## Need for documentation

https://slideplayer.com/slide/1510323/

Presentation on theme: "How the customer explained it How the project leader understood it"— Presentation transcript:

[1](https://slideplayer.com/slide/1510323/5/images/1/How+the+customer+explained+it+How+the+project+leader+understood+it+How+the+team+designed+it.jpg) How the customerexplained itHow the project leaderunderstood itHow the teamdesigned it

[2](https://slideplayer.com/slide/1510323/5/images/2/What+the+team+created+How+marketing+described+it+How+the+project+was.jpg) **What the team created How marketing described it How the project was**  
documentedIntro to Project ManagementJeanne Sawyer

[3](https://slideplayer.com/slide/1510323/5/images/3/How+the+customer+was+billed.jpg) **How the customer was billed**  
What the customer reallyneededWhat the team deliveredHow the customer was billedIntro to Project ManagementJeanne Sawyer

[4](https://slideplayer.com/slide/1510323/5/images/4/Introduction+to+Project+Management.jpg) **Introduction to Project Management**  
Key ConceptsWhat is a ProjectWhat is Project ManagementWhy Project Management[Introduce self] I’m here to talk with you about project management. My intention is to give you an introduction to this key set of skills, and to leave you with a couple tools you can start using right away. We can also talk about how project management as a career option and pretty much anything else you want to discuss.Let’s start with why we care:Think about two projects that you’ve recently been involved in, one that was successful and one that wasn’t great.Now, let’s look at some of the things that made your successful projects successful:Success characteristics Trouble characteristicsProject management is about finding systematic ways to avoid the things that cause projects to fail, and encourage the things that help projects succeed.Intro to Project ManagementJeanne Sawyer

[5](https://slideplayer.com/slide/1510323/5/images/5/What+is+a+Project+Specific+Purpose+Defined+beginning+and+end.jpg) **What is a Project? Specific Purpose Defined beginning and end**  
Formal definitionAn endeavor intended to meet aSpecific Purposethat has aDefined beginning and endWhy is this important?Have you heard terms like:Cost overruns? Schedule overruns? Scope creep?Starting with clarity about purpose, start and end can help you avoid some classic problems that cause many projects to fail.Intro to Project ManagementJeanne Sawyer

[6](https://slideplayer.com/slide/1510323/5/images/6/What+is+Project+Management.jpg) **What is Project Management**  
The Triple ConstraintScopeTime (schedule)CostProject Management is all the things you do to make sure your projects are successful. We’ve learned the hard way that projects are more likely to be successful if you plan them carefully and make sure that you manage everything that goes on during the course of the project. Managing them means making sure the things we’ve identified as “success factors” are true for your project, and that the things that tend to cause project failure don’t happen.It makes things more manageable to think about and plan the various aspects of a project separately. First, we have the triple constraint.[explain]Scope: what you will do….Schwalbe, Information Technology Project Management, 2004, p.6Intro to Project ManagementJeanne Sawyer

[7](https://slideplayer.com/slide/1510323/5/images/7/Project+Management+Knowledge+Areas%3A.jpg) **Project Management Knowledge Areas:**  
Key competencies in addition to scope, time, costQualityHuman ResourcesCommunicationsRisksProcurementWhatever you do, do it on purpose—and write it down!Just managing the triple constraint isn’t enough, though. You have to deal with the other knowledge areas as well…[go through them]Thinking consciously about these areas and making explicit decisions about how you’ll handle each can make a big difference in project success (and also how painful it is to get there).Of course, all your decisions have to be written down so that you remember to do everything you decide and so you can communicate those decisions to everyone else involved.Intro to Project ManagementJeanne Sawyer

[8](https://slideplayer.com/slide/1510323/5/images/8/Why+Project+Management.jpg) **Why Project Management**  
Hmmm—sounds like a lot of work!Yes, but it improves chances of a successful projectMeets stakeholder needsOn timeWithin budgetProjects by their nature deal with uncertainty!Let’s get to the fun stuffThe why is really very basic: we’ve learned through experience that projects are more likely to be successful if we use project management techniques.Note that this definition of a successful project deals with all three aspects of the triple constraint. Note also, success for scope is “meets stakeholder needs”.Let’s take a project: Is taking this class a project?Who are your stakeholders for this class? [list on the board] What’s their role? [also write these on the board].It seems obvious that if project success is defined as meeting stakeholder needs, you’ll need to start by identifying who the stakeholders are. Keep in mind that not all stakeholders have the same role or same importance in a project. If you know who your stakeholders are and what their role is, your chances for meeting their needs is much better than if you just hope that whatever you do will work for them.Questions at this point about stakeholders?Back to scope: Why do we define scope as “meets stakeholder needs” rather than “meets specifications?”Does this say the scope can’t change during the project?What’s the difference between a scope change that is “good”, i.e., the project is successful, and a scope change that is “bad”, i.e., indicates some failure in the project management?PlanDoReDoFigure it out firstPlanDoIntro to Project ManagementJeanne Sawyer

[9](https://slideplayer.com/slide/1510323/5/images/9/Project+Management+Tools+%26+Tips.jpg) **Project Management Tools & Tips**  
Don’t assume—find out what’s realLet’s do an exercise:Write down:Your neighbor just bought a really expensive car. How much did it cost?You had to wait in line for a really long time. How long did you wait?Your teammate sent you an asking for a response “right away”. How quickly do you need to respond? (what if the was from your professor? What if it was from your boyfriend/girlfriend)[go around or have them come up]Intro to Project ManagementJeanne Sawyer

[10](https://slideplayer.com/slide/1510323/5/images/10/Project+Management+Tools+%26+Tips.jpg) **Project Management Tools & Tips**  
Start using these nowUse Team Ground Rules to define how you will workAttendance/ParticipationTeam Decision MakingOfficial RecordRolesOne way to avoid the assumptions trap is to talk and write things down.The point is to decide IN ADVANCE how you are going to work together, and avoid misunderstandings and assumptions. This is back to doing things on purpose. We can’t read each other’s minds, so don’t assume that you all have the same definition of, for example, what it means to return a phone call promptly—or even the same expectation that it’s necessary to return a phone call promptly.Think back to the two projects I had you identify at the beginning of the session, and why they were successful or not so successful. I’d bet that most of you have things like good teammates, worked well together and such for your successful projects and poor communication, teammates who didn’t do their work, didn’t show up, etc. as behind less successful projects. Ground rules can help prevent some of the misunderstandings that make projects no fun—and give you a tool to make it easier to talk about performance issues if they arise.Intro to Project ManagementJeanne Sawyer

[11](https://slideplayer.com/slide/1510323/5/images/11/Proj.+Mgmt.+Tips%3A+Define+the+Project.jpg) **Proj. Mgmt. Tips: Define the Project**  
If you don’t know where you’re going, any road will do.Determine the triple constraint parameters & balance for the projectScopeWhat you will doWhat you will NOT doPerformance requirements (quality)When you will finishResource constraintsCheck all assumptionsDocument it in the CharterGet stakeholder agreementStart defining the project by considering the end:What will be different when you finish?You may have already done at least some of this, but this might be a good time to go back and look at again and make sure your project definition is complete.Your Charter is not just for you and it’s not just for your sponsor (or professor)—it’s for all of you. It’s like a contract, and is a key tool in making sure you don’t have surprises at the end. [if time, do the Christopher Columbus thing]Intro to Project ManagementJeanne Sawyer

[12](https://slideplayer.com/slide/1510323/5/images/12/More+Project+Management+Tips.jpg) **More Project Management Tips**  
Make a detailed planWhat are the things you need to create for your projectWhat are the steps necessary to create themWho will do each oneWhen is each dueUse it!Rethink and adjust as you goLate projects get that way one day at a timeTalk about tendency to procrastinate. Who waits until the night before it’s due to write a term paper? What happens if you underestimate how long it will take you? What happens if you get interrupted by something you hadn’t planned on?There are special tools and techniques for some very organized ways of doing this, but you can help make your projects more successful right now with an well thought out to-do list and a calendar.Intro to Project ManagementJeanne Sawyer

[13](https://slideplayer.com/slide/1510323/5/images/13/Try+it+Next+Team+Meeting.+Write+down+what+the+final+deliverable+for+your+project+is+in+this+class..jpg) Try itNext Team MeetingWrite down what the final deliverable for your project is in this classOutline the steps you’ve taken and still need to take to get it doneFill in the detailsWorking backwards, estimate how long you need to complete each taskDecide who will do each one & assign due datesIf time, take 15 min. to start it. If not, tell them to do it at their next team meeting.Intro to Project ManagementJeanne Sawyer

[14](https://slideplayer.com/slide/1510323/5/images/14/Project+Management+Resources.jpg) **Project Management Resources**  
If you want to learn more nowLots of booksFred Brooks, Mythical Man MonthProject Management InstituteConsider joining now, as a studentIt’s cheaperBuild your networkLearn more about project managementProject Management is a great career optionQuestions about project management, career stuff….anything?Intro to Project ManagementJeanne Sawyer

## Why do projects fail

# Why do 70% of projects fail in IT?

**In the age of agile development, advanced technologies, DevOps, and project management techniques, it is difficult to believe that most software projects fail to perform. But unfortunately, it is true.**



Picture Credit: [Shutterstock](http://shutterstock.com/" \t "_blank)

The software failure rate is high whether it is a small or big project. To get better clarity, let’s have a look into some of the research reports that point to the project failure rate.

* The [Standish Group’s 2020 CHAOS report](https://www.projectsmart.co.uk/white-papers/chaos-report.pdf) estimated that around 66% of the software project failed.
* According to the [Standish Group 2015 CHAOS Report](https://www.infoq.com/articles/standish-chaos-2015/), out of all 50K projects in the study, 71% failed to meet these three criteria: on time, on budget, and with satisfactory results.  
  The problem is exacerbated in large projects. Medium-sized projects fail at a rate of 91 percent, while large projects fail at a rate of 94 percent.
* [A 2020 CISQ Report](https://www.synopsys.com/software-integrity/resources/analyst-reports/cost-poor-quality-software.html?cmp=pr-sig&utm_medium=referral) states that the cost of unsuccessful development projects among US firms is estimated to be $260B. The total cost of operational failure caused by poor quality software is approximately $1.56trillion.
* According to the 2017 Project Management Institute report, [14 % of the IT project fails](http://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf?sc_lang_temp=en). Here the number represents a complete failure. Of the remaining projects which didn’t fail, 31% didn’t meet the goals, 43% exceeded their initial budgets, and 49 % were late.

Previously, IT failure meant high-priced projects, and large-scale software implementations took a long time to develop and implement. Those failures can still occur today, but the likelihood is lower.

However, IT failures caused today are different from it was in the past. Even if you build the product with multiple functionalities within calculated time and budget, if it is unable to fulfill the customer’s expectations, that’s considered a failure.

Although advancements in software development technology have changed the nature of how the IT team handles projects, IT projects continue to fail in unexpected ways.

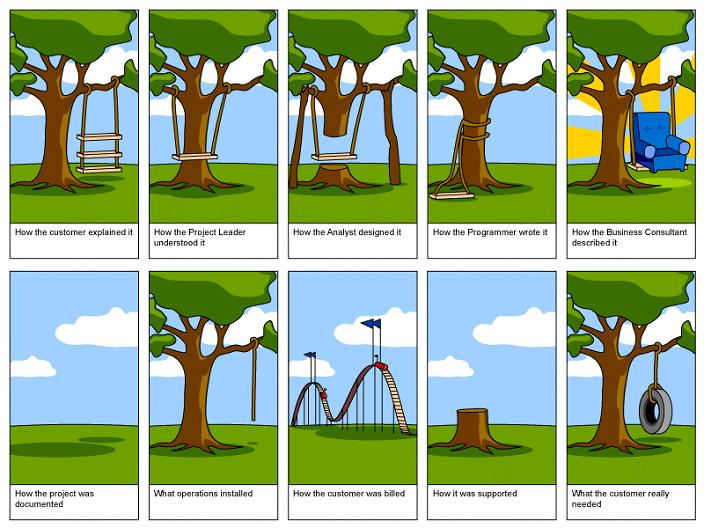
If you’re wondering why? Following are some of the most common reasons for IT project failure and how to avoid them.

## Common reasons for IT project failure

**1. Unrealistic user expectations**

The most common reason for project failure is a lack of understanding of what the end-user or client expects. It is often seen that customers or clients don’t know clearly what they want, how the software development process works, and how long it takes to deliver the final product.

Such clients might ask for the faster delivery of the high-quality project and that even at the lowest price. Naturally, the end result will be disappointing. Either the project exceeds its deadline and budget, or it fails to meet the expectations of the user.

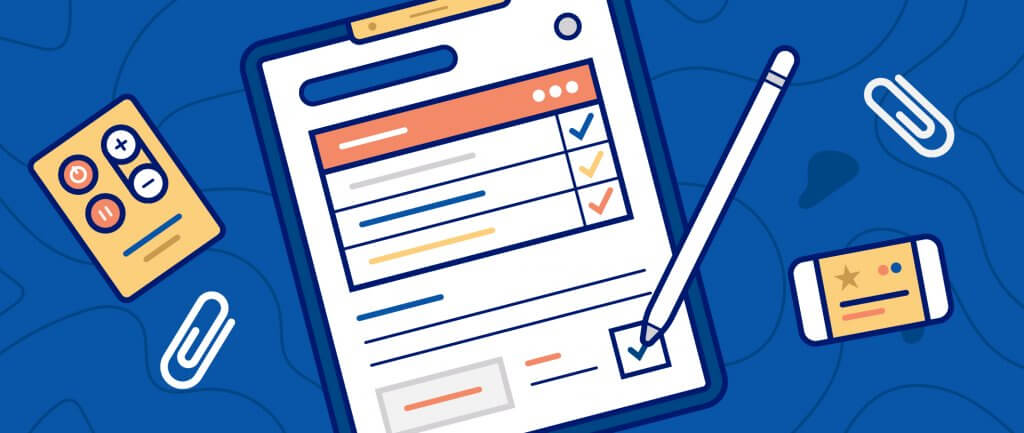


Other causes include a lack of coordination and understanding among the development team, decision-makers, stakeholders, and clients.

In any case, the project’s success must be realistic about what clients want, how much stakeholders can invest in the project (budgeting), and the project timeline during the initial stage of project planning.

**2. Poor technical documentation**

For software projects, reliable documentation is always a must. Not including the technical documentation for software architecture, requirements, testing, implementation, deployment, and technologies used can lead to confusion.



Having well-documented software projects simplify software development, maintenance, and knowledge transfer to the other developers and helps to improve the quality of software projects.

Also, without a guiding vision and a clear understanding of how a software project should be built, it is impossible to estimate how much a project will cost, its complexity, and the resources it will require.

**3. Unable to scale up rapidly**

If the software is unable to handle growing numbers of end-users due to unstable backend support, it can lead to IT project failure. It is mostly applicable to web and mobile applications.

A scalable and robust backend enables software to adjust according to the growing user traffic.



Furthermore, scalable software code allows developers to easily add new functionality to it rather than recreating it from scratch.

**4. Poorly defined project scope**

Defining the project scope helps you to focus on the important components of your project rather than wasting time on unnecessary things. Project scope creep occurs if the project parameter were not defined properly. Wasting time and resources on unnecessary tasks leads to project failure.



Poorly defined project scope often leads to,

* Wrong work completed
* Expectations missed
* Customers dissatisfaction
* Investments squandered

On the other hand, excessively limiting your project scope for a quick win can lead to complete project failure.

**5. Trying to make everyone happy**

It is impossible to create a sure-fire software project at once. While developing a software solution, always make sure, it should solve the core problem of the particular user group.

In the next versions of software, you can add other functionalities to support the primary function of your software.

Designing software to please everybody all the time is more complex and makes no sense. It only adds to the size of your software while consuming more resources and time. Such projects are difficult to maintain.

# Top 10 Reasons for Project Failure

By

**[Pavan Gumaste](https://www.whizlabs.com/blog/author/pavan/)**

* [**PROJECT MANAGEMENT**](https://www.whizlabs.com/blog/project-management/)

A project is assumed under project failure if it has not delivered its basic required expectations. Projects are generally started to meet a business ambition but it must also meet the stakeholder presumptions. If the project can meet both the business ambitions and all stakeholder presumptions, then it can be considered as a success.

However, the reality is not same as we expect it could be. As expectations of the customers and stakeholders keep on changing constantly, it becomes difficult for the project team to deal with issues and to achieve project goals. But the question is why project fails? Project failure occurs if the projects are the late, crossed budget, does not deliver the business value as expected or deliver the wrong product.

**“ *A study by Standish Group 2015 based on criteria of budget and time with the result for 50,000 projects around the world showed that only 29% of the projects were successful, 52% were challenged and 19% failed*.’’**

***Also Read:***[***10 Project Management Terms You Should Know***](https://www.whizlabs.com/blog/project-management-terms/)

## Top 10 Reasons for Project Failure

The reasons for project failure refer to the lack of presence of success factor for the project. Here are top 10 reasons for the project failure to answer the question why project fails.

#### 1. Inadequate Project Planning

Project planning is an important part of project management and it is the responsibility of the project manager to set a proper plan for the project. Project planning is generally used to organize distinct parts of a project including workload, project plan, management of team etc. You must have a clear vision of what you are going to do and know how to execute tasks to reach the project goal. But if the project is not planned properly then it may fail or doesn’t meet all the expectations of stakeholders and customers.

#### 2. Scope Creep

Scope creep refers to change in scope of the project and also known as requirement creep or feature creep. The scope is the work required for a project. Scope creep refers to how the requirements of a project keep on changing over a project lifecycle. It is caused by the changing requirements of the project stakeholders or due to miscommunication and dissidence. It may result in roadblocks, project delay or move over budget and hence project failure but sometimes it is not a bad thing. Scope creep is an actuality that every project manager predicts and plans for.

#### 3. Use of Unpracticed Tools and Techniques

Good tools and techniques are required for the success of a project. A common illusion is made by the project manager and team leader to utilize unpracticed tools and techniques at the starting of project. Sometimes, it can cause many severe problems during project life-cycle as the team has to deal with the learning curve of new techniques along with usual project duties. As team members are not familiar with the new techniques, the project may undergo deferment issues. For this, team members should use new tools in the combat successively.

#### 4. Shortage of Resources/ Requirements

Every project requires some resources according to need. The quantity of resources depends on the size and scope of the project. Sometimes, the project is inaccessible due to the shortage of resources and necessary requirements. This is one of the cases in which project success is not available. For example, if a project requires a skilled Java programmer and candidate appointed for the project do not have that skill then the project cannot continue until a necessary resource is provided.

#### 5. No or Poor Risk Management

At present, we have to deal with some actual facts in project management. Poor or no risk management has the ability to influence the project management. Project failure is the worst case of poor risk management. The goals are not reached and all the investment is wasted along with time and efforts that have been put into the project. Some of the most significant influences on the poor risk management are given as:

* Project failure
* Slow- running projects
* Risk of reputation damage
* Superfluity budget
* Unresponsive customers and less user adoption
* Inactive benefits

“Poor risk management is one of the primary contributors to project failure, and had a negative impact on project success.”

#### 6. Lack of User Engagement

A project which is aimed at developing some product is going to have especial patrons i.e. a group of people who does business at the establishment. Whether the product is a part of the software, a hardware widget or development of existing mechanism, users are the extreme customers. The user should be engaged during the project life- cycle as their suggestion and feedbacks are categorized to guide the project effectually. Failure to user engagement in the project leads to an improper end result product and project is failed to meet the user expectations.

Preparing for Project Management interview? Have a look at [*Top 12 Project Management Interview Questions*](https://www.whizlabs.com/blog/pm-interview-questions/) to crack the interview.

#### 7. Poor Controlling and Monitoring

Controlling and monitoring are the essential parts of project management. A project can succeed only when there is proper governance for the project management. Without proper planning and monitoring, the project may fail. The absence of controlling and monitoring impacts the project in many ways:

* Deviation in cost, scope, and schedule baselines
* Project may not be completed on time as expected
* Quality of the deliverables can be degraded
* Company/organization stature will damage
* Antagonism between project team can be raised.
* Poor project performance
* Unsatisfied customer

#### 8. Inexperienced Project Managers

Project failure is a common term that every project manager wants to rupture from. No one craves to take the responsibility of project failure as it may blot his career record. But if a project fails, then it simply means that project manager did a poor job. Ensure that the project manager has enough knowledge of what the best techniques are because hiring of well-skilled project manager can’t be omitted.

#### 9. Ineffective Communication

Project team knows their manager only through his communications. Whether it a project, an operation or personal life, communication plays an important role. Without communication, you are executing tasks in the murky area. Project managers should develop a communication plan. Even time-to-time meet-ups should be planned to discuss the project performance.

Project Management Institute (PMI) study revealed: “Ineffective communication had a negative impact on successful project execution.” And this issue even holds in 2017 after four years of the study.

#### 10. Poor Project Management

When there is no above-mentioned reason for project failure then the project may fail due to poor management. If a project is decently staffed, have a good WBS, proper planning, availability of resources, proper scheduling, and the support of sponsors but even gets fail then the only reason behind it is that project management was poor. The success of the project depends on the project manager and project team. Therefore, appoint right staff for your company/organization with significant experience.

Also, there should be a Project Management Office (PMO) that can monitor the project team efforts and provide the required support.

## Examples of Failed Projects in Project Management

**New Coke –**The launch of New Coke is a real-time example of the failed project in project management. The simple reason behind the failure of New Coke project was the good quality of old coke. Yes, nothing was wrong with the old coke, so the new old couldn’t replace or overtake it.

**McDonald’s Arch Deluxe Burger –**McDonald’s brought Arch Deluxe Burger in the market to go upscale but failed. It happened because it couldn’t attract adult range even with the high paid ad campaigns. The adults were not interested to pay considerably more amount for a little different product.

**Apple Lisa –**Apple Lisa is a popular example of a failed project in project management. Apple launched Lisa to target the business customers but the project couldn’t work as expected. The IBM PCs arrived with a lower price which didn’t allow Apple to seize much market share, and the project got failed.

Whether be a small or large scale project, it may undergo failure as the market is never predictable. Top 10 reasons of the project failure have been mentioned above while the failed projects can be traced to the few factors like Lack of interest, Scope changes, Poor communication, No bad news environment, and Lack of velocity.

## What is a Functional Requirement in Software Engineering? Specification, Types, Examples

## Ref: <https://www.guru99.com/functional-requirement-specification-example.html>

## Requirements graph.

## What is a Functional Requirement?

A **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called **Functional Specification**.

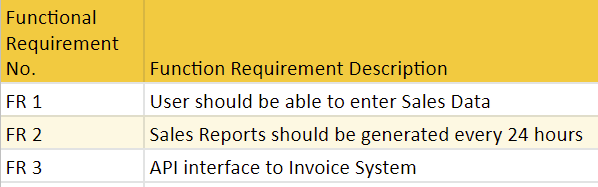
In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender’s necessity to detailed mathematical functional requirement specifications. [Functional software](https://www.guru99.com/learn-software-requirements-analysis-with-case-study.html) requirements help you to capture the intended behaviour of the system.

In this tutorial, you will learn more about:

* [What should be included in the Functional Requirements Document?](https://www.guru99.com/functional-requirement-specification-example.html#3)
* [Benefits of Functional Requirement](https://www.guru99.com/functional-requirement-specification-example.html#4)
* [Example of Functional Requirements](https://www.guru99.com/functional-requirement-specification-example.html#5)
* [Non-Functional vs. Functional Requirements](https://www.guru99.com/functional-requirement-specification-example.html#6)
* [Best practice of Functional Requirement](https://www.guru99.com/functional-requirement-specification-example.html#7)
* [Mistakes While Creating a Functional Requirement](https://www.guru99.com/functional-requirement-specification-example.html#8)
* [KEY LEARNINGS](https://www.guru99.com/functional-requirement-specification-example.html#9)

## What should be included in the Functional Requirements Document?

Here is how to write functional requirements document:

Example Functional Requirements

Functional Requirements of a system should include the following things:

* Details of operations conducted in every screen
* Data handling logic should be entered into the system
* It should have descriptions of system reports or other outputs
* Complete information about the workflows performed by the system
* It should clearly define who will be allowed to create/modify/delete the data in the system
* How the system will fulfill applicable regulatory and compliance needs should be captured in the functional document

## Benefits of Functional Requirement

Here, are the pros/advantages of creating a typical functional requirement document-

* Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
* A functional requirement document helps you to define the functionality of a system or one of its subsystems.
* Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
* Errors caught in the Functional requirement gathering stage are the cheapest to fix.
* Support user goals, tasks, or activities

## Types of Functional Requirements

Here are the most common functional requirement types:

* Transaction Handling
* Business Rules
* Certification Requirements
* Reporting Requirements
* Administrative functions
* Authorization levels
* Audit Tracking
* External Interfaces
* Historical Data management
* Legal and Regulatory Requirements

## Example of Functional Requirements

Below are the popular functional requirements examples:

* The software automatically validates customers against the ABC Contact Management System
* The Sales system should allow users to record customers sales
* The background color for all windows in the application will be blue and have a hexadecimal RGB color value of 0x0000FF.
* Only Managerial level employees have the right to view revenue data.
* The software system should be integrated with banking API
* The software system should pass [Section 508](https://www.section508.gov/) accessibility requirement.

## Non Functional vs. Functional Requirements

Here, are key differences between Functional and Nonfunctional requirements in [Software Engineering](https://www.guru99.com/what-is-software-engineering.html):

| **Parameters** | **Functional Requirement** | **Non-Functional Requirement** |
| --- | --- | --- |
| What it is | Verb | Attributes |
| Requirement | It is mandatory | It is non-mandatory |
| Capturing type | It is captured in use case. | It is captured as a quality attribute. |
| End result | Product feature | Product properties |
| Capturing | Easy to capture | Hard to capture |
| Objective | Helps you verify the functionality of the software. | Helps you to verify the performance of the software. |
| Area of focus | Focus on user requirement | Concentrates on the user’s expectation. |
| Documentation | Describe what the product does | Describes how the product works |
| Type of Testing | Functional Testing like System, Integration, End to End, API testing, etc. | Non-Functional Testing like Performance, Stress, Usability, Security testing, etc. |
| Test Execution | Test Execution is done before non-functional testing. | After the functional testing |
| Product Info | Product Features | Product Properties |

## Best practice of Functional Requirement

Important best practice for developing functional requirement document is as follows:

* Do not combine two requirements into one. Keep the requirements granular.
* You should make each requirement as complete and accurate as possible.
* The document should draft all the technical requirements.
* Map all requirements to the objectives and principles which contributes to successful software delivery
* Elicit requirements using interviews, workshops and casual communications.
* If there is any known, verified constraint which materially affects a requirement then it is a critical state that should be documented.
* It is necessary that you document all the assumption in the document.

## Mistakes While Creating a Functional Requirement

Here, are some common mistakes made while creating function requirement document:

* Putting in unjustified extra information that may confuse developers
* Not putting sufficient detail in the requirement document.
* You add rules or examples, scoping statements or objectives anything except the requirement itself.
* Left out a piece of important information that is an absolute must to fully, accurately, and definitively state the requirement.
* Some professionals start to defend the requirements they have documented when the requirement is modified, instead of finding the correct truth.
* Requirements which are not mapped to an objective or principle.

## KEY LEARNINGS

* Explain functional requirements in Software Engineering: A Functional Requirements define a system or its components
* Functional Requirements Document should contain Data handling logic and complete information about the workflows performed by the system
* Functional requirements along with requirement analysis help identify missing requirements
* Transaction corrections, adjustments, and cancellations, Business Rules, Certification Requirements, Reporting Requirements, Administrative functions, Authorization levels, Audit Tracking, External Interfaces, Historical Data management, Legal or Regulatory Requirements are various types of functional requirements
* As a good practice do not combine two requirements into one. Keep the requirements granular.
* Putting in unjustified extra information that may confuse developers should be avoided in the functional requirement document.

## Documentation Types and techniques

## Why do we need good requirement

Software requirements specifications (SRS) are the foundation of the pillars of software. They drive design, development, user experience, support documentation, and more. Yet, so often, they do not exist. Or do they?

https://blog.testlodge.com/importance-of-requirements/

The truth is, there are always requirements, but they often go undocumented. Every software has a purpose, therefor every software has requirements. But when they are not documented, testers are left to find the them their-selves.

This requires the tester putting on their detector hat and sifting through any existing clues and information to determine how exactly the software application should work. Some of these clues might be previous versions of the software or help documents. Other clues might be discussions with stakeholders and product owners. Either way, it’s going to take time.

## The risk in undocumented requirements

While testing can be done without requirements, there’s a risk and cost of not having them formally documented. The importance of requirements really spans across the entire team. Without any documented requirements, many assumptions are made during the development and testing phase. Developers and designers claim poorly functioning features are that way by design, and in general, things slip through the cracks.

Lets take a closer look at some of the things that can happen when requirements are not documented.

### Without requirements, testers don’t know what to test

Testers must make assumptions and spend time defining or looking for hidden requirements themselves. This essentially adds to the overall time and cost of the testing process.

### Developers don’t know what is considered “Complete”

Is it good enough if “delete” just simply deletes. Or does the delete functionality need to show a confirmation? Should it send an email notification? Should all users be able to perform a delete action? Without requirements, these decisions are going to be made somewhere along the line. Hopefully by the right person, and hopefully with the user in mind.

### Customers don’t know what to expect

Software requirements establish the agreement between your team and the customer on what the application is supposed to do. Without a description of what features will be included and details on how the features will work, the users of the software can’t determine if the software will meet their needs.

### Bugs can slip through the cracks

It’s not uncommon for bugs to be introduced due to unclear requirements or a misunderstanding of them. Make sure you’re not introducing requirements into the software from the start by ensuring your requirements are correct, complete, and communicated clearly.

## Conclusion

Taking the time up front to document requirements will save you and your team time further down the line. Requirements don’t always need to be extremely detailed documents but they should exist in some form. They are the document of record to make sure every one is on the same page.

## Requirement Engineering

**Requirements engineering (RE)** refers to the process of defining, documenting, and maintaining requirements in the engineering design process. Requirement engineering provides the appropriate mechanism to understand what the customer desires, analyzing the need, and assessing feasibility, negotiating a reasonable solution, specifying the solution clearly, validating the specifications and managing the requirements as they are transformed into a working system. Thus, requirement engineering is the disciplined application of proven principles, methods, tools, and notation to describe a proposed system's intended behavior and its associated constraints.

## Requirement Engineering Process

It is a four-step process, which includes -

1. Feasibility Study
2. Requirement Elicitation and Analysis
3. Software Requirement Specification
4. Software Requirement Validation
5. Software Requirement Management



### 1. Feasibility Study:

The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to established standards.

**Types of Feasibility:**

1. **Technical Feasibility** - Technical feasibility evaluates the current technologies, which are needed to accomplish customer requirements within the time and budget.
2. **Operational Feasibility** - Operational feasibility assesses the range in which the required software performs a series of levels to solve business problems and customer requirements.
3. **Economic Feasibility** - Economic feasibility decides whether the necessary software can generate financial profits for an organization.

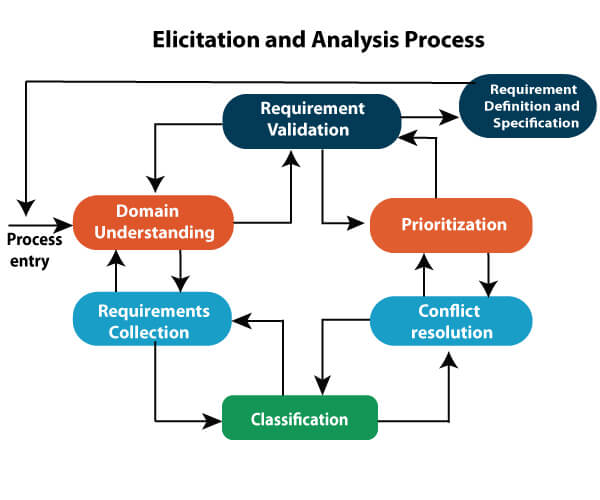
### 2. Requirement Elicitation and Analysis:

This is also known as the **gathering of requirements**. Here, requirements are identified with the help of customers and existing systems processes, if available.

Analysis of requirements starts with requirement elicitation. The requirements are analyzed to identify inconsistencies, defects, omission, etc. We describe requirements in terms of relationships and also resolve conflicts if any.

**Problems of Elicitation and Analysis**

* Getting all, and only, the right people involved.
* Stakeholders often don't know what they want
* Stakeholders express requirements in their terms.
* Stakeholders may have conflicting requirements.
* Requirement change during the analysis process.
* Organizational and political factors may influence system requirements.



### 3. Software Requirement Specification:

Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources - the requirement received by the customer written in ordinary language. It is the job of the analyst to write the requirement in technical language so that they can be understood and beneficial by the development team.

The models used at this stage include ER diagrams, data flow diagrams (DFDs), function decomposition diagrams (FDDs), data dictionaries, etc.

* **Data Flow Diagrams:** Data Flow Diagrams (DFDs) are used widely for modeling the requirements. DFD shows the flow of data through a system. The system may be a company, an organization, a set of procedures, a computer hardware system, a software system, or any combination of the preceding. The DFD is also known as a data flow graph or bubble chart.
* **Data Dictionaries:** Data Dictionaries are simply repositories to store information about all data items defined in DFDs. At the requirements stage, the data dictionary should at least define customer data items, to ensure that the customer and developers use the same definition and terminologies.
* **Entity-Relationship Diagrams:** Another tool for requirement specification is the entity-relationship diagram, often called an "**E-R diagram**." It is a detailed logical representation of the data for the organization and uses three main constructs i.e. data entities, relationships, and their associated attributes.

AD

### 4. Software Requirement Validation:

After requirement specifications developed, the requirements discussed in this document are validated. The user might demand illegal, impossible solution or experts may misinterpret the needs. Requirements can be the check against the following conditions -

* If they can practically implement
* If they are correct and as per the functionality and specially of software
* If there are any ambiguities
* If they are full
* If they can describe

**Requirements Validation Techniques**

* **Requirements reviews/inspections:** systematic manual analysis of the requirements.
* **Prototyping:** Using an executable model of the system to check requirements.
* **Test-case generation:** Developing tests for requirements to check testability.
* **Automated consistency analysis:** checking for the consistency of structured requirements descriptions.

### Software Requirement Management:

Requirement management is the process of managing changing requirements during the requirements engineering process and system development.

New requirements emerge during the process as business needs a change, and a better understanding of the system is developed.

The priority of requirements from different viewpoints changes during development process.

The business and technical environment of the system changes during the development.

## Prerequisite of Software requirements

Collection of software requirements is the basis of the entire software development project. Hence they should be clear, correct, and well-defined.

A complete Software Requirement Specifications should be:

* Clear
* Correct
* Consistent
* Coherent
* Comprehensible
* Modifiable
* Verifiable
* Prioritized
* Unambiguous
* Traceable
* Credible source

## The importance of requirements management

The Internet of Things (IoT) is changing not only the way products work, but their design and development. Products are continuously becoming more complex with more lines of code and additional software — some of which allow for even greater connectivity. With requirements management, you can overcome the complexity and interdependencies that exist in today’s engineering lifecycles to streamline product development and accelerate deployment.

Issues in requirements management are often cited as major causes of project failures.   
Having inadequately defined requirements can result in scope creep, project delays, cost overruns, and poor product quality that does not meet customer needs and safety requirements.

Having a requirements management plan is critical to the success of a project because it enables engineering teams to control the scope and direct the product development lifecycle. Requirements management software provides the tools for you to execute that plan, helping to reduce costs, accelerate time to market and improve quality control.

Requirements Quality Assistant Brings Watson AI to Requirements Management (2:22)

Requirement management planning and process

**Requirements management plan (RMP)**A requirements management plan (RMP) helps explain how you will receive, analyze, document and manage all of the requirements within a project. The plan usually covers everything from initial information gathering of the high-level project to more detailed product requirements that could be gathered throughout the lifecycle of a project. Key items to define in a requirements management plan are the project overview, requirements gathering process, roles and responsibilities, tools, and traceability.

**Requirements management process**When looking for requirements management tools, there are a few key features to look for.

A typical requirements management process complements the systems engineering V model through these steps:

* Collect initial requirements from stakeholders
* Analyze requirements
* Define and record requirements
* Prioritize requirements
* Agree on and approve requirements
* Trace requirements to work items
* Query stakeholders after implementation on needed changes to requirements
* Utilize test management to verify and validate system requirements
* Assess impact of changes
* Revise requirements
* Document changes

By following these steps, engineering teams are able to harness the complexity inherent in developing smart connected products. Using a requirements management solution helps to streamline the process so you can optimize your speed to market and expand your opportunities while improving quality.

**Digital requirements management**Digital requirements management is a beneficial way to capture, trace, analyze and manage requirements changes. Digital management ensures changes are tracked in a secure, central location, and it allows for strengthened collaboration between team members. Increased transparency minimizes duplicate work and enhances agility while helping to ensure requirements adhere to standards and compliance.

**Requirements attributes**In order to be considered a “good” requirement, a requirement should have certain characteristics, which include being:

* Specific
* Testable
* Clear and concise
* Accurate
* Understandable
* Feasible and realistic
* Necessary

Sets of requirements should also be evaluated and should be consistent and nonredundant.

Benefits of requirements management

Some of the benefits of requirements management include:

* Lower cost of development across the lifecycle
* Fewer defects
* Minimized risk for safety-critical products
* Faster delivery
* Reusability
* Traceability
* Requirements being tied to test cases
* Global configuration management

## Sources of requirement

1. **Stakeholders/Buyers/customers :**  
   They are the persons responsible for accepting and executing the software. They can be individual individuals, organizations, trusts or even the government or public of a country.
2. **Operators/End users :**  
   They are the persons who work on the software to make the services of the software available to its beneficiaries or the end users.
3. **User/Beneficiaries :**  
   These are the users of the product for which the product is intended.
4. **Developer / Development team:**  
   The software engineering responsible for developing the software to make it provide the expected services. They are responsible for software design, prototype development, and technical feasibility. They work closely with the end-users, buyers, and application experts.
5. **Automated Tools :**  
   In the new generation of information technology and software development paradigm, many automated and semi-automated tools are available that allow for the affirmation and management of the need for building the system. such software also provides input. System/software requirements.
6. **Past Experience/Case Studies :**  
   An organization working in the similar or same domain may provide its past experience or even documented case studies. This helps have a clearer picture of the requirements, which may otherwise be left hidden.
7. **Connected People/Machine/Environment :**  
   People associated with software or environmental factors and IT domain may give a lot of provide information about constraints involved in development, development, its and environment implications on software.
8. **Tester :**  
   Testers are a good source of information about the user’s behavior or the predictive behavior of the system’s condition. continuous contact with real users for their input. In such cases, examiners may use their experiences and analytical skills to provide input.
9. **Domain experts :**  
   They are professionals with experience and expertise of the domain in which the software provides its services, viz. insurance, financials, banking, communication, data transfer, networking, etc. Domain experts unwind the hidden or unseen probable requirements or risks involved in product development.

## What is a use case

**What Are Use Cases?**

Have you ever been disappointed when a new software release does not include that one common feature all users want? Perhaps you were part of planning a new system and could not figure out why the final product was all botched up? If so, understanding how use cases improve your business may be beneficial.

A **use case** is a description of how a person who actually uses that process or system will accomplish a goal. It's typically associated with software systems, but can be used in reference to any process. For example, imagine you're a cook who has a goal of preparing a grilled cheese sandwich. The use case would describe through a series of written steps how the cook would go about preparing that sandwich. A use case helps you understand where errors could occur in the process and design features to resolve those errors.

Three elements that a use case must contain:

1. **Actor**, which is the user, which can be a single person or a group of people, interacting with a process
2. **System**, which is the process that's required to reach the final outcome
3. **Goal**, which is the successful user outcome

Additional elements that are included in a complex use case:

1. **Stakeholders**, which are those who have an interest in how the system turns out, even if they aren't direct users
2. **Preconditions**, which are things that must be true before a use case is run
3. **Triggers**, which are events that occur for a use case to begin
4. **Example**
5. On Saturday mornings at 11:30, the cook prepares grilled cheese sandwiches at the local cafe. The cook would be the primary actor using this system. The system contains a frying pan, a cooking device, bread, cheese, butter, a spatula, and a plate. The goal is to create the perfect grilled cheese sandwich for the visitors to the cafe during the lunch hour.
6. A use case is a list of steps that illustrate how a process will be carried out in a system. The document walks you through the steps the actor will take to achieve a goal. A use case is written by a business analyst who meets with each user, or actor, to write out the explicit steps in a process.

|  |
| --- |
| A use case is a textual document that outlines the steps needed to complete a goal in a process. |

# Login Model - Use Case Example

## 

# Use Case Diagram for ATM Machine

## 

# Use Case for College Admission

## 

## Use case for movie Ticket booking

## 

## Software development models

Software development models have been first introduced to the software development community no sooner than in the 1950s-1960s. At that time, it was quite a good practice to structure everything and programming was one of the processes that had to be planned.

Today, this process is much needed in everyday project management as without a certain flow of events, project development can be quite chaotic. Therefore, any activity from project planning and up to its release and maintenance has a conceptual framework everyone needs to use.

So, this article will be useful for you if you:

* Struggle with the undefined process of software development
* Lack certain knowledge about the project management techniques
* Search for a way to optimize and structure the development phase
* Want to accelerate the delivery process
* Reduce project risks
* Shorten the project timeline and excessive costs

Let’s define the software development life cycle, make an overview of the seven potentially used models, and find out their differences and benefits in project development. Moreover, let’s make a comparison of the software models in software engineering to find out which one might suit you best!

**Why are the software project development models needed?**

Every software outsourcing company has its [software development lifecycle (SDLC)](https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc), which usually encompasses the following [phases of app development](https://inoxoft.com/stages-of-app-development/):

* requirement analysis
* design
* development and testing
* implementation
* documentation
* maintenance

## 7 Software Development Models Comparison | Inoxoft.com

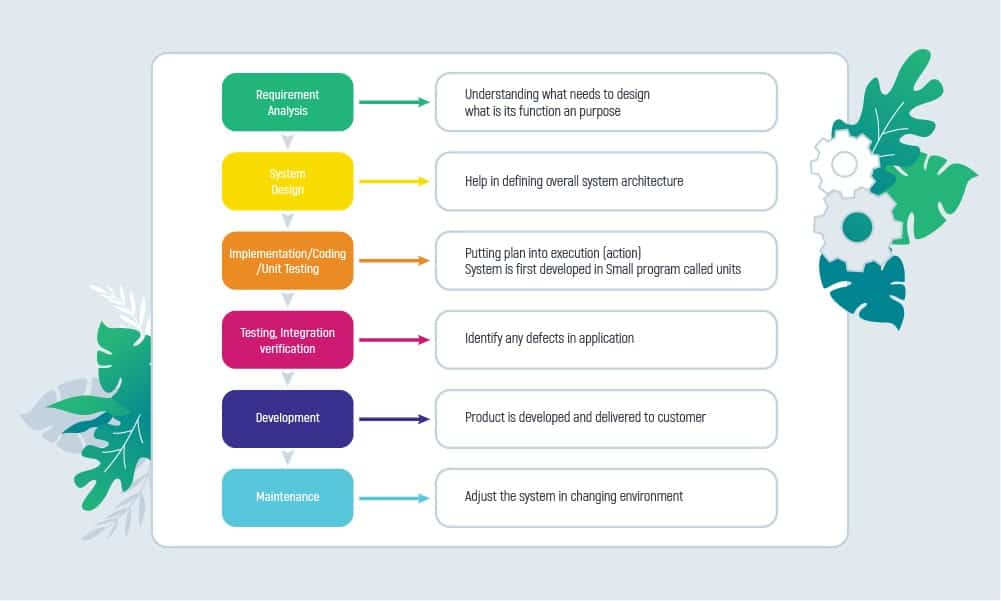
These stages mark the usual process of software project development from gathering client requirements to launching a fully-fledged product with 24/7 support. At present, companies prefer a personalized approach to their workflow, so there are several conceptual models that include these development phases. The models have their key principles, basic flows, and timelines pre-written and established, and, for this reason, we call them methodologies.

Currently, there are seven featured methodologies to pay attention to. These are:

* Waterfall Model
* Iterative and Incremental Model
* Spiral Model
* V-shaped Model
* Prototype Model
* Agile Model
* DevOps Model

### 1. Waterfall Model

The Waterfall model is the first of the software project development models that have been introduced in the software development circles. It is also called linear as the model implies each stage to be completed before going on to the next one. So, the model goes linearly step by step giving no possibilities to overlap any stage. Schematically, it would look like this:

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NTk0OjEzMQ==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NTk0OjEzMQ==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

Genuinely, the model resembles a real-life waterfall that has a beginning and an end with water falling freely from the cliff without being interrupted or altered.

The Waterfall model is applied to the software development life cycle when it is most suitable for a certain project.

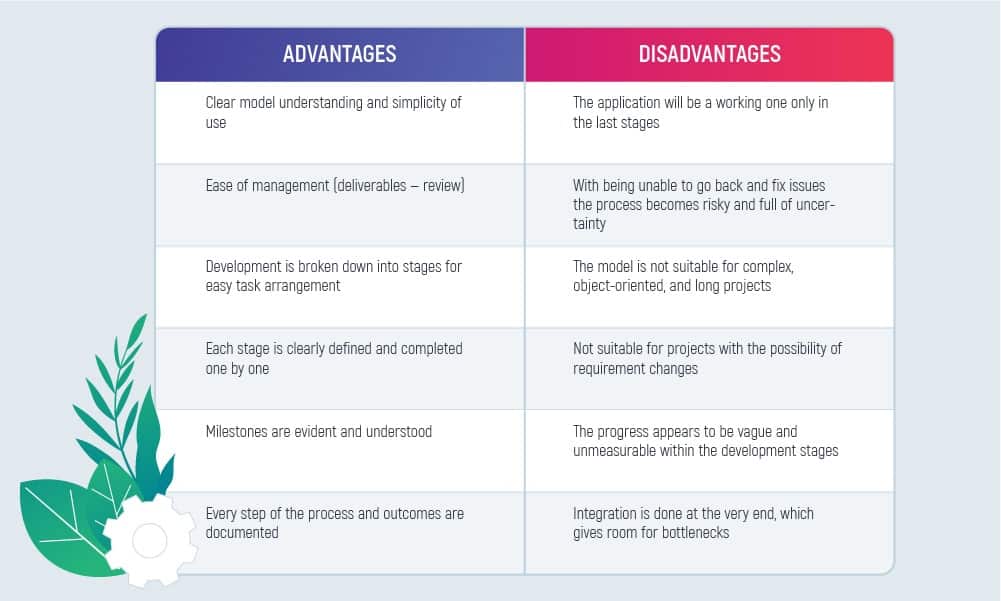
#### What factors contribute to choosing the Waterfall approach?

* Short-term projects
* Clear, fixed, and well-documented requirements
* Static technology that is understood
* Stable product definition
* A variety of resources for product support
* Low Product Owner Involvement
* Strictly-set timeline
* Fixed budget that cannot be changed

In all the other situations choose the other model to approach your project development.

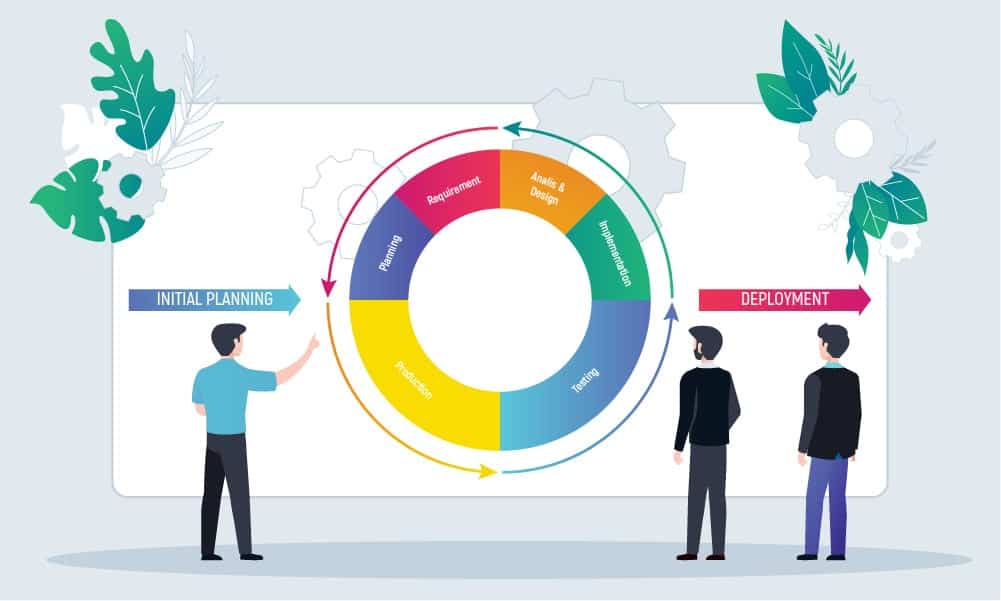
The main advantage of Waterfall is that the model is in strict order with regards to the tasks carried out and their timelines. Due to being fixed, the tasks can be easily divided between software departments and monitored as to task completion. Clients may find the Waterfall Model a reliable approach as they will be aware of the whole process from the start till finish beforehand. As for the disadvantages of the Waterfall Model, there are some as well. For instance, Waterfall has no revision of a task. The farther the application development goes forward into the other stage the harder it is to go back and change its conceptual basics or documentation.

Waterfall Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjA5OjEzMQ==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjA5OjEzMQ==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 2. Iterative and Incremental Model

In the Iterative model, the development starts with implementing several small requirements, and then the product is enhanced with the help of numerous iterations (incrementally). The process of enhancement goes on in increments (iterations) until the application features are complete and it is ready to be deployed. Developing only a small part of the software is a smart approach as you can easily review what was done, quickly identify further requirements and proceed with the implementation or provide some changes to the app and then proceed. Either way, you’ll benefit from the solutions as every new iteration will bring you a new and better version of the software.

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjEyOjEzNg==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjEyOjEzNg==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

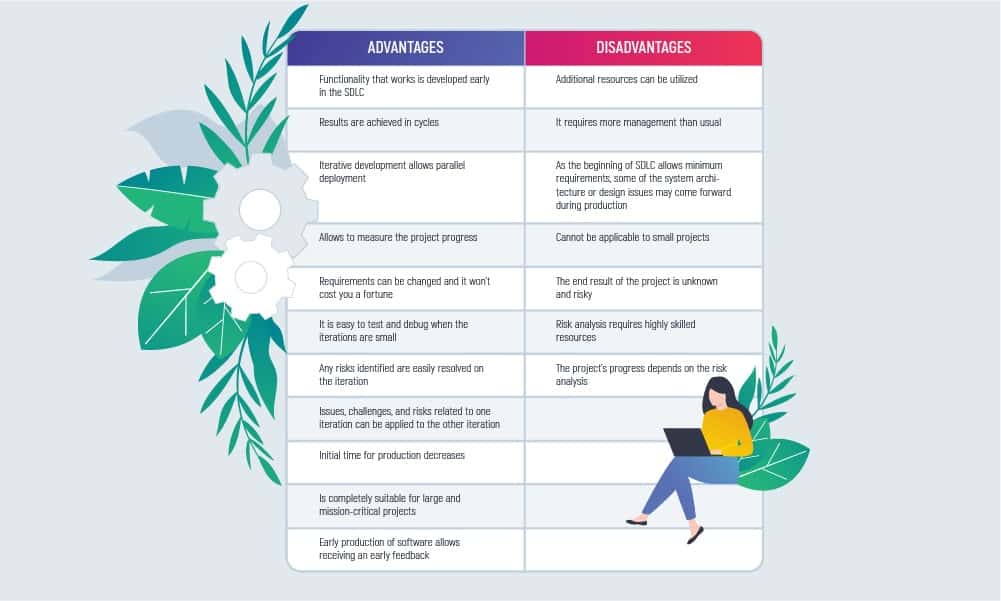
Unlike the Waterfall model, the Iterative Model allows carrying out multiple iterations at a time and receives the product output simultaneously. The key to success, which the model offers, is in validating the versions of the software and testing them according to the requirements. This allows a better understanding of where the team is at with the project, the product’s functionality verification, and 100% confidence that the product will be a success.

#### Where is the Iterative Model applicable in software development?

* System requirements are clearly defined and fully understood
* Additional requirements can occur with time
* When the team learns new technology and works on the project simultaneously
* Specific resources are to be used only when they are required for specific iterations according to the contract
* When the business goals might change in the future due to high-risk features

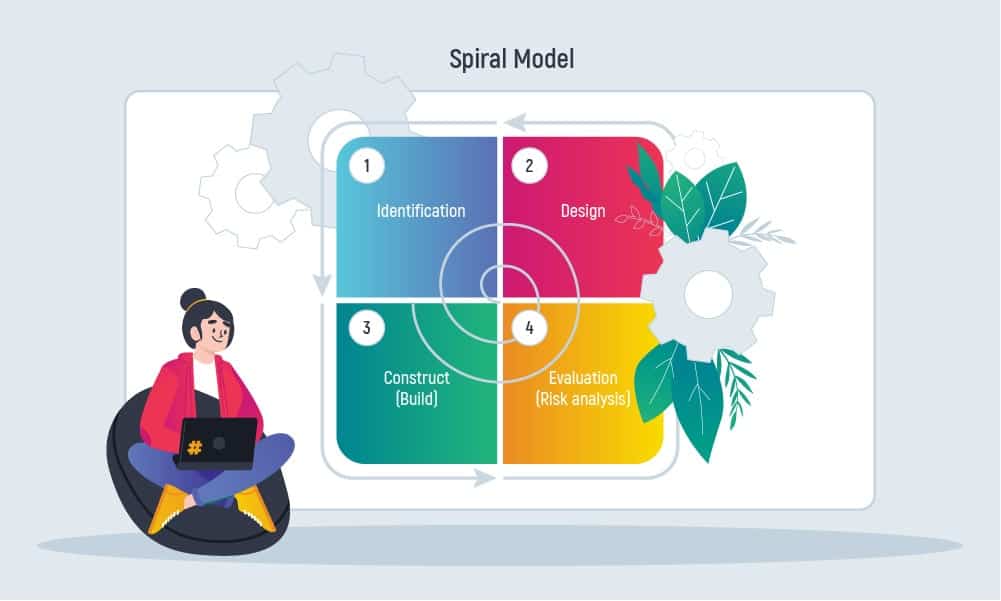
The biggest advantage of the Iterative model is that it allows for finding feature and design flaws at the beginning of development. Working in iterations and validating every development cycle will result in a risk-free product with the ability to add or remove features as well as alter the costs and timeline of the final product. Besides advantages, we should talk about some of the minuses that are also present.

Iterative Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjIyOjEzNg==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjIyOjEzNg==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 3. Spiral Model

The Spiral Model is a combination of the Iterative and Waterfall models. From the Iterative methodology, it took the cycled development, and from the Waterfall one, it borrowed a systematic approach and the ability to control the process. The biggest emphasis of the current model falls on risk analysis. Also, it delivers a project in iterations, which swirl around the spiral.

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjI1OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjI1OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

The Spiral Model is designed in 4 stages. The product under development passes all of these stages during its iterations, which are also called spirals in this methodology. The stages include:

* Identification
* Design
* Construct (Build)
* Evaluation (Risk analysis)

Identification is the stage where it is best to gather business requirements. This is done in a basic spiral. The following spirals identify system, subsystem, and unit requirements. Besides gathering requirements, this stage is all about communication between the customer and the system analyst to understand what is required to deploy.

The design also starts in the basic spiral with the conceptual design and includes subsequent spirals with architectural design, logical design of modules, physical product design, and the final design.

The Construct (Build) stage is the development of the actual software product at each and every spiral. In the basic spiral, when the concept needs to be evaluated according to its feasibility, you should develop a POC (Proof of Concept) and try to obtain valuable customer feedback.

Getting aware of all the requirements and design details, we produce a numbered software working model (build) version. The customer reviews all the builds and gives feedback.

Evaluation (Risk Analysis) is a process of identifying, estimating, and monitoring the technical feasibility and management risks. For example, schedule overdue and cost overrun.

After the customer has evaluated the build, the next iteration comes forward. This one follows the linear approach to ensure the feedback given by the customer is implemented. The SDLC includes these spiral iterations practically until its finalization.

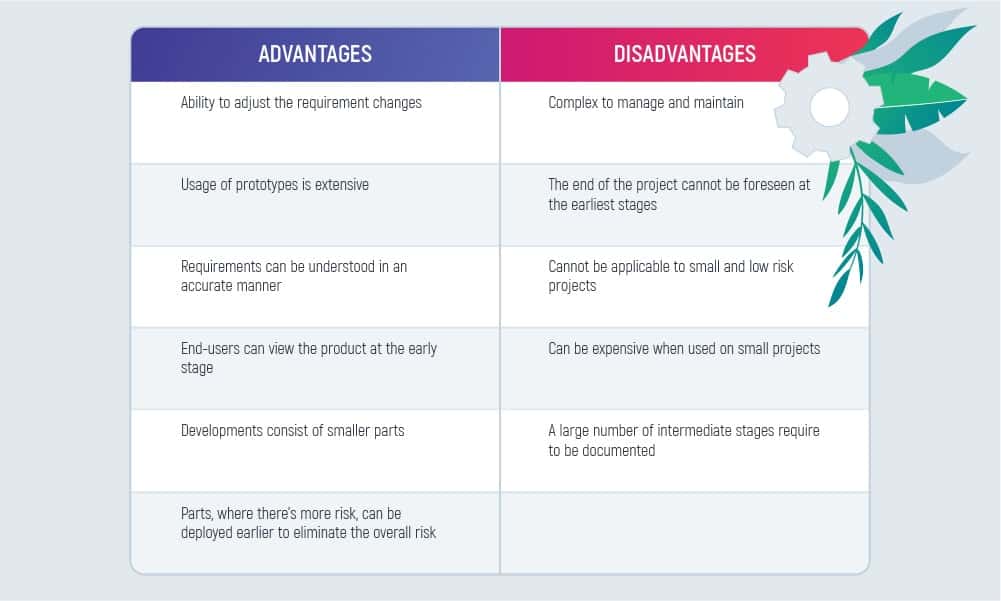
#### When is it best to use the Spiral Model?

The Spiral Model is applied when there is a need to “learn with the product’s maturity” and:

* There’s a budget limit and you need to foresee risks
* You have a medium or high-risk project to carry out
* There is a long-term project, and changes might occur
* The requirements are vague to the customer
* Requirements need more clarity due to their complexity
* The primary product has to be released in several stages to get valuable customer feedback
* You expect major changes to the product during the SDLC

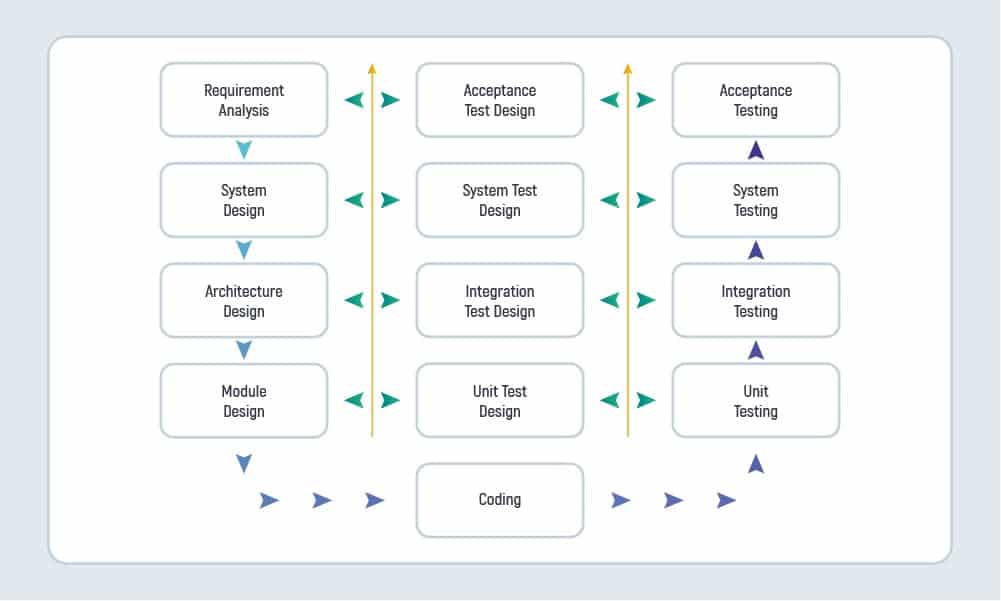
Based on the applicability of the Spiral Model, let’s view its benefits and drawbacks. The main advantage of the model is that you can add elements to the product in case they become available or known. Also, the Spiral Model involves users at the early stages of product development, which allows receiving constructive feedback.

Spiral Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjQ4OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjQ4OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 4. V-shaped Model

The V-model allows a project to be carried out in a sequence and according to the V shape. The other name of this model is the Verification and Validation Model because it is based on the Waterfall Model. However, here, in the SDLC, every stage of development includes testing as well. So, you carry out a phase, complete it, test it and only then go to the next phase that will look identical to the previous one with regards to its structure.

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjUxOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjUxOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

In the V-shaped Model, the development phase corresponds to the testing phase and they go parallelly. This means you can see the Verification phases on one side of the image and the Validation phases on the other. These sides are being joined with the Coding Phase.

Based on the picture above, there are 4 Verification phases. These are:

* Requirements analysis
* System design
* Architectural design
* Module design

The validation phase also includes 4 stages:

* Unit testing
* Integration testing
* System testing
* Acceptance testing

The Coding Phase, as it is seen in the picture, combines the two other stages and makes a V shape.

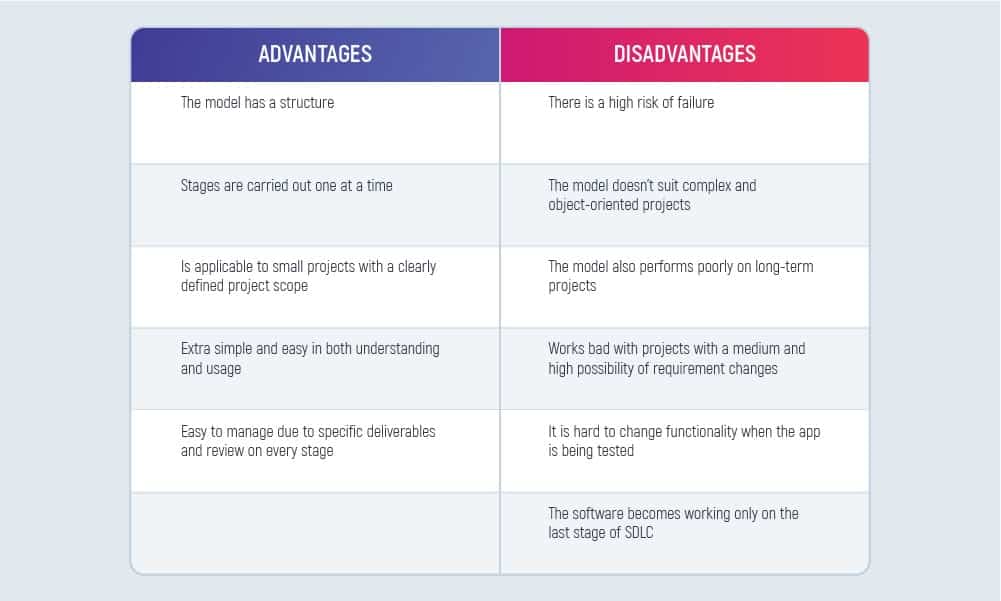
The V-shaped model has the same applicability as the Waterfall Model. Requirements have to be clearly defined from the very beginning of the SDLC as going back and making changes will increase the product development costs. This model suits best the healthcare industry.

#### When to implement the V-Shaped Model?

* There are clearly defined, documented, and stable requirements
* The technology used is also stable and the project team has a full understanding of it
* All of the requirements are extra clear
* The project is short-term and not complex

The best advantage of the V-shaped Model is that you can both understand, apply to, and manage it in an SDLC easily. However, on the contrary, the model lacks flexibility when changes are on the horizon and it is quite expensive for the customer to implement those changes as they trigger additional development costs.

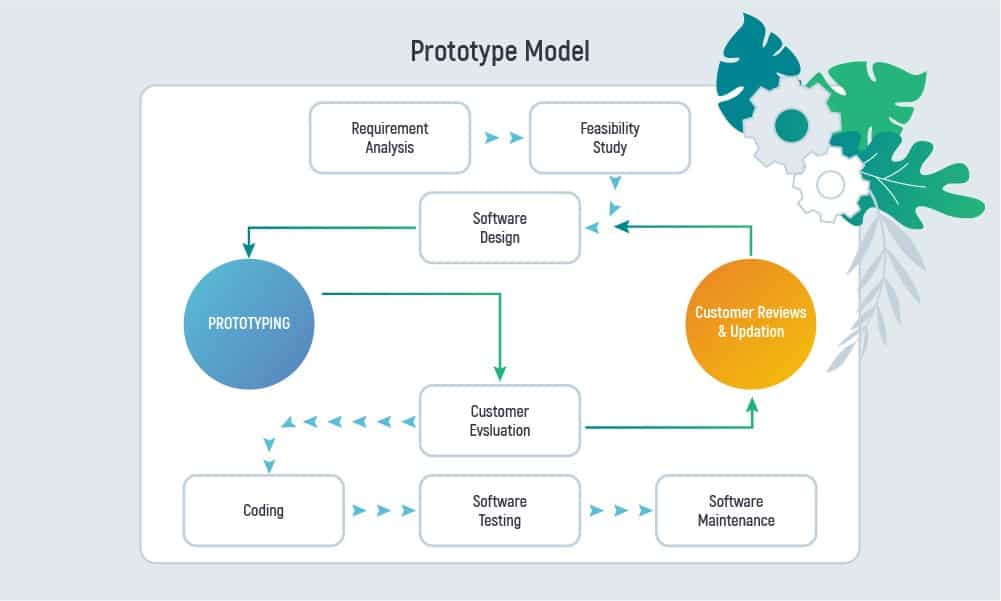
V-shaped Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjcyOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjcyOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 5. Prototype Model

Prototyping is the creation of software application prototypes that show the visual representation of the minimal features of the incomplete product. This enables understanding customer requirements early in the development stage and getting valuable feedback. The model includes:

* Identification of basic requirements
* Initial prototype development
* Prototype review
* Prototype revision and enhancement

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=Njc5OjE0Mw==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=Njc5OjE0Mw==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

There are different types of software prototypes. These are:

* Rapid Prototyping – on understanding requirements the system is built with a clearer understanding of user needs
* Evolutionary Prototyping – development of functional prototypes with limited functionality based on the understood requirements and adding the other requirements in the process
* Incremental Prototyping – building numerous functional prototypes that form a system and putting them together
* Extreme Prototyping – is a web development domain with three phases: basic prototype in the HTML; prototype service layer; implementation and integration in the final prototype

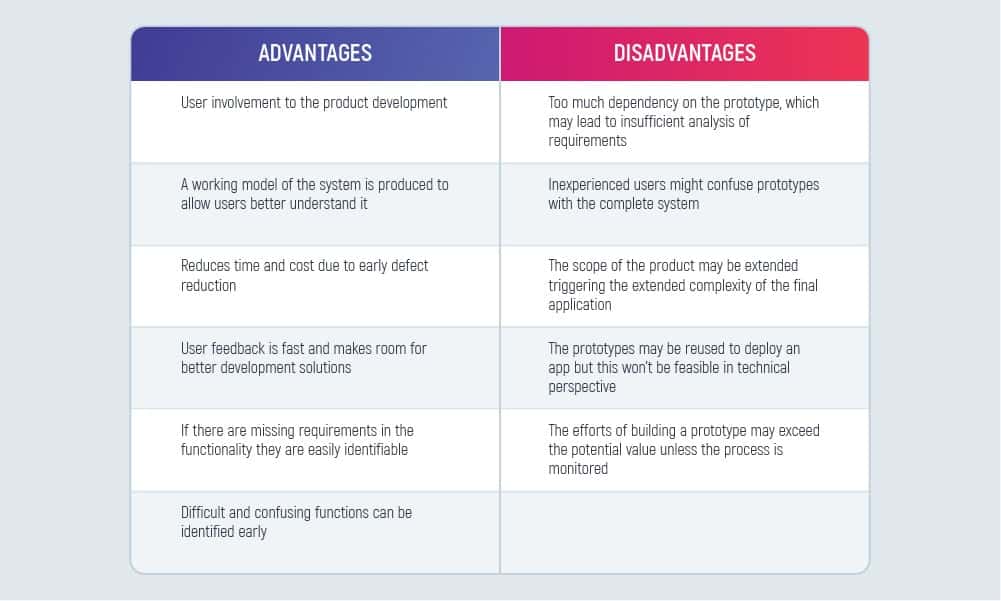
#### Where and when is the Prototyping Model applicable?

Prototyping is applicable to online software system development where there is a high level of user interaction. For example,

* Systems that require users to fill out forms
* Software that requires lots of data processing

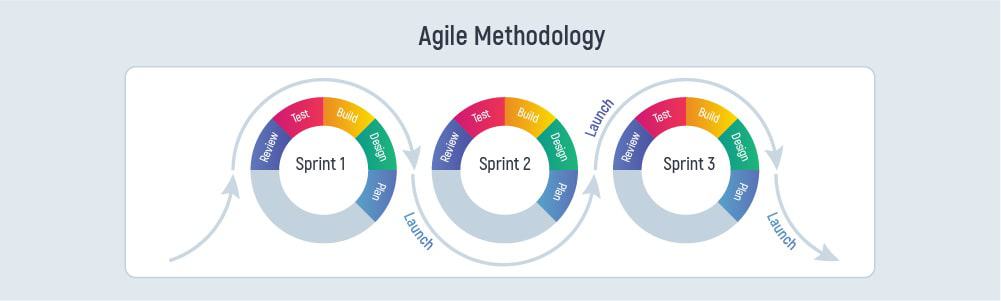
The Prototyping Model also has its advantages and disadvantages. Of course, they depend on the organization’s needs and are considered before choosing this model.

Prototyping Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NjkxOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NjkxOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 6. Agile Model

Agile methodology is based on the iterative and incremental model but it is more adaptive and delivers working software at a speedy pace. Due to meeting the client’s requirements and being very flexible, the Agile methodology brings customer satisfaction. Agile projects tend to break the production process down into incremental builds or we call them iterations. Every iteration has a timeline of 1 to 3 weeks and includes different teams working on various project areas simultaneously. This allows a decrease in the time of production and it may lower production costs as well.

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=Njk0OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSAzMDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjMwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=Njk0OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSAzMDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjMwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

Every iteration consists of the same processes, which are:

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing
* Acceptance Testing

At the end of every iteration, you get a working product, which is presented to the customer to understand whether the application is built according to the customer’s requirements and receive feedback on the work being done.

Agile methodology is divided into several methods that have their own principles. For instance,

* Rational Unified Process (RUP)
* Scrum
* Kanban
* ScrumBan
* Extreme Programming (XP)
* ScrumXP
* Iterative Development
* Lean Software Development

#### When can we use the Agile Methodology?

* In small to medium-sized software projects
* To produce the required multiple variants of the product
* When you need to break down the development process into smaller parts – iterations
* When there’s a need for software delivery acceleration
* When the team needs to be flexible in changing priorities
* When the need is to enhance software quality, productivity, and predictability in delivery
* To promote project risk reduction and achieve cost reduction possibilities
* To manage the distributed teams effectively on the project

To learn more about Agile Methodology, get to read our [Agile Software Development Life cycle Phases](https://inoxoft.com/blog/agile-software-development-lifecycle-phases/) article.

Like any other methodology, Agile has its positive and negative aspects. For example,

Agile Model Advantages & Disadvantages

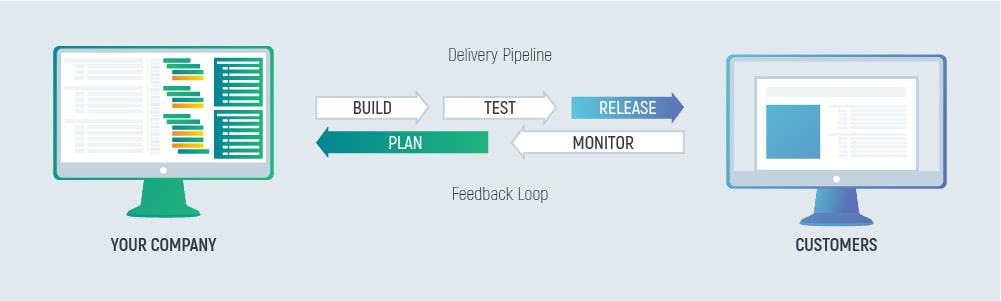
[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NzI0OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NzI0OjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

### 7. The DevOps Model

The DevOps model has its unique principles that promote automation and team collaboration. Therefore, the primary focus of the model is the enhancement of collaboration between teams that are driven by sequential feedback. The DevOps model resembles the Agile methodology, but differs due to the collaboration focus between technical team members and managers, while Agile focuses on client involvement.

What does [DevOps](https://aws.amazon.com/devops/what-is-devops/) carry out?

* Continuous Integration and Delivery
* Microservices
* Infrastructure as Code
* Monitoring and Logging
* Communication and Collaboration

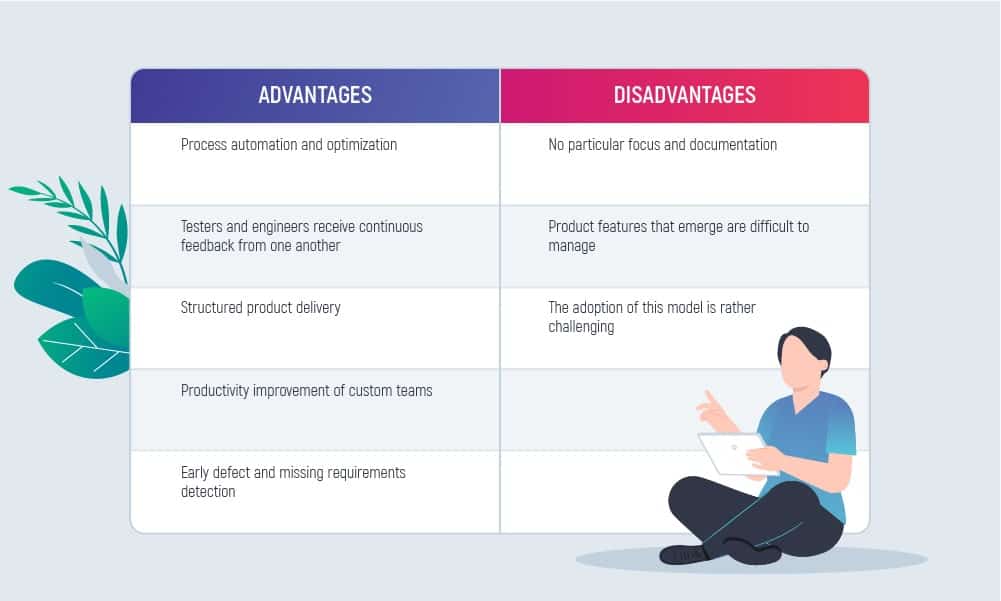
[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NzMzOjE0Mw==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSAzMDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjMwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NzMzOjE0Mw==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSAzMDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjMwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)

#### When can we apply the DevOps Model?

* In complex projects
* Projects that require QA and testing
* Projects that include large teams from multiple departments

Being quite popular, the DevOps Model also has some pros and cons. It depends on what projects you are planning to adopt this model and whether you are prepared for the quite hard adoption curve. Let’s see what the model has to offer with regard to productivity from both sides.

DevOps Model Advantages & Disadvantages

[<="" a="" style="box-sizing: inherit; background-color: transparent; color: rgb(26, 92, 141); text-decoration: underline; touch-action: manipulation; outline: 0px;">[](data:image/svg+xml;nitro-empty-id=NzQwOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)](data:image/svg+xml;nitro-empty-id=NzQwOjEzOA==-1;base64,PHN2ZyB2aWV3Qm94PSIwIDAgMTAwMSA2MDEiIHdpZHRoPSIxMDAxIiBoZWlnaHQ9IjYwMSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIj48L3N2Zz4=)