Case study of software development life cycle model

Model	Approach	Stage	When it uses	Advantages	Disadvantages
Waterfall model	It is a Sequential In order	1.Requirements 2.Analysis 3.Design 4.Coding/Implementation 5.Testing 6.Operation/Deployment 7.Maintenance	When requirements are very well known clearly & fixed	Suitable for large system and teams	Longer length in each iteration or increment
Iterative model	To become the final product or software	1.Requirements 2.Design & Development 3.Testing 4.Implementation	When requirements are defined clearly & easy to understand	Appropriate for Stable development environment	Lack of user involvement throughout the life cycle of the product
Spiral model	Sequential development process I pre-defined phases	1.Identification 2.Design 3.Construct/Build 4.Evaluation & Risk Analysis	It favored for large, expensive & complicated projects	Flexible umber changes are allowed I spiral model	It is not suitable for the small & low-risk product because its cloud be costly for a smaller project
V – model	It executes of processes happens in a sequential manner in a V-shape	1.Requirements & Analysis 2.System Design 3.Archeitectural Design 4.Module Design 5.Coding phrase 6.Unit testing 7.Integration testing 8.system testing 9.Acceptance testing	Small projects where project requirements are clear	It provides a proactive error tracking feature for developer	Software is developed during the phrase of implementation, so no initial prototype of the software are produced
RAD Model (Rapid Application Model)	Based on prototyping & iterative development with no specific planning involved	1.Business Modelling 2.Data Modelling 3.Process Modelling 4.Application Modelling 5.Testing &turning	When a system can be modularized to be delivered in an incremental manner	Requirements can be changed at the time	Need strong team collaboration

Increment al Model	A process of s/w development where the product is design, implemented & tested incrementally	1.Requirements Analysis 2.Design & Development 3.Testing 4.Implementation	When the requirements of the complete system are clearly defined and understood	The model is less costly compared to others	Rectifying a problem in one unit requires correction in all the unit & customer a lot of time
Agile model	Collaborative decision making between requirement & solution teams, & a cyclical iterative progression of producing working software	1.Requirements Analysis 2.Design 3.Testing 4.Planning 5.coding	When the product vision or features are not well defined	Customer satisfaction by rapid continuous delivery of useful software	There is lack of emphasis on necessary designing & documentation
Big-Bang model	SDLC model where we do not follow and specific process	Very little or no planning	Small project with one or two developers working together & is also useful for academic or practice project	Very flexibility to developers	Very High risk & uncertainty