Adv & Disadvantages.

(follow PPT)

Comparison of All SDLC Models.

Classical Waterfall	Iterative	Prototype	Incremental	Evaluationary	RAD	Sprial	Agile
Basic, Rigid, inflexible, Not for Real project	Basic, Problem is well understood	User Requirement is not clear, No early look on Requirement, costly, High user involvment, Reusability.	model by model delivery, Easy to test, Easy to debug. Requirement is local	Large Project, Requirement is not lock	Time & cost constant, User interaction at all level, Reusability	Risk, not for small Project, No Early look on Requirements, less Experienced can work	Flexible, Advantage, Parallel, Processes divided into Sprints

Software Requirements

*) It is the description of Features & functionalities of the target system.

*) It is the description of what the system do

*) Requirement Engineering (RE) refers to the process of defining, documenting & maintaining requirements in the engineering design process.

*) It is a Four step process, which includes
- Feasibility study
- Requirement gathering / Elicitation
- Software Requirement Specification
- " " Validation

Tool Support for RE

*) observation Report (User observation)
*) Questionnaires (interviews, surveys ...)
*) Use cases
*) User stories
*) Requirement workshops
*) Mind Mapping
*) Role-playing
*) Prototyping

Functional / Non functional Requirements

*) Requirements, which are Related to functional working aspect of software
*) Non-functional Requirements are expected characteristics of target software (Security, storage, configuration, performance, cost, Interoperability, Flexibility, Disaster Recovery, Accessibility)

Software Requirement Specification (SRS)

*) SRS is a description of a software system to be developed.

*) It lays out functional & Non functional Requirements of the s/w to be developed.

*) It may include a set of use cases that described user interactions that the s/w must provide to the user for perfect interaction

SRS ~~Ston~~ Structure.

1) Introduction
   - *) purpose
   - *) Intended Audience
   - *) Scope
   - *) Definitions
   - *) References

11) Overall Description
   - *) User interface
   - *) System "
   - *) constraints, assumptions & dependencies
   - *) User characteristics

III) System features & Requirements

*) Functional Requirements
*) Use case
*) External Interface Requirements
*) Logical database Requirements
*) Non Functional Requirements

IV) Deliver for Approval

iv) Deliver [illegible]

Functional Requirement-

*) It defines a system or its component

*) It specifies "what should the S/w system do"?

*) Functional requirement is specified by user

*) Defined at a component level

*) Helps us verify the functionality of the S/w

*) e.g. Authentication of user whenever he/she logs into the system

Non functional Requirement:

*) It defines a quality attribute of a S/w system

*) It places constraints on "How should the S/w system fulfill the functional requirements",

*) Non functional requirement is specified by technical peoples i.e. Architect, Technical leaders & S/w developers

*) Applied a system as a whole.

*) Helps us to verify the performance of the S/w

*) Emails should be sent with a latency of no greater than 12 hours from such an activity.