Checklists as a Tool for Entry-Exit Phases of SDLC

Ranjith Govindaraja Menon

ranjith_menon@infosys.com

Infosys Technologies Limited
Mysore, India

This paper showcases how we have developed a tool for capturing the needs of entry and exit phases of a System Development Life Cycle (SDLC).As part of our endeavor towards high end process consulting, we came across cases where the customer was not able to capture the requirements to the full potential. Moreover there are was need to capture special requirements, statutory and legal requirements, etc. in compliance with ISO, BS and Euro regulations. At a corporate level in Infosys the requirements elicitation and gathering tools was available more-or-less at a very generic level typically catering to simple software projects like web site deployment, etc.

For a heavy engineering customer involved in complex PLM suite of applications it was not sufficient. Hence there was a need to develop a tool for capturing the same at the entry and exit phases of the lifecycle to elicit all the typical scenarios. The same was arrived at after having a series of brainstorming sessions with process & quality and technical experts.

Introduction

“No single factor is responsible for more wasted effort, rework, or failed projects than inadequate requirements” - Forrester Research - January 2006

As per IEEE Requirement is:

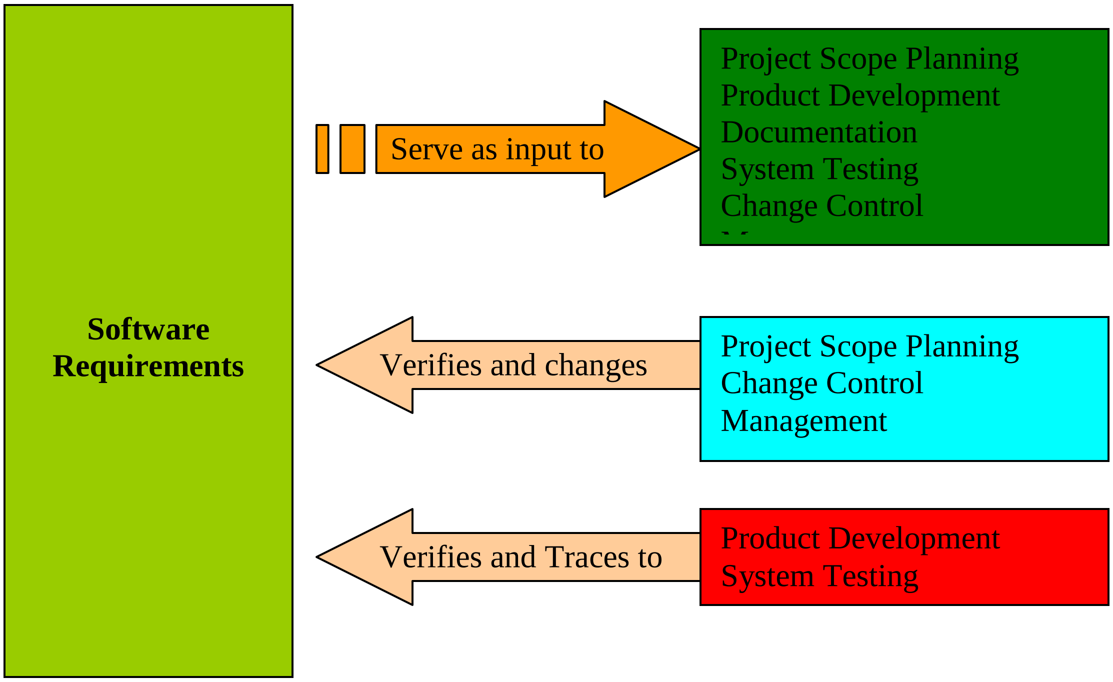
1) A condition or capability needed by a user to solve a problem or achieve an objective.

Good requirement engineering processes can give great benefits. The greatest is by reducing rework during other stages of development and throughout the maintenance phase. Sound Requirements processes emphasize a collaborative approach to product development, involving multiple stakeholder perspectives in the project.

Programmers benefit from a clear understanding of the product required leading to efficient detailing and impact analysis. As a result the defect injection is minimized leading to easier maintenance. An effective change control process would minimize the adverse impact of requirements changes.

Customers are satisfied with their ideas being implemented as per expectations/understanding. Finally, documented and unambiguous requirements greatly facilitate system testing, increasing the chances of delivering high-quality products that satisfy all stakeholders.

Software Requirements processes



Good requirements

A requirement is generated owing to a new/modified business need. The process of communication of the business need is called “Requirement Development”. This includes the discussions, JAD (Joint Application Session) and Doubt clearing sessions amongst the Customers, Developers and Business Analysts.

Once requirements are defined, there can be further changes or modifications in the initial specifications. The process of tracking any such changes is covered under “Requirements Maintenance”.

The importance of a well drafted requirements document can’t be over emphasized since the success of all the phases that follow in software cycle wiz, development, testing etc. are highly dependent on this.

Characteristics of well defined requirements:

- ✓ Complete
- ✓ Correct
- ✓ Feasible
- ✓ Necessary

- ✓ Prioritized
- ✓ Unambiguous
- ✓ Verifiable

Requirements Development and Maintenance

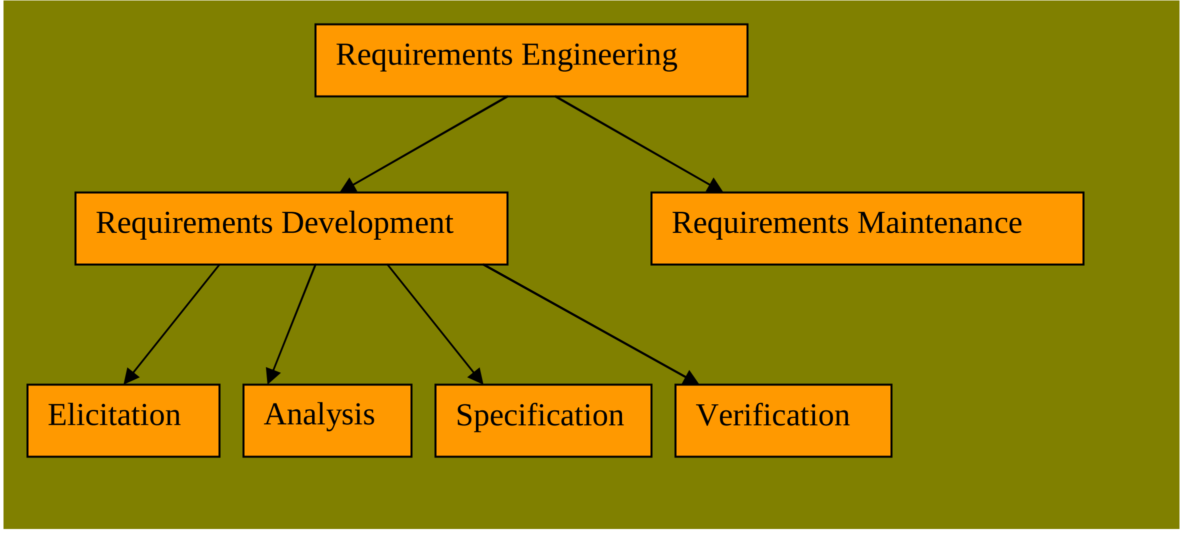
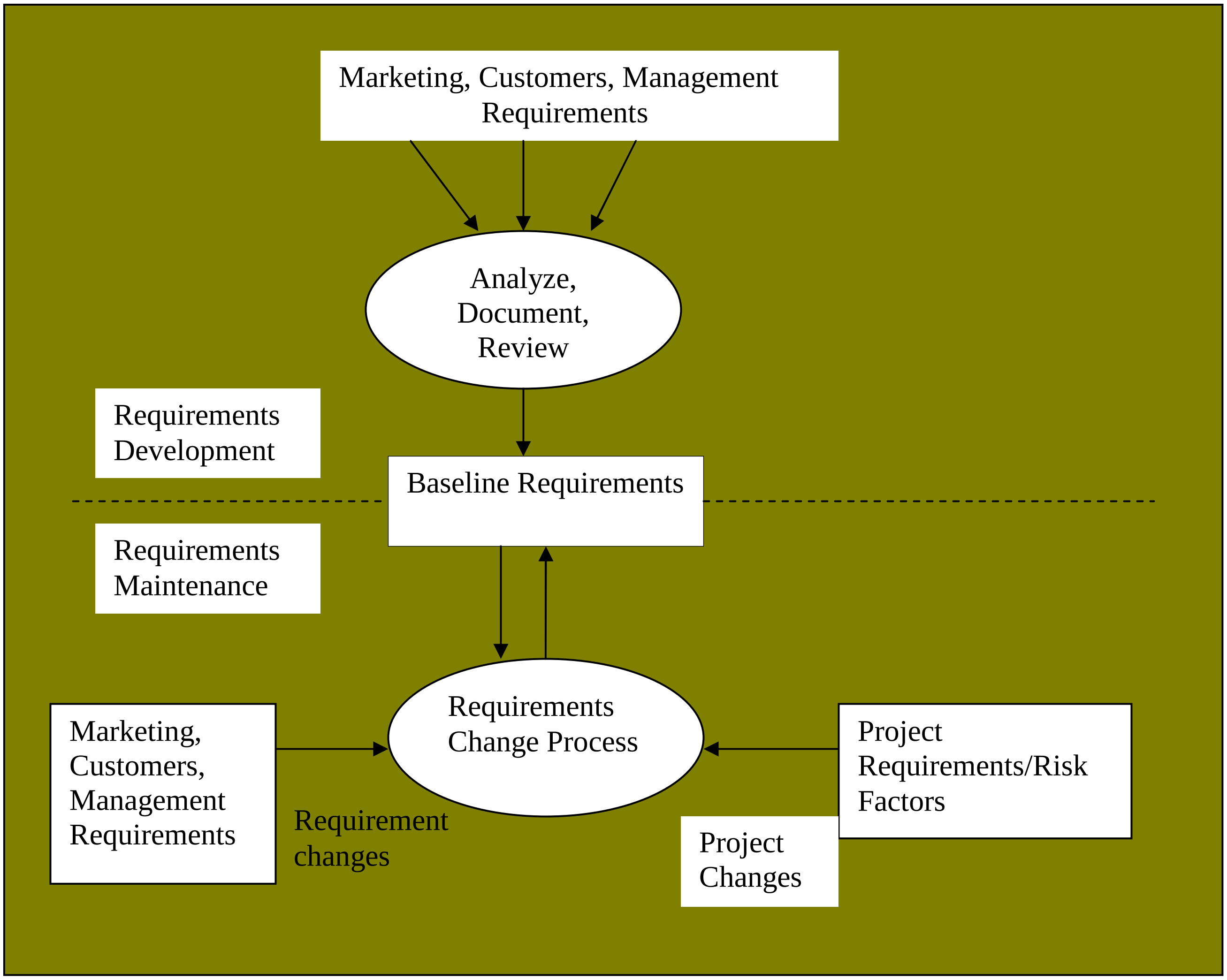


Fig 1-1

As shown above, entire domain of software requirements engineering can be sub-divided into "R*equirements* development" and "R*equirements maintenance*"
Requirements development can be further divided into elicitation, analysis, specification and verification. These sub-disciplines involve all the activities involved with gathering, evaluating and documenting the requirements for software or product.

Figure 1-2 below provides detailed view of the transition from requirements development to requirements management.


Marketing, Customers, Management Requirements
Analyze, Document, Review
Requirements Development
Baseline Requirements
Requirements Maintenance
Marketing, Customers, Management Requirements
Requirements Change Process
Project Requirements/Risk Factors
Requirement changes
Project Changes

Key Requirements Process Elements

Development Phase

- ✓ **Requirement Checklist** - A list of important activities or actions to ensure that important details are not missed.
- ✓ **Project level Scope Document** – A concise, high-level description of the product's business requirements that provides a reference for prioritizing requirements and changes.
- ✓ **Use case Template** – This should document each task a user wishes to perform with a software system including task description, exceptions involved and any alternate paths.

- ✓ **Requirement Specifications and Use case checklist –** Involves formal check on the consistency of the requirements. The most commonly found errors the requirement documents can be listed down at one place. Key characteristics of requirements specifications are :
    - Complete
    - Consistent
    - Modifiable
    - Traceable

Maintenance Phase

- ✓ **Requirement Traceability Matrix Template -** In order to ensure exhaustive coverage of the complete requirements, each product component needs to be traced back to a set of business rule and vice-versa.

- ✓ **Change Control Procedure –** Once base lined, the requirements can change owing to several factors. Once the proposed changes are approved by the Change Control Board (CCB), the same are included in the requirement documents in a new version. Hence, the need for an effective change controls process so that all the changes added to original requirements are tracked and incorporated in final product.

- ✓ **Change Impact Analysis Checklist –** While implementing any changes to the initial requirements, following checks must be performed:

    - Do any existing requirements in the baseline conflict with the proposed change?
    - What are the businesses or technical consequences of not making the change?
    - What are possible adverse side effects or other risks of making the proposed change?
    - How will the proposed change affect the sequence, dependencies, effort, or scheduled duration of any tasks currently in the project plan?
    - How much effort has already been invested in the project will be lost if this change is accepted?
    - Identify any user interface changes, additions, or deletions required.
    - Identify the design components that must be created, modified, or deleted.
    - Identify existing unit, integration, and system test cases that will be added, deleted or required.
    - Identify work products that must be reviewed after they are modified.

<u>As a business case:</u>

Requirements form the most vital component for any SDLC.
As per SEI of Carnegie Mellon:

- Requirements are the first case for failure.
- Many requirements' engineers are not qualified
- Cost of Quality tends to be on the higher side – the later we tend to fix it.
- Requirements typically contain defects
- Requirements impact all down-stream work
- Requirements form the primary reason for failure

Requirements form the first engineering chance for failure.
The Standish Group's annual CHAOS report indicates three of the top five reasons for project failure are related to requirements. In addition, requirements errors are a primary factor behind most rework efforts, which according to industry statistics, can add up to 40 percent of the total development effort within a given project

Many requirement engineers aren't qualified enough

Requirement Myth

Most managers tend to think that anybody can do requirement engineering
Requirements engineers lack training in:

- Requirements tasks
    - Identification
    - Analysis
    - Specification
    - Management
- Requirements techniques(Use Case Modeling)
- Requirements tools

Requirements Contain Defects

The percentage of defects originating during requirements engineering are estimated as:

- 50%(Karl Wiegers,2001)
- 42%(A Wingrove)
- 60-64%(requirements and design – EBG Consulting)

Requirements typically lack

Cohesiveness,Completeness,Correctness,Consistency,Currency,Essential,Feasibility,Lack of Ambiguity,Relevance,Testability,Usability,Validatability

Requirements Engineering Impacts

- Scope Management
- Architecture(architecturally significant requirements)
- Design and implementation
- Testing
- Quality Engineering(determines defects)
- Safety and Security requirements
- Reuse and Training

Requirements Defects snowball

Defect Costs are Excessive

Requirements engineering defects cost to correct once fielded:

- 50-200 times as much (Barry Boehm, 1988)
- 10-100 times as much(Steve Mc Connell,2001)
- 15 times as much(IBM institute)
- 10 times as much(Hughes Aircraft)

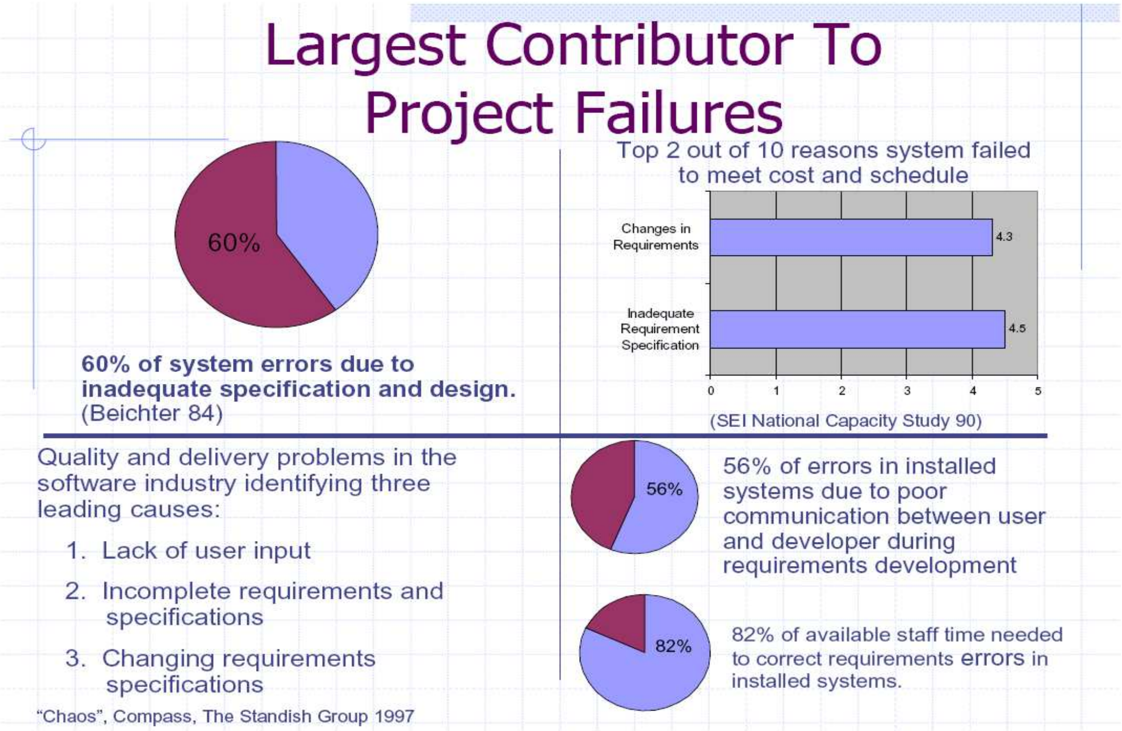
Reworking requirements defects costs

- 40-50% of the effort(Caper Jones)
- 80% of the effort(Karl Wiegers 2001)

Bad Requirements cause failures:

- Significantly Over budget
- Significantly past schedule

- Significantly reduced scope
- Poor quality applications
- Not significantly used once delivered
- Cancelled



Largest Contributor To Project Failures

60% of system errors due to inadequate specification and design. (Beichter 84)

Top 2 out of 10 reasons system failed to meet cost and schedule

(SEI National Capacity Study 90)

Quality and delivery problems in the software industry identifying three leading causes:

1. Lack of user input
2. Incomplete requirements and specifications
3. Changing requirements specifications

"Chaos", Compass, The Standish Group 1997

56% of errors in installed systems due to poor communication between user and developer during requirements development

82% of available staff time needed to correct requirements errors in installed systems.

As part of our endeavor towards high end process consulting, we came across cases where the customer was not able to capture the requirements to the full potential. Moreover there are was need to capture special requirements, statutory and legal requirements, etc. in Compliance with ISO, BS and Euro regulations.

At a Corporate level in Infosys the requirements elicitation and gathering tools was available more-or-less at a very generic level typically catering to simple Software projects like Web Site deployment, etc.

For a heavy engineering customer involved in Complex PLM suite of applications it was not sufficient. Hence there was a need to develop a tool for capturing the same at the entry and exit phases of the lifecycle to elicit all the typical scenarios.
The same was arrived at after having a series of brainstorming sessions with process & quality and technical experts

Details of the tool

We have developed the checklists to capture requirements for the Entry and handover checklists for exit points of the System Development Life Cycle (SDLC)

1. *Business Projects => Software Development*
2. *Software Development => Deployment and Training*

At the **entry phase** we developed the criterion to cover **–**

1. Project Information --- Refers to the project initiation and the related processes
    a. Project Plan / Charter
    b. Business Commitment, Stakeholder Sign off
    c. Rollout Plan
    d. Project Organization
    e. PID --- Project Initiation Document
2. Return on Investment(ROI) --- Refers to the project analysis from a monetary point of view
    a. Investment Plan
    b. Savings Plan
    c. ROI Calculation
3. Business Requirements --- Elicits the requirements as required by the business
    a. BR Specification Document (BRD)
    b. Business Processes As-Is
    c. Problem Statement
    d. Business Processes To-Be / Solution Workflow
    e. Requirements' Ranking By Importance
    f. Access Rules
    g. Feasibility Document
    h. Organizational Process Map
    i. Impact Analysis Document
    j. System Requirements
    k. Scope and Boundary Statement
    l. Technical Feasibility Document
    m. Context Diagram, Business events
    n. Screens and Reports
    o. Acceptance Criteria
4. External Interface Requirements --- Gives the external interfaces that are required to be created
5. Operating Environment Requirements --- The requirements related to the operating conditions
    a. Hardware
    b. Software
    c. Network

    d. Communication
6. Performance Requirements --- Requirements related to software performance
    a. Peak and Normal Conditions
    b. No. of users, Response time etc
7. Standards Requirements --- Related to standards to be followed
    a. User Interfaces
    b. Document Standards
8. Special requirements --- As stated below.
    a. Security
    b. Reliability
    c. Legal and Statutory
    d. Backup and Recovery
    e. Data Migration and retention
    f. Transaction Volume and Data Volume
    g. Prototype
    h. SLA/OLA Update Needed
9. Key Constraints --- Related constraints
10. Stakeholders
    a. Business Specification Lead (SPOC)
    b. Business Specification Team (& Alpha, Beta Testers)

And in **exit phase** it covers

1. Deployment --- Check points for Deployment
    a. Has System Requirements documented, reviewed and approved
    b. Has the System Design been reviewed and approved
    c. Traceability Matrix- traceability between requirements, design, and testing
    d. Business Test Results
    e. Product Acceptance/ Test Results sign-off
    f. Compliance with security standards
    g. Data Framework creation/conversion Completion
    h. Installation, User Manual
    i. Transitions Plan for Support activities
    j. Release Notes
    k. Site, Environment (e.g., correct OS, memory, etc.) Preparation
    l. Software Licenses Availability
    m. Existing Bugs / Application Gaps / Service Packs
    n. Maintenance Agreements
2. Training Information --- Training activity steps
    a. Training System Installation
    b. Prepare User profile matrix- role played by users
    c. Scope Definition(with respect to user type, numbers, user modules, mode of training delivery and number of sessions)
    d. User Training Plan

e. User Documentation, Manuals
f. Training Material
g. Inventory of required material for preparing training content

This tool can be tailored for use in any PLM/SDLC Project and Planning phase.

PF the tool embedded

Checklists

References

1. Forrester Research
2. Software Engineering Institute --- Carnegie Mellon University site
3. http://wikipedia.org

Glossary

BR ~ Business Requirements
BS ~ British Standards
ISO ~ International Organization for Standardization
OLA ~ Operations Level Agreement
OS ~ Operating System
PID ~ Project Initiation Document
PLM ~ Product Lifecycle Management
ROI ~ Return on Investment
SDLC ~ System Development Life Cycle
SEI ~ Software Engineering Institute
SLA ~ Service Level Agreement
SPOC ~ Single Point of Contact