Requirement Process

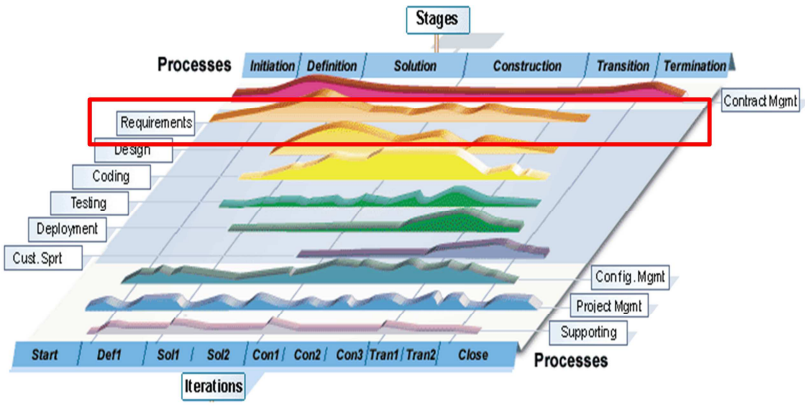
Instructor:

Agenda

- Requirement Process
- Requirement Clarifying
- Common practices, problems

Requirement Process

- First phase of Software engineering



Introduce overview about FSOFT software lifecycle (~RUP)

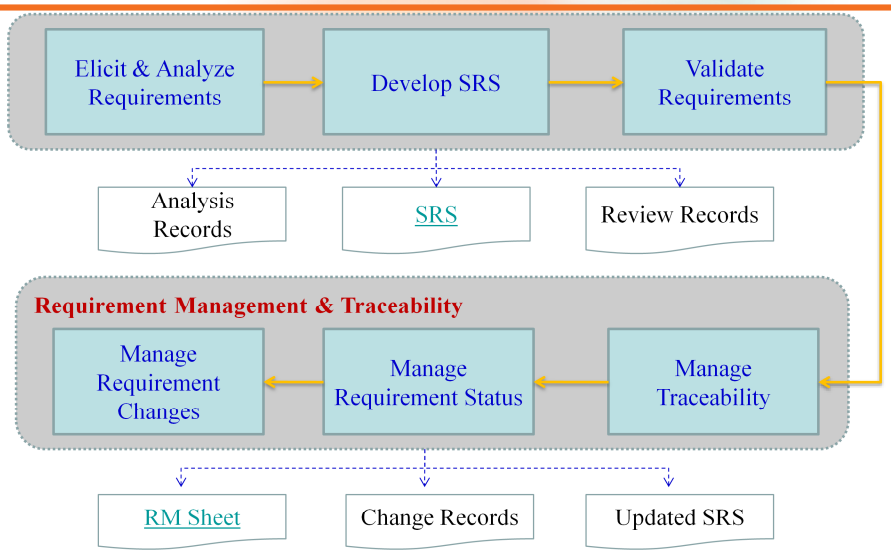
1. 6 phase of project life cycle
2. Introduce process (disciplines) of software development
3. Develop iteratively

    It is best to know all requirements in advance; however, often this is not the case. Several software development processes exist that deal with providing solution on how to minimize cost in terms of development phases.

Requirement Process Objectives

- To ensure that requirements for the software product are defined and understood.
  - Get to know what customer's requirement is
  - Understand the customers' needs & expectation
- To create SRS - Establish and maintain requirements agreement with the requestor and affected groups
- To ensure that the requirements are met.
- Requirements are documented and controlled to establish a basis for software development and project management use.

Requirement Process Workflow

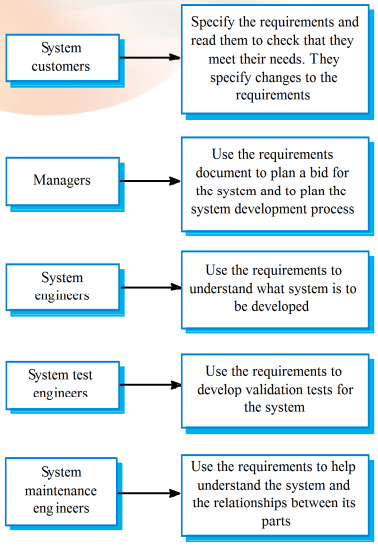


Requirement Process
Elicit & Analyze Requirements

- Sometimes called **requirements discovery**
- Requirements are often not given to you, you have to **elicit** them; must work with **customers** and relevant **stakeholders** to elicit:
  - the **services** that the system should provide
  - the **constraints** that the system should satisfy
- Requirement analysis is done to:
  - Detect and resolve conflicts between requirements
  - Discover the bounds of the software and how it must interact with its environment
  - Elaborate system requirements to derive software requirements

Requirement Process
Develop SRS – Requirement documents 1/3

- Requirements document is the official document of what is required for the system
- Often include only system requirements but sometimes may also include user requirements
- It is NOT a design document. Describe WHAT the system should do rather than HOW it should do



Requirement Process
Develop SRS – Requirement documents 2/3

- URD – User requirement definition
  - Address what users need to do their jobs
  - Composed all business requirements formulated by customer, business rules and other constrains
- SRS – Software requirement specification
  - A set of software requirements as complete, consistent, and correct as possible, from the developer's point of view
  - Document which after base lining, common reference point of the software requirements for customer, developer, tester and PM .

Sample: YourBank_CRM_SRS_v1.0.1.docx

Sample: AQUILA_SRS_v1.0.doc

Trainer show sample of FSOFT document as URD and SRS

URD – User requirement

SRS – System requirement

Requirement Process
Develop SRS – Requirement documents 3/3

- Benefit of good document
  - Basis for agreement between the customers and the team on what the software product is to do.
  - Reduce the development effort.
  - Provide a basis for estimating costs, schedules.
  - Provide a baseline for validation and verification.
  - Facilitate transfer.
  - Serve as a basis for enhancement

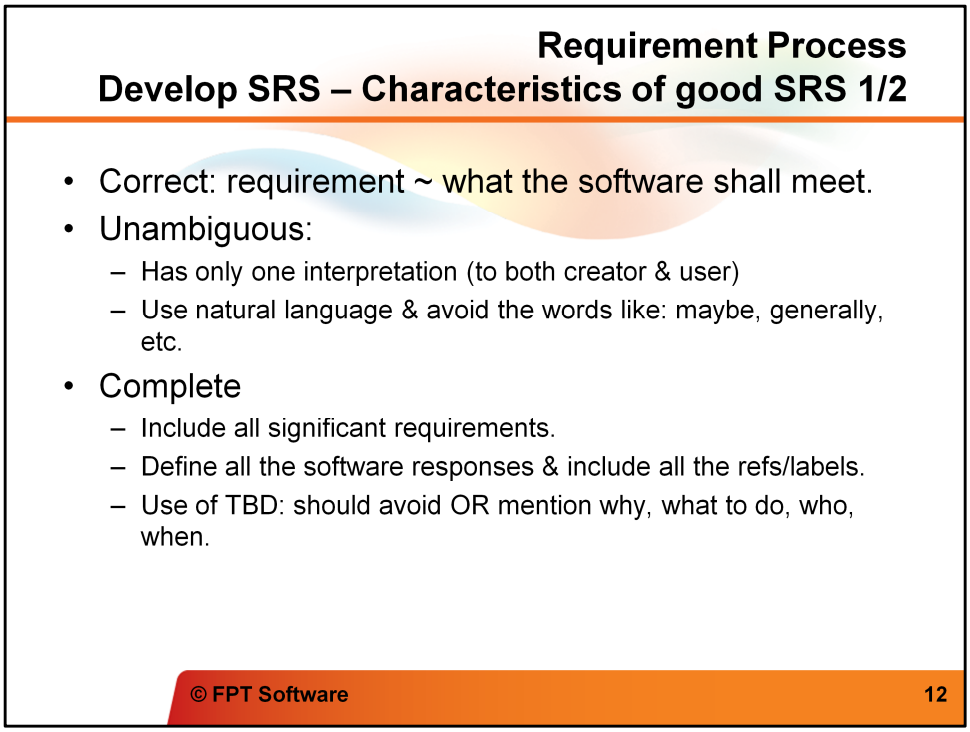
Requirement Process
Develop SRS – Steps & Activities

- Study URD:
  - Analyze user requirement
  - Prepare Q&A list to clarify unclear items with customers
  - Call/interview customers if needed
- SRS:
  - Develop use cases, system requirement
  - Develop functional specification
- Review and approve SRS:
  - Call up meeting for review
  - Keep meeting minutes records

Refer: Checklist_SRS Review.xls

Requirement Process
Develop SRS – Techniques

- Specify requirements using **structured natural language** (forms, tables, etc.)
- **Functional requirements** can be specified using modeling - a combination of graphical notations and structured natural language
  - Use cases, Use case diagrams, Use case specification
  - Activity Diagram, State Diagram
  - DFD, Concept ERD
  - Prototype: Screen Flow, Screen spec scpecification
  - ...
- **Non-functional requirements** can't be modeled => specified using structured natural language only



Correct - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

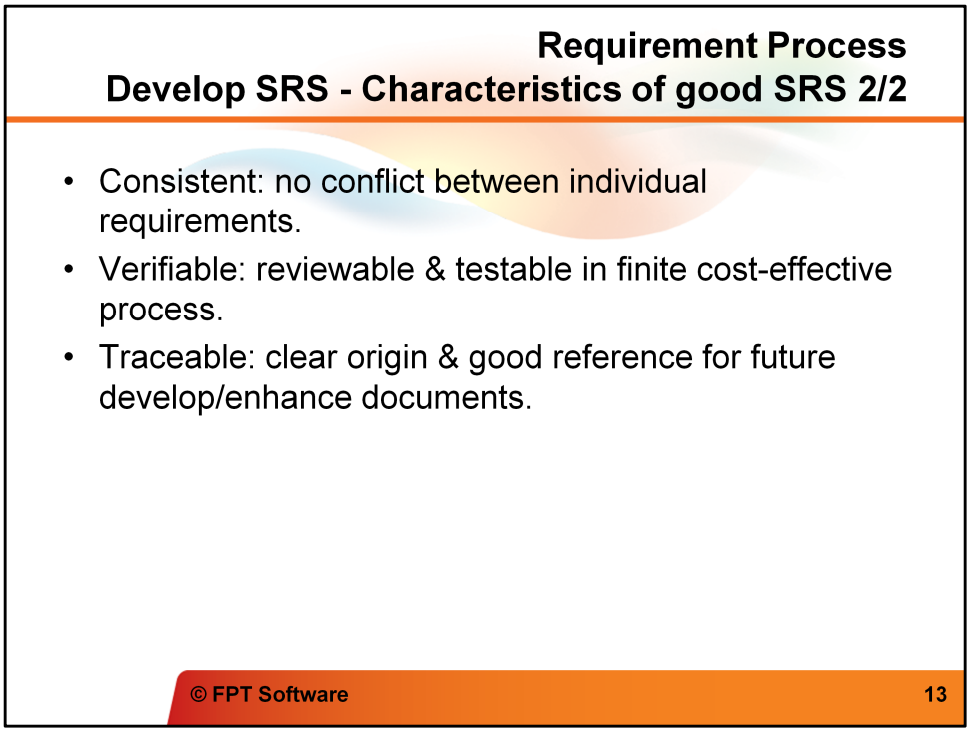
Unambiguous - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

Complete - A simple judge of this is that is should be all that is needed by the software designers to create the software.

Consistent - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

Verifiable - Don't put in requirements like - "It should provide the user a fast response." Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

Traceable - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?



Correct - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

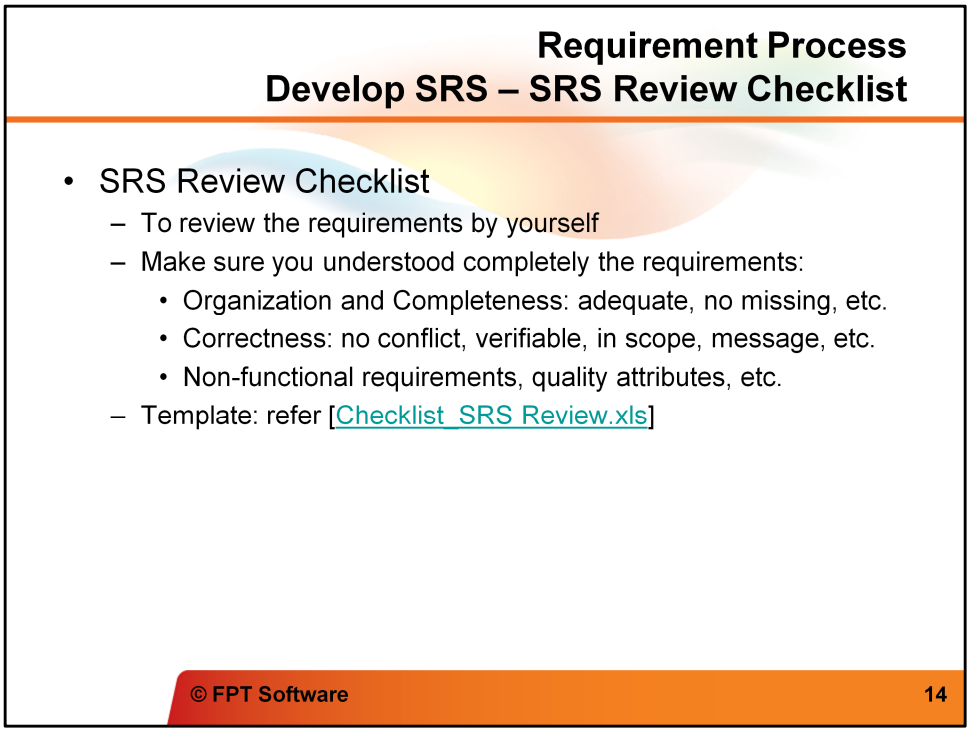
Unambiguous - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

Complete - A simple judge of this is that is should be all that is needed by the software designers to create the software.

Consistent - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

Verifiable - Don't put in requirements like - "It should provide the user a fast response." Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

Traceable - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?



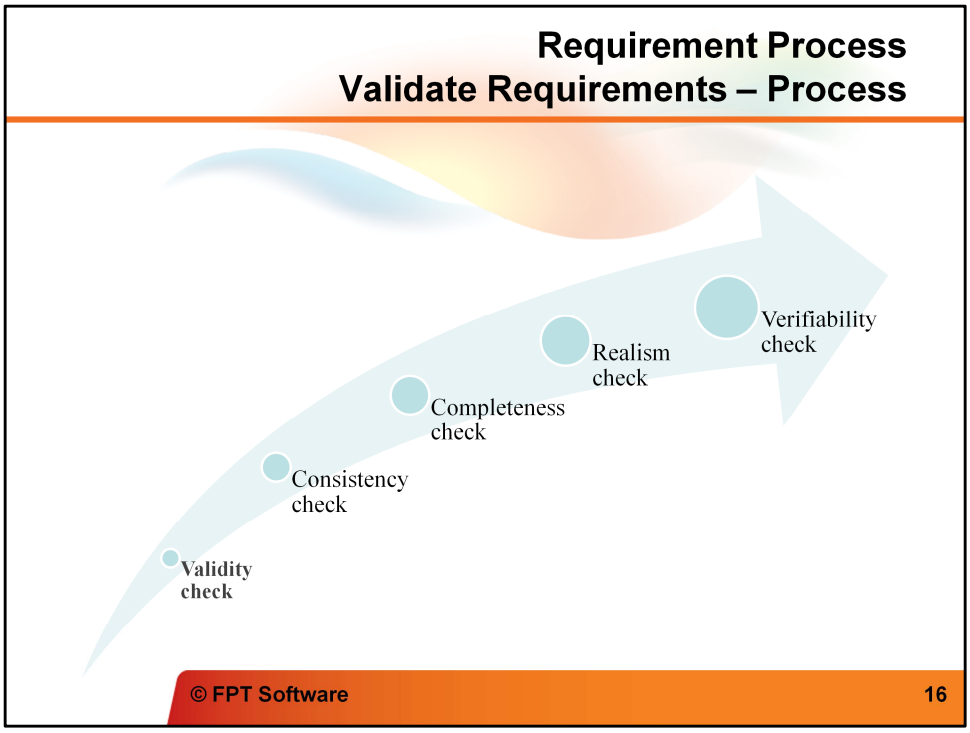
Trainer show sample of FSOFT document as URD and SRS

URD – User requirement

SRS – System requirement

Requirement Process
Validate Requirements – Purpose

- Make sure that the requirements define the system that the customer really wants
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error



1. Validity check
    1. A user may think that a system is needed to perform certain functions. However, further thought and analysis may identify additional or different functions that are required. Systems have diverse stakeholders with distinct needs and any set of requirements is inevitably a compromise across the stakeholder community.
    2. Does the system provide RIGHT functionalities as customer's NEED?

2. Consistency check
    1. Requirements in the document should not conflict. That is, there should be no contradictory constraints or descriptions of the same system function.
    2. Is there any requirement CONFLICT?

3. Completeness check
    1. The requirements document should include requirements which define all functions, and constraints intended by the system user.
    2. Are all functions required by customer INCLUDED?
    3. Is there any common checklists that you can use?

4. Realism check
    1. Using knowledge of existing technology, the requirement should be

Requirement Process
Validate Requirements – Techniques

- Requirements Review
  - Systematic manual analysis of the requirements
  - Involving development staff, customers and relevant stakeholders
- Prototyping
  - Using an executable model of the system to check requirements
- Model Validation
  - Validate the quality of the models developed during analysis
- Test-case generation
  - Developing tests for requirements to check testability

Requirement Process
Requirements management

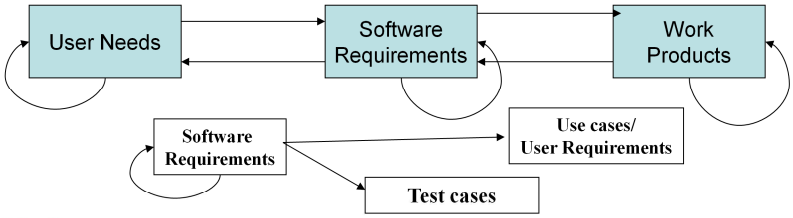
- Manage requirement
  - Requirement Management Sheet, Excel sheet, used to track the status, relationship and change of requirements during the whole project.
  - A mandatory document (dynamic version of SRS)
    - Classify requirement to functional/non-functional requirement
    - To maintain the common reference for all related parties (traceability of requirement and software product)
    - To track the project progress (status of requirement)
    - To track the change (including change request)
    - To collect requirement related metrics for reporting
  - The sheet is created the first time client requirement come

Refer: Template_Requirement Management Sheet.xls

Trainer show sample of RMS here and introduce overview about this document.

Requirement Process
Manage Traceability & Requirement Status

- Why is traceability necessary?
  - The requirements can change at any stage during the product's life.
  - If the requirements are traceable, then when changes happen, it is far easier to find the impacted parts of the product



Batch edit

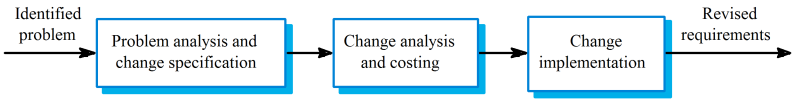
#	Requirement	Deliverable	Type	Size	Requirement section	Design section	Code mo
1	Change on the ICT questionare and report data	5.Final_Code	CR	4	Mails: KBC-RP\Audit\C		frmInstru
2	Export data	4.Rel_Code_I3	New	2	HLD 2.7.4.2; Exporting	DD 3.1.33; Exporting d	dlgG2aFilt
3	Manage Inquiry reports	4.Rel_Code_I3	New	3	HLD 2.8.1; Reporting d	DD 3.1.37 to 3.1.39; R	dlgOxxFilt

 19

Trainer show sample of RMS here and introduce overview about this document.

Requirement Process
Manage Requirements Changes & Status

- Requirements change (CR – Change request)
  - The priority of requirements from different viewpoints changes during the development process
  - Customers may specify requirements from a business perspective that conflict with end-user requirements
  - The business and technical environment of the system changes during its development
- Requirements change process

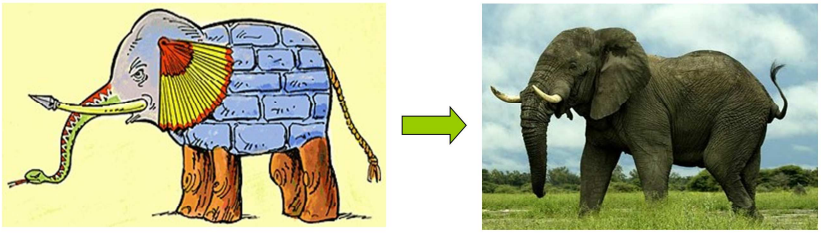


Requirement Clarifying

- PM/TL/BA present the SRS to members in the team
- Self study related materials: top-down approach
- Using SRS review checklist
- Clarify unclear item(s) using Q&A
- Discuss with other members
  - To clarify or confirm your understanding
  - Media: direct discussion or via team brainstorming
- Inform the PM/TL/BA about
  - Any requirement conflicts
  - Changes, comparing to the last version

Requirement Clarifying
Clarifying requirements via Q&A 1/4

- Why do we need Q&A?
  - Problems of understanding
  - Want for knowledge, must be ask



Refer: Q&A Making Guideline

Requirement Clarifying
Clarifying requirements via Q&A 2/4

- How to make Q&A effectively?
  - Identify the issue: unclear, get for more information, etc.
  - Check in all documents that customer supplied to make sure your question has not solved;
  - With technical question, check your team /group/company or ask “Google” to solve it before asking out
  - Give the cross-reference clearly, completely
  - Attach sample screen, demo, give your suggestions if any
  - Convert questions to Y/N or multiple-choice types if possible
  - In Q&A, give deadline that you want to receive the answer. It there is no answer until the deadline, what is impact?
  - Take the receiver to re-read the question before sending

Requirement Clarifying
Clarifying requirements via Q&A 3/4

- Q&A focus:
  - Question for idea conveyed by words like: maybe, generally, etc.
  - What is the TBDs - Ask PL to remove all TBDs before handling to you for designing or coding
  - Conflict between requirements. Read the requirement matrix
  - Don't make assumptions, just ask your PM, PL or BA

Requirement Clarifying
Clarifying requirements via Q&A 4/4

- Follow up the Q&A
  - Track the discussion history for easier following up
  - If your question has not been replied or impacts to your task must be report to your PM, BA, or TL immediately
  - Keep in mind your manager/customer are very busy. So it is necessary to remind them about your pending issues daily, weekly. If not, your task will be impacted
- Template: refer [Template_QA Management Sheet.xls]

Common Practices, Problems

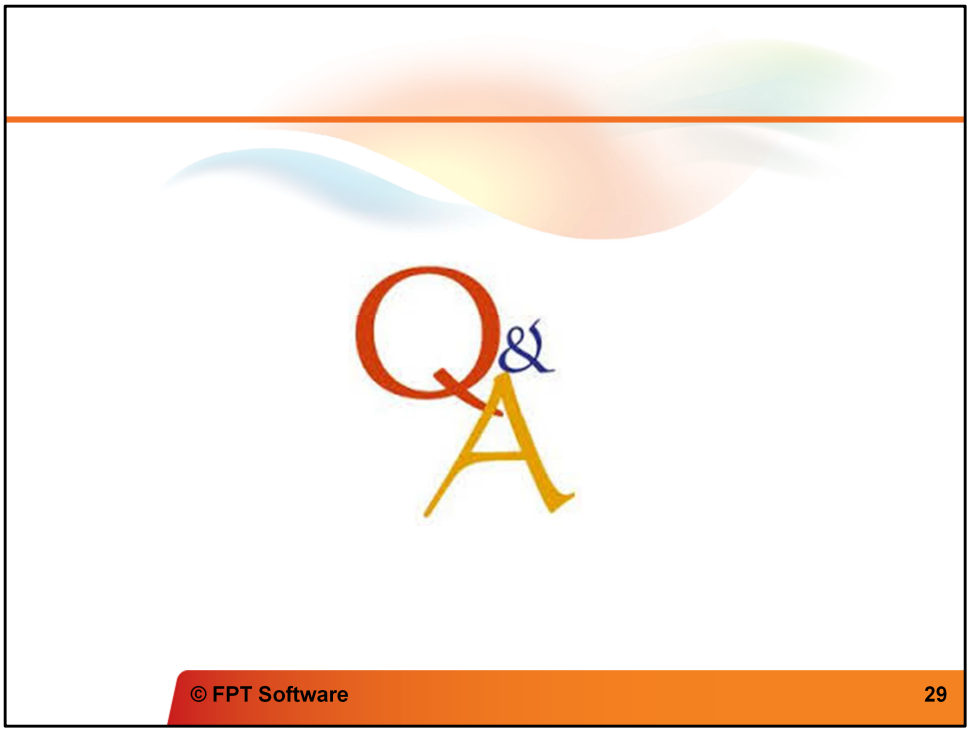
- Common issues:
  - Requirement isn't clear. Don't understand customer mean.
  - Requirement analysis is not documented/centralized recorded -> misunderstanding
  - How to verify the understanding of team members about requirement

Common Practices, Problems

- What should we do
  - Maintain SRS
  - Communicate with the customer via onsite
  - Don't make assumption. Must confirm with customer about specs more and more about what are still not clear or too general.
  - Should involve all members to investigate requirements from the beginning of the project, not only PL.
  - Conduct meeting to verify the requirement understanding.
  - One meeting to introduce the requirement/design.
  - Another meeting to verified the understanding of team member. Every one have to present their understanding.

Resources & References

- Resources & References
  - Requirement development & management process
  - Q&A management sheet template
  - SRS review checklist
  - RMS template
  - SRS template


Q&A
© FPT Software
29