There are various software development approaches defined (e.g. [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/), [incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/), [iterative model](http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/)). Each process model follows a particular life cycle in order to ensure success in process of software development.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. In the development phase the developers test their code which is called Unit testing. After coding and development the testing team verifies the deliverable of the implementation phase against requirements. Two types of testing happen one we call CIT (Continuous Integration Testing) and the other one is SIT (System Integration Testing). Once the software is checked and validated at Organization level the same software would get tested by the Client’s organization by their users which is called UAT (User Acceptance Testing). Post UAT the software gets deployed in Client’s environment officially.

There are following six phases in every software development life cycle model:

1. Requirement gathering and analysis
2. Design
3. Development or coding
4. Testing (CIT, SIT and UAT)
5. Deployment
6. Application Maintenance

**1) Requirement gathering and analysis:**  Business requirements are gathered in this     phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the new system.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

**2)  Design:**  In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

**3)  Development / Coding:**  On receiving system design documents, the work is divided in modules/units and actual coding starts. Since the code is produced in this phase so it is the main focus for the developer. This is the longest phase of the software development life cycle.

**4)**[**Testing**](http://istqbexamcertification.com/what-is-a-software-testing/)**:**  After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

**5)  Deployment:** After successful testing the product is delivered / deployed to the customer for their use.

**6) Maintenance:** Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

**Different types of software development models or methodologies:**

There are many development life cycle models that have been developed in order to achieve different required objectives.

The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use.

There are various software development models or methodologies. Most common are as follows:

1. [Waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/)
2. [RAD model](http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/)
3. [Agile model](http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/)
4. RUP model
5. [Iterative model](http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/)

Choosing right model for developing of the software product or application is very important. Based on the model the development and testing processes are carried out.

Based on the software application or product, they select the type of development model whichever suits to the application. These days in market the ‘Agile Methodology’ is the most widely used model. ‘Waterfall Model’ is the very old model. In ‘Waterfall Model’ testing starts only after the development is completed. Because of which there are many defects and failures which are reported at the end. So, cost of fixing these issues is high. Hence, people prefer ‘Agile Model’. In ‘Agile Model’ after every sprint there is a demo-able feature to the customer. Hence customer can see the features whether they are satisfying their need or not.

1. **Waterfall Model:**

The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**.  It is very simple to understand and use.  In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In **waterfall model phases** do not overlap.

**Diagram of Waterfall-model:**

[](http://istqbexamcertification.com/wp-content/uploads/2012/01/Waterfall-model.jpg)

**Advantages of waterfall model:**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are very well understood.

**Disadvantages of waterfall model:**

* Once an application is in the [testing](http://istqbexamcertification.com/what-is-a-software-testing/) stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

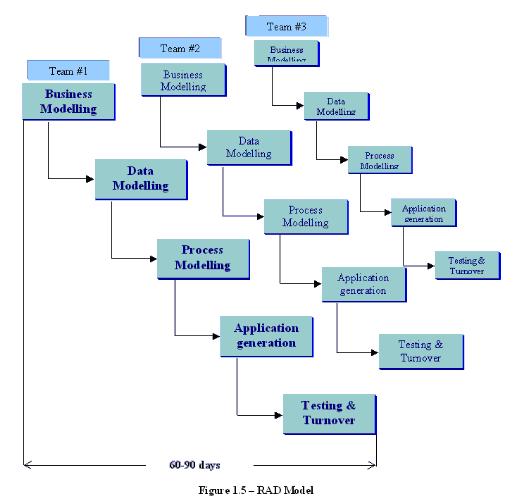
**When to use the waterfall model:**This model is used only when the requirements are very well known, clear and fixed.

* Product definition is stable.
* Technology is understood.
* There are no ambiguous requirements
* Ample resources with required expertise are available
* The project is short.

Very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demoed to the end users. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the log**ic.**

1. **RAD MODEL:**

RAD model is Rapid Application Development model.It is a type of [incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

**Diagram of RAD-Model:**[](http://istqbexamcertification.com/wp-content/uploads/2012/01/RAD-model.jpg)

**The phases in the rapid application development (RAD) model are:**

1. **Business modeling:** The information flow is identified between various business functions.
2. **Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.
3. **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.
4. **Application generation:** Automated tools are used to convert process models into code and the actual system.
5. **Testing and turnover:** Test new components and all the interfaces.

**Advantages of the RAD model:**

* Reduced development time.
* Increases reusability of components
* Quick initial reviews occur
* Encourages customer feedback
* Integration from very beginning solves a lot of [integration issues](http://istqbexamcertification.com/what-is-system-integration-testing/).

**Disadvantages of RAD model:**

* Depends on strong team and individual performances for identifying business requirements.
* Only system that can be modularized can be built using RAD
* Requires highly skilled developers/designers.
* High dependency on modeling skills
* Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

**When to use RAD model:**

* RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
* It should be used if there’s high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
* RAD [SDLC model](http://istqbexamcertification.com/what-are-the-software-development-models/) should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

1. **Agile Model:**

Agile development model is also a type of [Incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly [tested](http://istqbexamcertification.com/why-is-testing-necessary/) to ensure [software quality](http://istqbexamcertification.com/what-is-software-quality/) is maintained. It is used for time critical applications.  Extreme Programming (XP) is currently one of the most well known agile [development life cycle model](http://istqbexamcertification.com/what-are-the-software-development-models/).

**Diagram of Agile model:**



**Advantages of Agile model:**

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools.
* Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Daily coordination between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed

**Disadvantages of Agile model:**

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

**When to use Agile model:**

* When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
* To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
* Unlike the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/) in agile model very limited [planning](http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/) is required to get started with the project. Agile assumes that the end users’ needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be removed based on feedback. This effectively gives the customer the finished system they want or need.
* Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

1. **Rational Unified Process**

The **Rational Unified Process** (**RUP**) is an [iterative](http://en.wikipedia.org/wiki/Iterative_and_incremental_development) [software development process](http://en.wikipedia.org/wiki/Software_development_process) framework created by the [Rational Software](http://en.wikipedia.org/wiki/Rational_Software) Corporation, a division of [IBM](http://en.wikipedia.org/wiki/IBM).

RUP is not a single concrete prescriptive process, but rather an adaptable process [framework](http://en.wikipedia.org/wiki/Software_framework), intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs. RUP is a specific implementation of the [unified process](http://en.wikipedia.org/wiki/Unified_process).

**RUP phases and disciplines**

The RUP has determined a project life-cycle consisting of four phases. These phases allow the process to be presented at a high level in a similar way to how a 'waterfall'-styled project might be presented, although in essence the key to the process lies in the iterations of development that lie within all of the phases. Also, each phase has one key objective and milestone at the end that denotes the objective being accomplished.

#### Inception phase

The primary objective is to scope the system adequately as a basis for validating initial costing and budgets. In this phase the business case which includes business context, success factors (expected revenue, market recognition, etc.), and financial forecast is established. To complement the business case, a basic use case model, project plan, initial risk assessment and project description (the core project requirements, constraints and key features) are generated. After these are completed, the project is checked against the following criteria:

* [Stakeholder](http://en.wikipedia.org/wiki/Stakeholder_(corporate)#In_management) concurrence on scope definition and cost/schedule estimates.
* Requirements understanding as evidenced by the fidelity of the primary use cases.
* Credibility of the cost/schedule estimates, priorities, risks, and development process.
* Depth and breadth of any architectural prototype that was developed.
* Establishing a baseline by which to compare actual expenditures versus planned expenditures.

If the project does not pass this milestone, called the lifecycle objective milestone, it either can be cancelled or repeated after being redesigned to better meet the criteria.

#### Elaboration phase

The primary objective is to mitigate the key risk items identified by analysis up to the end of this phase. The elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form.

The outcome of the elaboration phase is:

* A use-case model in which the use-cases and the actors have been identified and most of the use-case descriptions are developed. The use-case model should be 80% complete.
* A description of the software architecture in a software system development process.
* An [executable architecture](http://en.wikipedia.org/wiki/Executable_Architecture) that realizes architecturally significant use cases.
* Business case and risk list which are revised.
* A development plan for the overall project.
* Prototypes that demonstrably mitigate each identified technical risk.
* A preliminary user manual (optional)

This phase must pass the lifecycle architecture milestone criteria answering the following questions:

* Is the vision of the product stable?
* Is the architecture stable?
* Does the executable demonstration indicate that major risk elements are addressed and resolved?
* Is the construction phase plan sufficiently detailed and accurate?
* Do all stakeholders agree that the current vision can be achieved using current plan in the context of the current architecture?
* Is the actual vs. planned resource expenditure acceptable?

If the project cannot pass this milestone, there is still time for it to be cancelled or redesigned. However, after leaving this phase, the project transitions into a high-risk operation where changes are much more difficult and detrimental when made.

The key domain analysis for the elaboration is the system architecture.

#### Construction phase

The primary objective is to build the software system. In this phase, the main focus is on the development of components and other features of the system. This is the phase when the bulk of the coding takes place. In larger projects, several construction iterations may be developed in an effort to divide the use cases into manageable segments that produce demonstrable prototypes.

This phase produces the first external release of the software. Its conclusion is marked by the initial operational capability milestone.

#### Transition phase

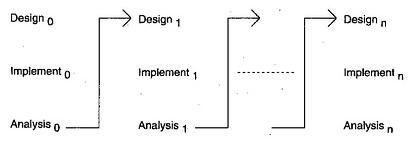
The primary objective is to 'transit' the system from development into production, making it available to and understood by the end user. The activities of this phase include training the end users and maintainers and beta testing the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase.

If all objectives are met, the product release milestone is reached and the development cycle is finished.

1. **Iterative**[**life cycle model**](http://istqbexamcertification.com/what-are-the-software-development-models/)**:**

An iterative [life cycle model](http://istqbexamcertification.com/what-are-the-software-development-models/) does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

**Diagram of Iterative model:**

[](http://istqbexamcertification.com/wp-content/uploads/2012/01/Iterative-model.jpg)

**Advantages of Iterative model:**

* In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
* In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
* In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
* In iterative model less time is spent on documenting and more time is given for designing.

**Disadvantages of Iterative model:**

* Each phase of an iteration is rigid with no overlaps
* Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

**When to use iterative model:**

* Requirements of the complete system are clearly defined and understood.
* When the project is big.
* Major requirements must be defined; however, some details can evolve with time.