments taking place.
- In fact, the methodology meant for software development is considered as a structure used for planning and controlling the procedure of creating a specialized software.
- Certainly, these innovative methods are concerned with highlighting the [process of software development](#), which does not involve the usage of any technical aspect.
- The only matter considered is proper planning for the purpose of highly integrated software development.
- The simple tenacity of these procedures is to offer customized software development as per the requirements.

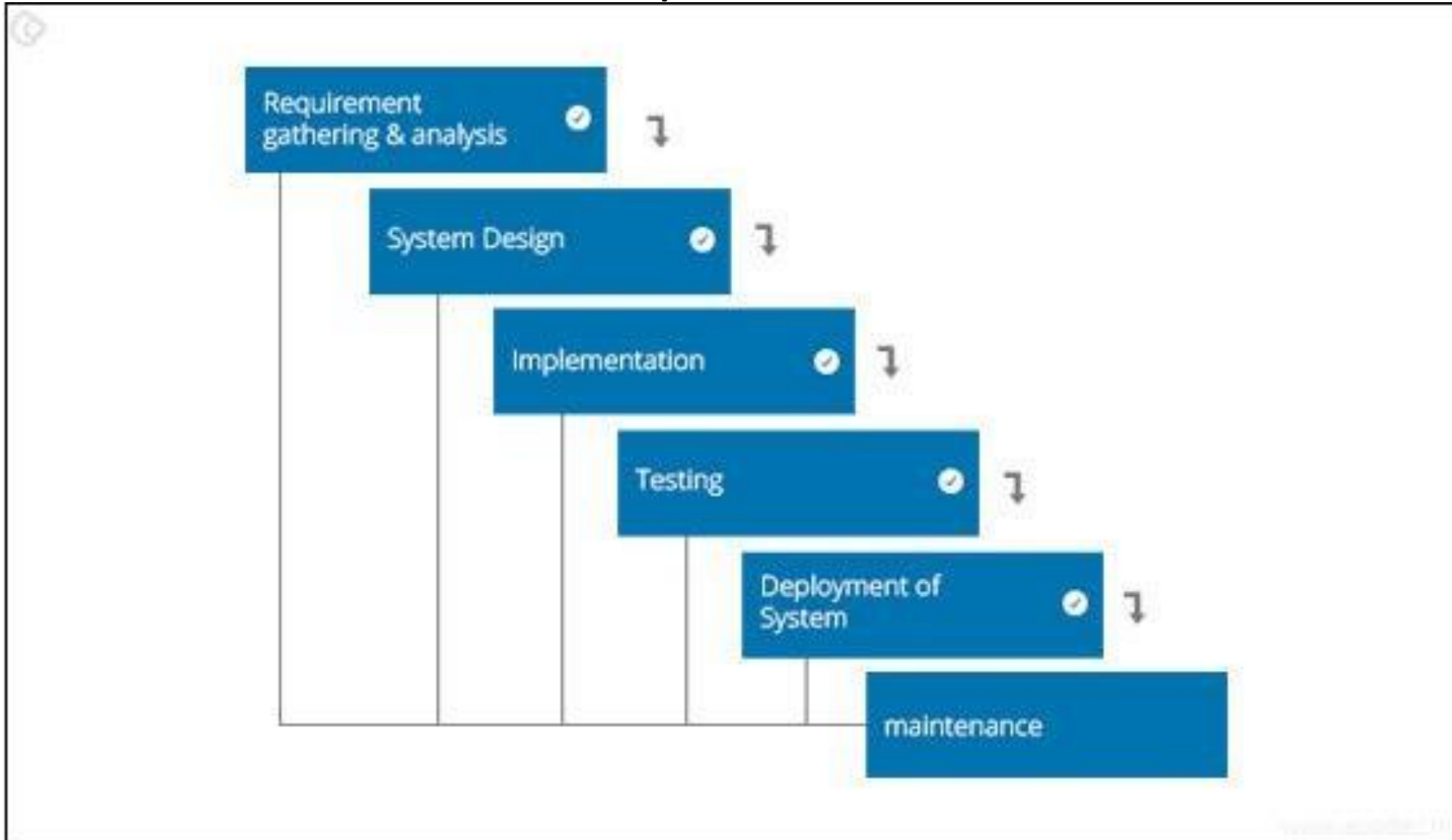
12 Best Software Development Process Models

1. Waterfall Model
2. Prototype Methodology
3. Agile Software Development Methodology
4. Rapid Application Development
5. Dynamic System Development Model
6. Spiral Model
7. Extreme Programming Model
8. Feature Driven Development
9. Joint Application Development
10. Lean Development
11. Rational Unified Process
12. Scrum Development

1. Water Fall Model

- If you are into software development at some point or the other, you would have bumped into the [Waterfall Model](#).
- Considered as the traditional method of explaining the software development process in software engineering, waterfall model happens to clarify the process into a linear flow with a specified sequence to let the users understand that further level is made progressive on completion of the previous one.
- Moreover, this methodology also talks about the fact that going back to deal with the changes is not possible.

1. Waterfall Development Model



1. Waterfall Development Model

- **Pros:**

- Easy to understand and functional
- Simple enough to handle as model is rigid
- Saves significant amount of time
- Allows for easy testing and analysis

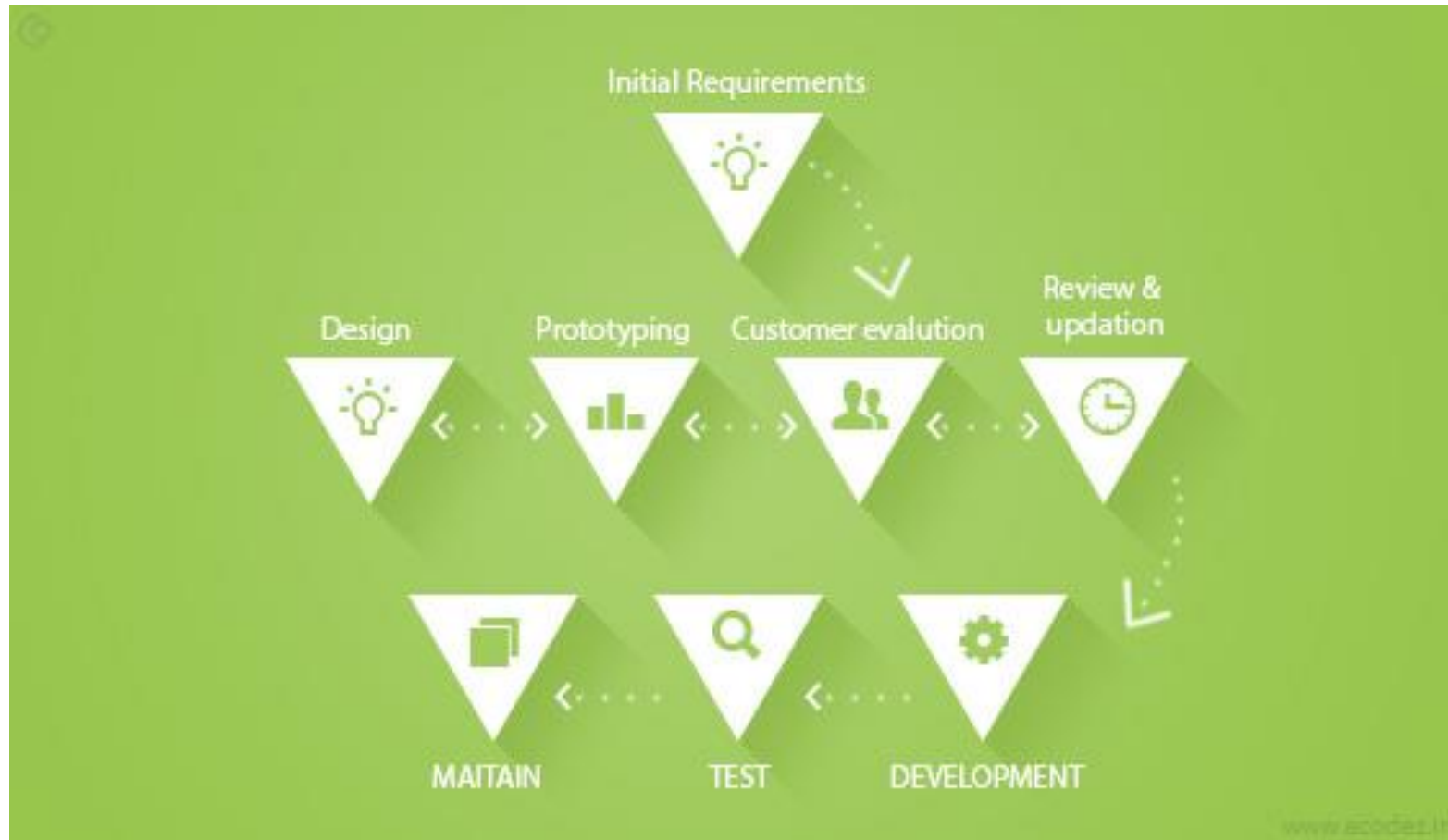
- **Cons:**

- Only matches precise needs
- Not applicable for maintenance projects
- No option to know possible outcome of a project
- Not excellent for long and ongoing projects

2. Prototype Methodology

- It is a specialized software development procedure that initiates developers towards making only the sample of the resolution to validate its functional essence to the customers and make essential changes before creating the authentic final solution.
- In fact, the best part of [this methodology](#) is that it tends to resolve a set of diversifying issues occurring with the waterfall method.

2. Prototype Methodology



2. Prototype Methodology

- **Pros:**

- Gives clear idea about the functional process of the software
- Reduces the risk of failure in a software functionality
- Assists well in requirement gathering and the overall analysis

- **Cons:**

- Chances of extension in management cost
- Excessive involvement of client can affect processing
- Too many changes affect the workflow of the software

3. Agile Software Development Methodology

- As an innovative approach, [the agile software development methodology](#) is used for articulating a well-organized project management procedure allowing for recurrent alterations.
- Certainly, such type of a methodology is one theoretical outline for undertaking several software engineering projects.
- Another good thing about it is that it minimizes peril by creating software in short time boxes, known as iterations, which happen to last from one week to one month.

3. Agile Software Development Methodology

- **Pros:**

- Adaptive approach that responds to changes favorably
- Allows for direct communication to maintain transparency
- Improved quality by finding and fixing defects quickly and identifying expectation mismatches early.

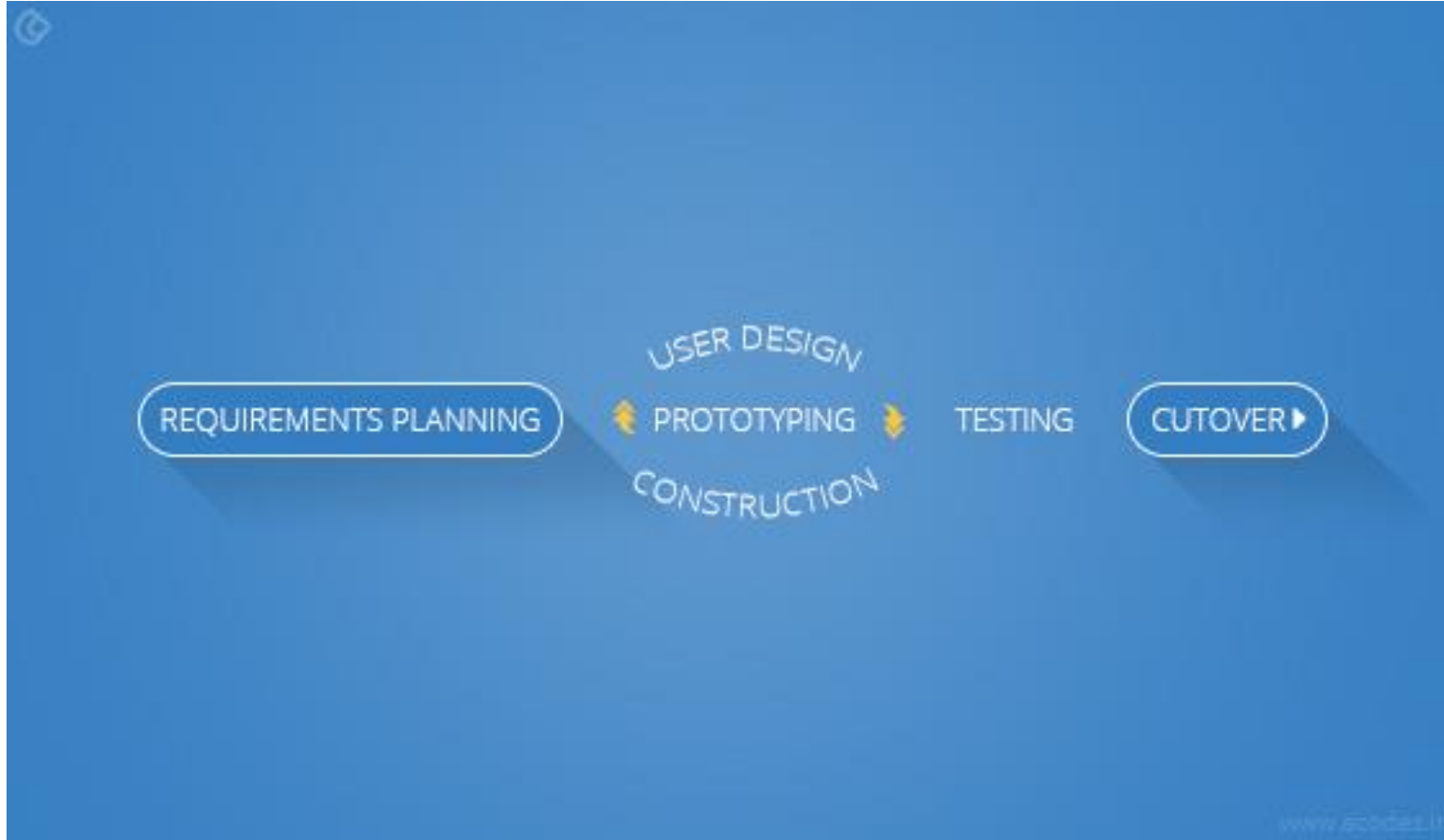
- **Cons:**

- Focuses on working with software and lacks documentation efficiency
- Chances of getting off-track as outcome are not clear

4. Rapid Application Development Methodology

- Aimed at providing quick results, [rapid application development](#) is meant to give excellent development processes with the assistance of other development approaches.
- It is created to take the maximum advantage from the development software.
- Undoubtedly, it is designed to augment the workability of the whole software development procedure for highlighting the participation of an active user.

4. Rapid Application Development Methodology



4. Rapid Application Development Methodology

- **Pros:**

- Makes the entire development process effortless
- Assists client in taking quick reviews
- Encourages feedback from customers for improvement

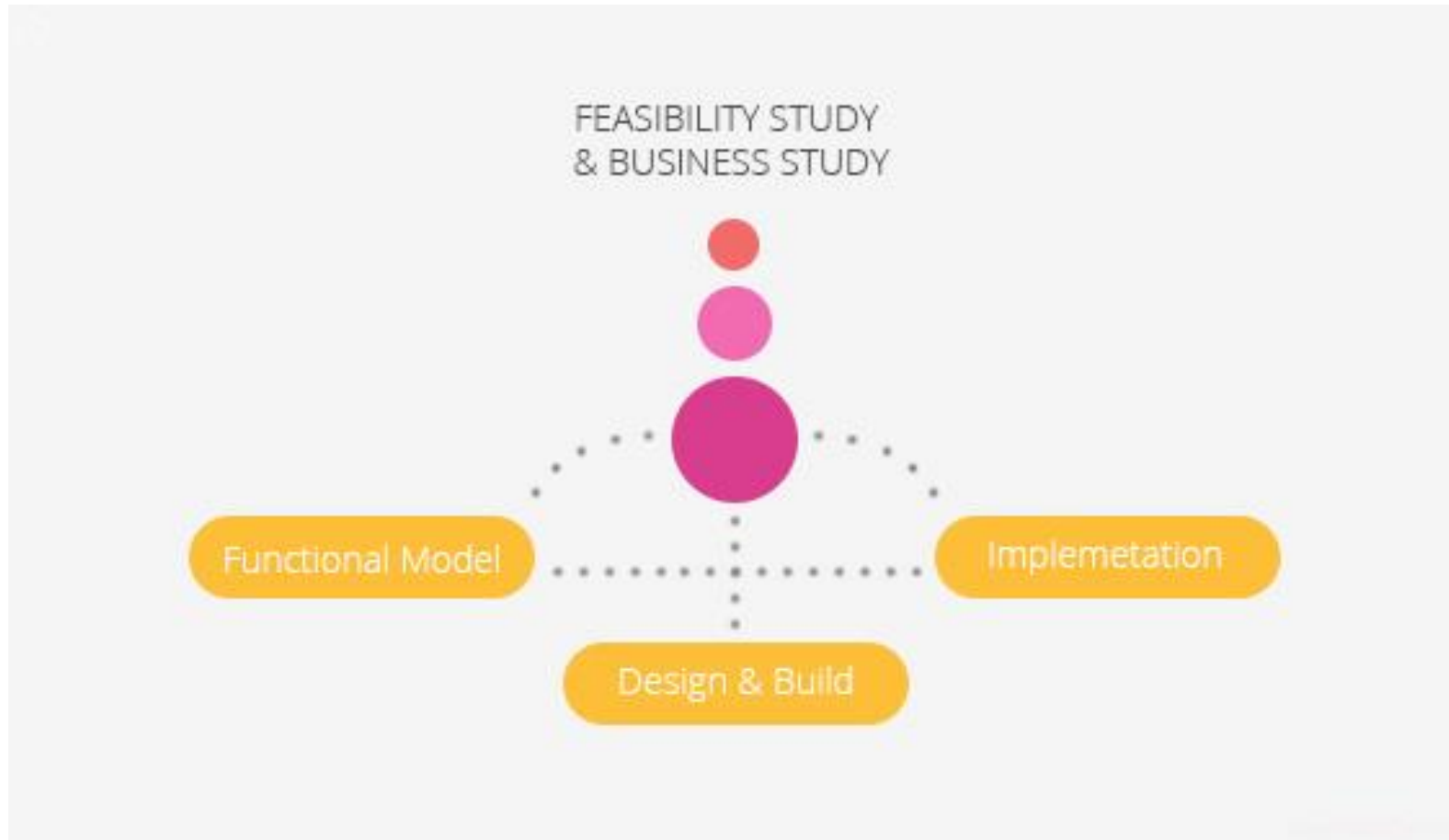
- **Cons:**

- Dependent on the team for performance
- Works on modularized system confined on this methodology
- Requires extremely skilled personnel to handle complexities
- Not applicable for the small budgeted projects

5. Dynamic System Development Model

- Authentically formulated and derived from the rapid application development methodology, it is an iterative and incremental approach that focuses on the involvement of the user.
- The task of [this methodology](#) is to provide software development systems within the specified time frame and the allocated budget.
- The very reason why it is quite in demand in the world of software development.

5. Dynamic System Development Model



5. Dynamic System Development Model

- **Pros:**

- Users getting a grip of the software development process
- Functionality deliverables are quick
- Offers easy access to end users by the developers

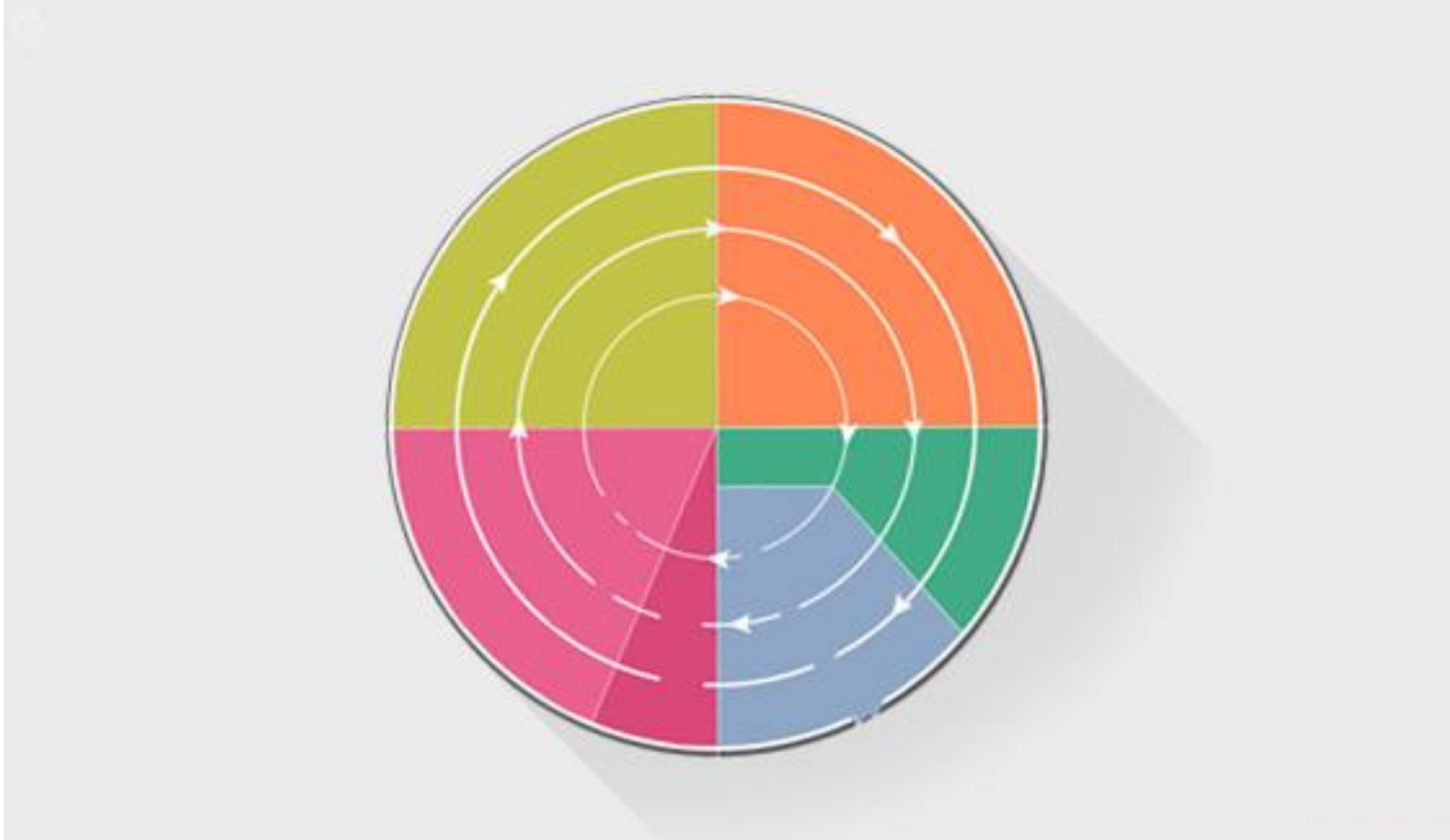
- **Cons:**

- This methodology is costly to implement
- Not suitable for small organizations

6. Spiral Model

- Being a highly sophisticated design, it is meant to reduce the early risks in the project.
- As per the process going by, the developers initiate on a smaller level and explore the included risks in it.
- Further to this, the developers are intended towards crafting a plan for iterating of the spiral.
- The accomplishment of any [Spiral Lifecycle model](#) is based on consistent, observant, and conversant management of the project.

6. Spiral Model



6. Spiral Model

- **Pros:**

- Risk factors are considerably reduced
- Excellent for large and complex projects
- Allows for additional functionality later
- Suitable for highly risky projects with varied business needs

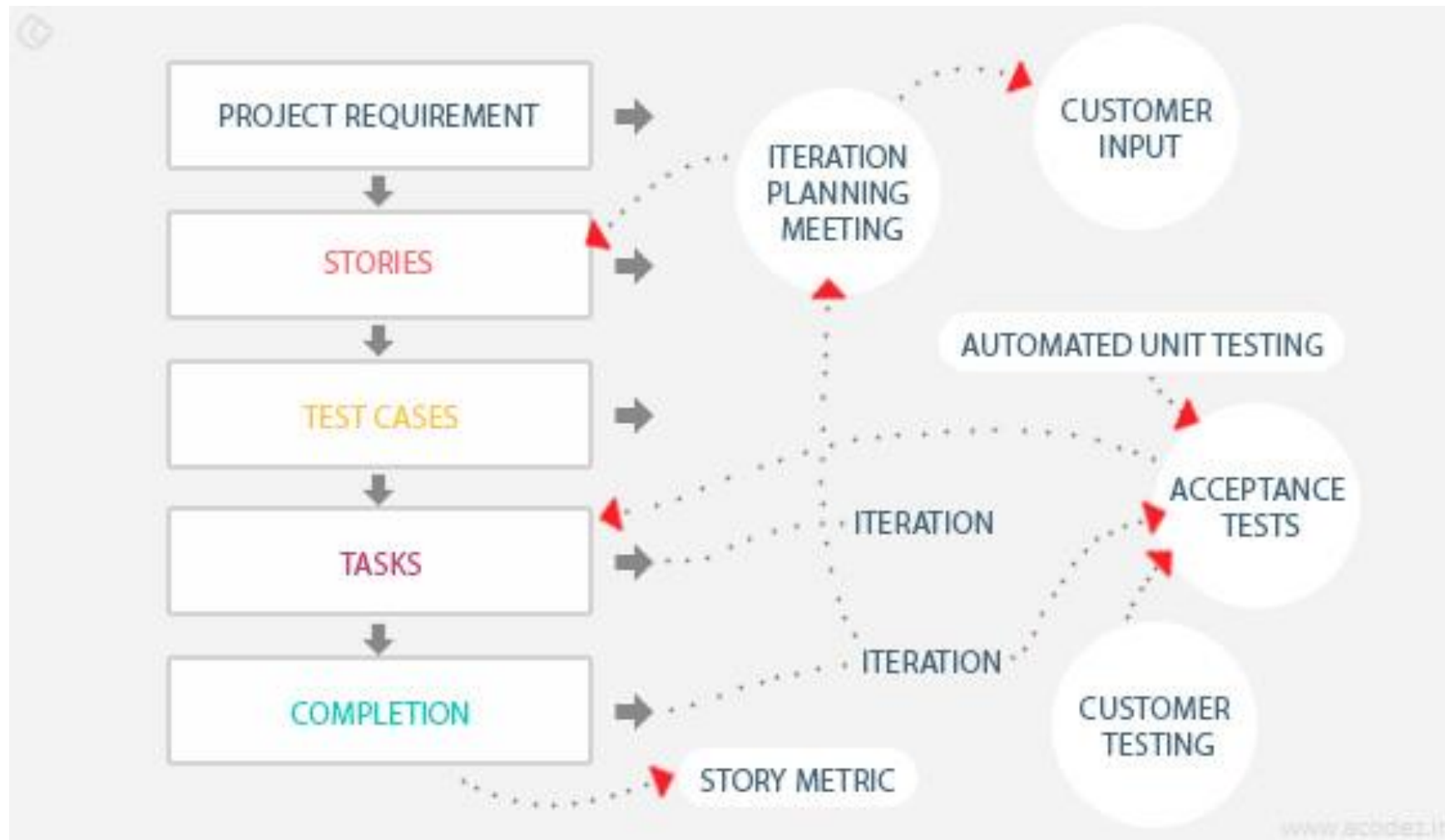
- **Cons:**

- Costly model in software development
- Failure in risk analysis phase may damage the whole project
- Not appropriate for low-risk projects
- Might get continued and never finish

7. Extreme Programming Methodology

- As an agile software engineering methodology, [extreme programming methodology](#) is presently known as XP methodology.
- It is chiefly used for crafting software within a very unbalanced atmosphere.
- It enables greater tractability within the modelling procedure.
- The foremost aim of this XP model is to reduce the cost of software essentialities.
- It is fairly mutual in the XP model that the price of altering the requirements on future stage in the project can be really whooping.

7. Extreme Programming Methodology



7. Extreme Programming Methodology

- **Pros:**

- It lays focus on customer involvement
- Establishes rational plans and schedules
- Developers are exceptionally committed to the project
- Equipped with modernistic methods for quality software

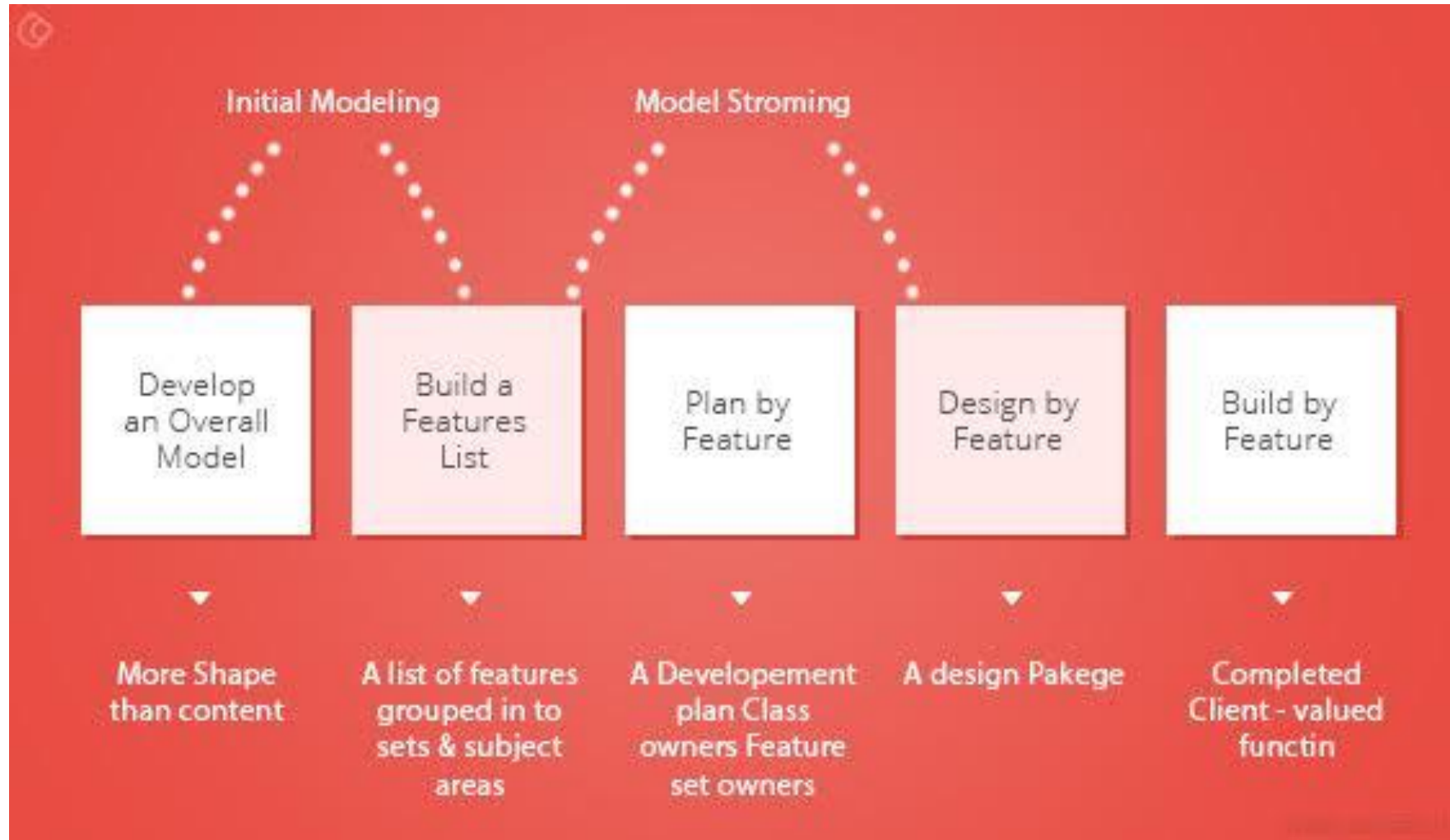
- **Cons:**

- Effectiveness depends on the people involved
- Requires frequent meeting for development raising total costs
- Necessitates for excessive development changes
- Exact possibilities and future outcomes are really unknown

8. Feature Driven Development

- Being an iterative methodology for software development, it is aimed at serving a large number of teams working on a project based on object-oriented technology.
- This sort of model is decent for companies that are passing on from a phase-based method to an iterative approach.
- It is already known as a [FDD methodology](#) and is highly functionable and creative enough to deal with varied complexities.

8. Feature Driven Development



8. Feature Driven Development

- **Pros:**

- Moves bigger projects with continuous success
- Easiest 5 procedures bring outcome in a better manner
- Built on pre-set standards of software development, it is programmed for easy development

- **Cons:**

- Not suitable for smaller projects and single developer
- Highly dependable on the leading developers, necessitating for the complete structure
- No written document provided to software owner

9. Joint Application Development Methodology

- The [Joint Application Development Methodology](#) is a requirements-classification and user-interface expansion approach that necessitates for the end-users, clients and developers attend a powerful off-site conference to accentuate and confirm software system.
- This methodology serves towards including the client in the design and expansion of an application.
- This is effortlessly proficient through a sequence of concerted workshops known as JAD sessions.
- It tends to lay emphasis on the business difficulty rather than methodical details.

9. Joint Application Development Methodology

- **Pros:**

- Allows for simultaneous congregation and alliance of excessive information.
- Produces huge amount of valuable information in short period
- Immediate resolving of differences with suitable assistance
- Provides forum to explore multiple points

- **Cons:**

- Takes excessive amount of time for planning and scheduling
- Requires significant investment of time and effort
- Calls for highly trained experts, which is tough to find

10. Lean Development Methodology

- As a technical advancement, [Lean Development model](#) lays emphasis on the formation of effortlessly manageable software.
- This exquisitely designed development technique is more deliberately engrossed than any other form of agile methodology.
- The objective of this procedure is to improve the software in one-third of the time, with very restricted budget, and very fewer amount of essential workflow.

10. Lean Development Methodology

- **Pros:**

- Lower budget & time requirements
- Allows for delivery of product early

- **Cons:**

- The workability of the team decides success of software development process
- Unsuitable business analyst can be severely problematic
- Excessive flexibility leads developer to lose focus

11. Rational Unified Process Methodology

- Smartly called as RUP, [Rational Unified Process methodology](#) powers software development using rational tools.
- This methodology segregates the expansion process into four different stages that each includes business modeling, scrutiny and design, enactment, testing, and disposition.
- This is an object-based and web-empowered program growth methodology.
- The model tends to assist software developer for stating guidelines, templates, and specimens for all features and stages of software development.

11. Rational Unified Process Methodology

- **Pros:**

- Lays high focus on precise documentation
- Removes project risks linked with client evolving needs
- Very less requirement for integration

- **Cons:**

- Needs excessively expert software developer
- Development procedure of the methodology is complicated
- Integration might cause confusion
- Very complicated to understand

12. Scrum Development Methodology

- SCRUM is the most widely preferred agile software development approach.
- (Likewise, KANBAN is a process that helps teams to collaborate and work effectively.)
- Basically, this excellent development is suitable for those development projects that are constantly altering or extremely developing requirements.
- The [Scrum Software development model](#) initiates with an ephemeral planning, conference and completes with a concluding review.
- This growth methodology is used for prompt development of software that happens to include a series of iterations to generate required software.
- It is a perfect approach because it effortlessly brings on track the deliberate progressing projects.

12. Scrum Development Methodology



12. Scrum Development Methodology

- **Pros:**

- Decision making lies in the hands of the team
- Business requirement document is considered insignificant
- Lightly controlled method empathizing with constant updating

- **Cons:**

- The processing method suffers because of wavering costs
- Not suitable for big sized projects
- Requires highly expert team, which has no place for novices

Summing Up!!!

- Technology has paved way for exclusive developments and software development procedures are not different.
- The main thing in this aspect is that it deals with a variety of complexities, which requires expert handling.
- Software development has specified methodologies that work on certain platforms, which allows them freedom to operate.
- This calls for high-quality performance under the guidance of professionals, who have years of experience in handling technical issues with efficiency.
- With varied forms of methodologies applicable to a different set of software development projects, the developers have loads of options to create excellently working software.

Summing Up!!!

- It is the world of technology that everyone is looking, and the constant changes have led to various software developments.
- It is the technique of creating useful software that adds value to the overall business procedure and creates ways for technical methodologies.
- The essential factor of developing high-quality software is that they simplify complex procedure; but, requires an extensive way of dealing with technicalities and expert knowledge.
- It is the support of experts that such software works efficiently; otherwise, they tend to spoil the entire process.