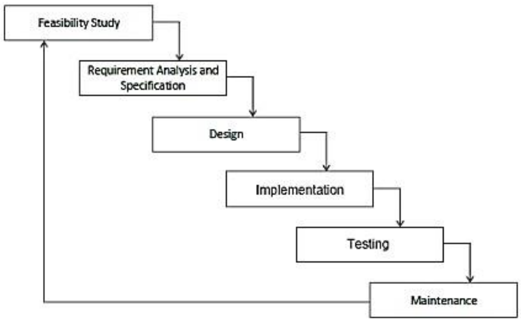
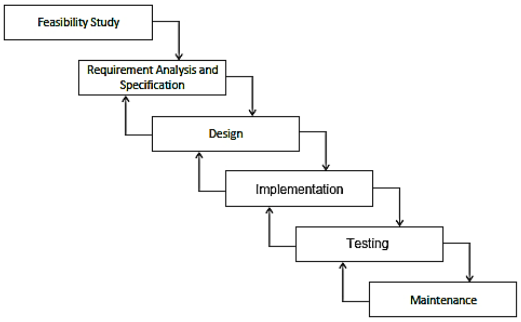
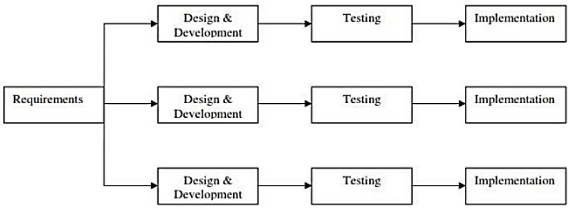
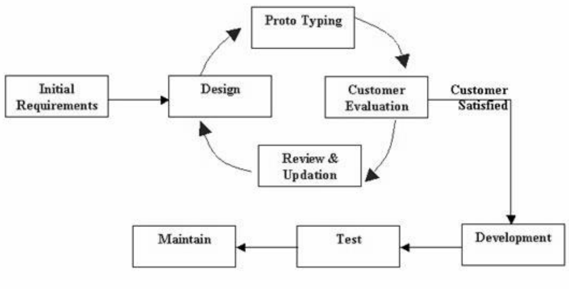
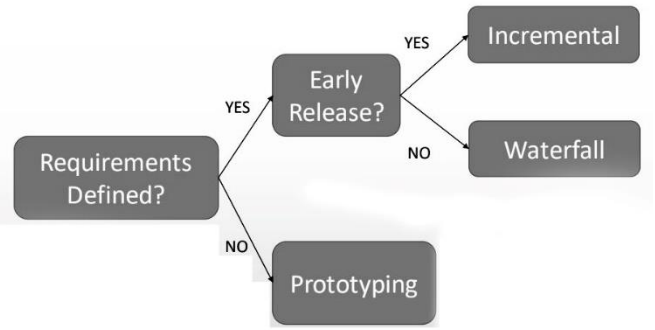
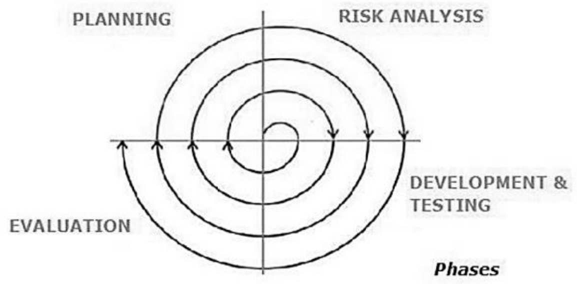
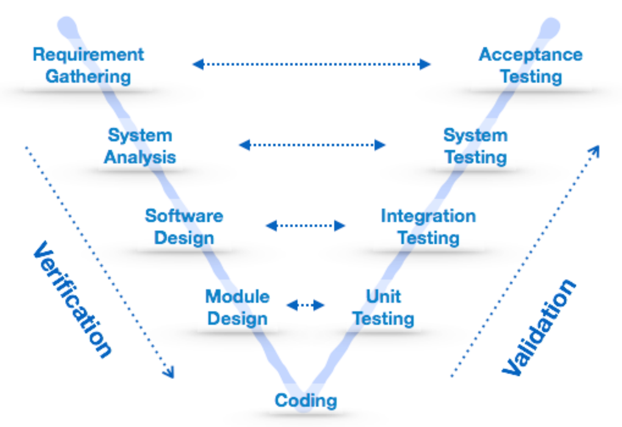
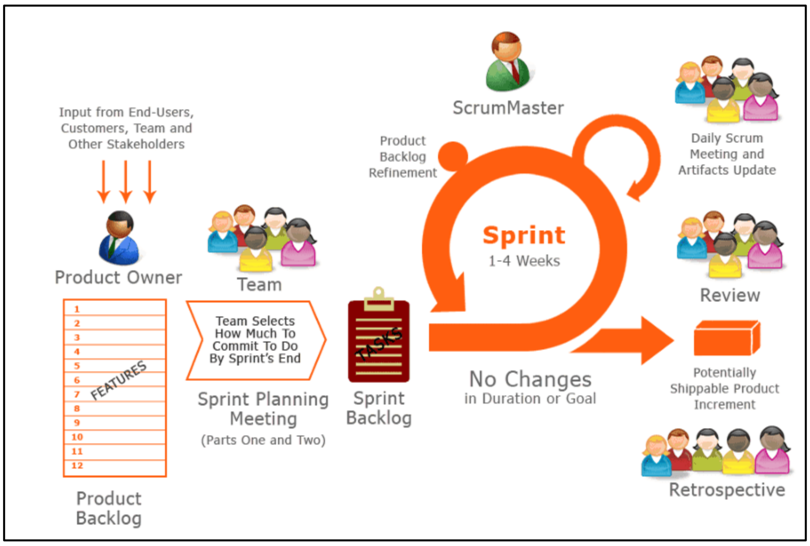
SDLC Model	Strengths	Weaknesses	When to use
 Feasibility Study Requirement Analysis and Specification Design Implementation Testing Maintenance	• Simple and easy to manage– each phase has specific deliverables. • Milestones are better understood • Sets requirements stability **• Works well for smaller projects where requirements are very well understood.** • A schedule can be set with deadlines.	• No working software is produced until end. • High uncertainty. • Delays discovery of serious errors. • After requirements phase, there is no formal way to make changes to the requirements. **• Not a good model for,** ▪ ***complex projects*** ▪ ***projects where requirements are at a moderate to high risk of changing***	• Software requirements clearly defined and known • Product definition is stable • New version of the existing software system is created • Software development technologies and tools are well known • Ample resources with required expertise are Available.
 Feasibility Study Requirement Analysis and Specification Design Implementation Testing Maintenance	• Defects are detected and fixed early through the feedback path.	**• Limited customer interactions.** **• Difficult to incorporate change requests.**	*Same as the classical waterfall model*
 Requirements Design & Development Testing Implementation Design & Development Testing Implementation Design & Development Testing Implementation	**• Generates working software quickly and early during the software life cycle.** • More flexible - less costly to change scope and requirements. • Easier to test and debug. • Easier to manage risk. • Lowers initial delivery cost. • Less stress for the development team.	• Requires good planning and design • Each phase of an iteration is rigid and do not overlap each other. • Demarcation of increments can be difficult in a practical application. • Total cost of the system might not be lower. • Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.	•On projects which have lengthy development schedules **• A need to get basic functionality to the market early.** • Most of the requirements are known upfront but are expected to evolve over time **• On a project with new technology.**
 Proto Typing Initial Requirements Design Customer Evaluation Customer Satisfied Review & Updation Maintain Test Development Proto Type Model	•Ability to clarify user's expectations for the system to be developed •Prototype stimulates awareness of additional needed functionality •Better user satisfaction •Early user feedback	**• Scope Creep - The system scope may expand beyond original plans.** •Overall maintainability may be overlooked. • The customer may want the prototype be delivered. **• Process may continue forever.**	**• Requirements are unstable or have to be clarified** • Many user interfaces **•New technology** **•New, original development** • Developers are not familiar with the technical and development tools

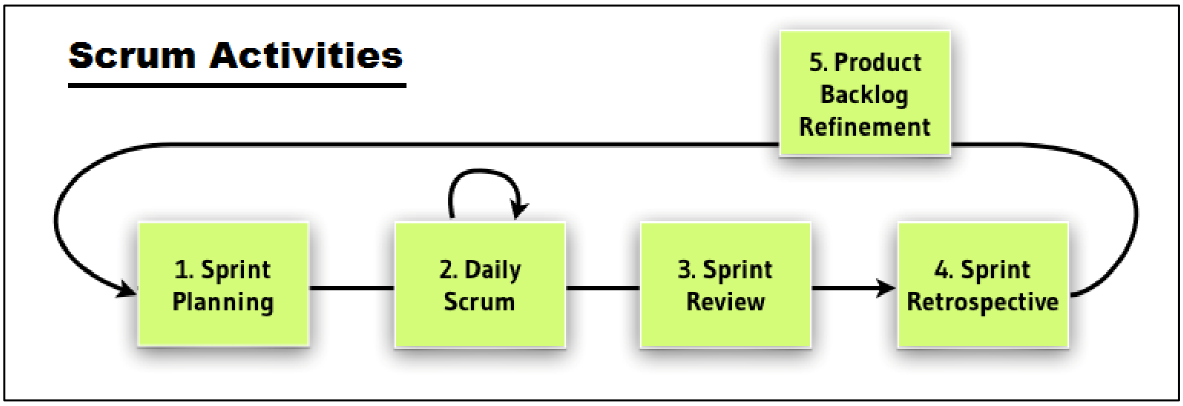


 PLANNING RISK ANALYSIS DEVELOPMENT & TESTING EVALUATION Phases	• Focus on risk analysis. • Good for large and **mission critical projects (Flight control, weapon direction, medical equipment)** • A working software is produced early • The design does not have to be perfect • Early and frequent feedback from users • Cumulative costs assessed frequently	•Can be a costly model to use. •Risk analysis requires expertise. **•Success is highly dependent on the risk analysis phase.** **•Doesn't work well for smaller projects.**	•For medium to high-risk projects •New technology to be used **•Complex, constantly changing and continuous requirements** •Significant changes are expected (research and exploration) •Users are unsure of their needs



Waterfall model	V-model
The cost of Waterfall model is low.	V-model is expensive.
Re-usability of Waterfall model is Limited.	V-model can be Re-use for some extent.
In Waterfall model testing activities start after the development activities are over.	In V-model testing activities start with the first stage.
Guarantee of success through Waterfall model is low.	Guarantee of success through V-model is high.
Waterfall model is a continuous process.	V-model is a simultaneous process.


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
Product Backlog Refinement
ScrumMaster
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective


Scrum Activities
5. Product Backlog Refinement
1. Sprint Planning
2. Daily Scrum
3. Sprint Review
4. Sprint Retrospective