Models	Summary	Stages	Pros	Cons
Prototype	The Prototype Model is an iterative software development model that involves creating a working model of the software early in the development cycle. This model is applied when detailed information related to input and output requirement of the system is not available. Here, customer provides continuous feedback on the prototype, which is then incorporated into subsequent iterations until a software product that meets the customer's needs is delivered.	Requirement gathering and analysis, Build prototype, User evaluation and feedback, Refining prototype, Engineer product	⇨ The customer is involved throughout the development process, providing feedback on the prototype and ensuring that the final product meets their needs. ⇨ The Prototype Model is flexible and can accommodate changes requested by the customer, which can be incorporated into subsequent iterations of the prototype. ⇨ Helps in reducing risks associated with the software	⇨ If the user is not satisfied by the developed prototype, then refining prototype process goes on until a perfect prototype is developed which is time-consuming and expensive.
RAD Model (Rapid Application Development)	RAD Model or Rapid Application Development model is a software development process based on prototyping without any specific planning. In RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.	Requirement gathering, Data Modeling, Process Modeling, Application Modeling, Testing and Turnover	⇨ Requirements can be changed at any time ⇨ Encourages and priorities customer feedback ⇨ Reviews are quick	⇨ Cannot work with large teams ⇨ Need highly skilled developers ⇨ High team collaboration is needed and may result in lower-quality end product if not executed properly.

Iterative Model	The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.	Requirement analysis and specification, Design, Implementation and unit testing, Integration and System Testing, Operation and Maintenance	⇨ Divides project into smaller parts ⇨ Creates working model early and provides valuable feedback ⇨ Feedback from one phase provides design information for the next phase ⇨ Very useful when more staffing is unavailable	⇨ User community needs to be actively involved in the project. This demands on time of the staff and add project delay ⇨ Communication and coordination skills take a center stage ⇨ Informal requests for improvement for each phase may lead to confusion
Incremental Model	A iterative model where development is divided into smaller, more manageable increments, with each increment building on the previous increment.	requirements, design and development, testing, and implementation	⇨ Easier to test and debug ⇨ It is used when there is a need to get a product to the market early	⇨ Requires careful planning and coordination to ensure each increment is cohesive and compatible with the previous increment.
Evolutionary Model	An iterative and incremental model that involves rapid prototyping and continuous refinement of software based on feedback from stakeholders.	communication, planning, modelling, implementation, and testing.	⇨ Allows for frequent feedback and refinement, accommodates changes, and promotes customer involvement.	⇨ Requires significant resources for rapid prototyping, can be difficult to manage and control without proper planning and coordination.
Big Bang Model	A simple, undisciplined approach where all components of a system are developed and integrated simultaneously.		⇨ Quick and easy to implement, suitable for small, simple projects.	⇨ High risk of failure due to lack of planning and testing, difficult to manage and debug.

Waterfall Model	The Waterfall model is an example of sequential model. It is the pioneer of the SDLC processes. It was the first model that was widely used in the software industry. The development of one phase starts only when the previous phase is complete. There is no overlapping in the waterfall model in terms of phase. Since the phases fall from a higher level to lower level, like a waterfall, it's named as the waterfall model.	Requirement Analysis, System Design, Implementation, System Testing, System Development, System Maintenance,	⇨ Simple and easy to understand and use. ⇨ Since the phases are rigid and precise, one phase is done at a time, it is easy to maintain ⇨ Works well and yield the appropriate results. ⇨ The entry and exit criteria are well defined, so it is easy and systematic to proceed with quality. ⇨ Results are well documented.	⇨ Cannot Adopt the changes in requirements ⇨ It becomes very difficult to move back to the phase ⇨ Delivery of the final product is late as there is no prototype which is demonstrated intermediately ⇨ For bigger and complex projects, this model is not goo as a risk factor is higher. ⇨ Doesn't work for long and ongoing projects.
Agile Model	A flexible, iterative approach where requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams.	Planning, Analyzing, Architectural design, Coding, Unit testing, Delivery, Feedback	⇨ Supports customer involvement and customer satisfaction ⇨ Strong communication of the software team with the customer. ⇨ Focus on user and customer ⇨ Rapid development ⇨ Allows changes easily ⇨ Cost-saving, faster delivery ⇨ Promotes team works	⇨ Can be challenging for teams new to agile methodologies, requires ongoing communication and coordination between team members.

Spiral Model	Spiral model is a combination of both, iterative model and one of the SDLC model. This model considers risk, which often goes unnoticed by most other models. The model starts with determining objectives and constraints of the software followed by prototyping the software phase. This includes risk analysis. Then one standard SDLC model is used to build software. This fourth phase is plan of next iteration.	Planning, Risk Analysis, Engineering, Evaluation	⇨ This model is used for large projects which involve risk and cost on every changes. ⇨ The spiral model enables gradual releases and refinement of a product through each phase of the spiral as well as the ability to build prototypes at each phase.	⇨ Doesn't work well for smaller projects ⇨ Spiral model demands risk assessment expertise
V Model	The V shaped model is an extension of the waterfall model. The V-shaped model shows the relationships between each phase of development and the associated phase of testing. It's also referred to as the 'verification and validation model'. This is because each verification phase is associated with a validation phase. The main aspect of any software is to see how it performs. It needs to be tested many times. Therefore, the main focus of V shaped model is in testing	Requirement Analysis, System Requirements, Implementation Phase / Coding, Component Test Execution/Unit Testing: Integration Test Execution/Integration Testing: System Test Execution/ System Testing: Acceptance Test Execution/User Acceptance Testing:	⇨ This is a highly-disciplined model and Phases are completed one at a time. ⇨ Works well for smaller projects where requirements are very well understood. ⇨ Simple and easy to understand and use. ⇨ Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.	⇨ High 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. ⇨ Once an application is in the testing stage, it is difficult to go back and change a functionality.