

WHITEPAPER

SQS Requirement Development & Management Framework



sqs.com

Authors: Sreenivasan NS (Senior Project Manager & Senior Manager, Client Relations USA)
Bharathi R (Test Manager – Banking Practice)
Elamathy P (Test Lead – Banking Practice)
Ramakrishnan R (Associate Vice President – Banking Practice)
Rajesh AS (Senior Project Manager – Banking Practice)
Balaji CG (Associate Projects Manager – Banking Practice)

SQS India BFSI Limited

Published: August 2015



SREENIVASAN NS

Senior Project Manager & Senior Manager, Client Relations USA
sreenivasan.ns@sqs.com

Sreenivasan NS is a Chartered Accountant & Project Management Professional (PMP) with over 21 years of work experience in IT, Finance and Accounting. He has been with SQS for 10+ years and handled many core banking transformation projects. He has commendable experience in Financial Services and has worked as SME on many projects. Currently he is the Engagement Manager for a global US bank.



BHARATHI R

Test Manager – Banking Practice
bharathi.r@thinksoftglobal.com

Bharathi R is an IT graduate with over 12 years of experience in IT Services and Process Engineering. She has helped to set up test practices for clients and successfully managed many manual and automated testing projects across the Cards and Banking domains. She has worked extensively in Quality Assurance and Process Management and helped to streamline many processes involved in quality management systems. She has been instrumental in driving the requirements repository for many global clients.



ELAMATHY P

Test Lead – Banking Practice
elamathy.p@thinksoftglobal.com

Elamathy P is a postgraduate in Business Administration (Finance) and a Project Management Professional (PMP) with over 10 years of experience in IT Services. She has successfully led many testing assignments for global clients across the Cards, Banking and Treasury domains. She has helped clients to achieve significant cost saving through defect prevention, adopting structured requirement reviews and static testing practices



RAMAKRISHNAN R

Associate Vice President – Banking Practice
ramakrishnan.r@sqs.com

Ramakrishnan R is a Chartered Accountant with over 18 years of experience in IT Services. He has been with SQS for over 12 years and has in-depth knowledge and experience in Core Banking, Cards, Payments, Trade Finance Operations, Forex and Domestic Deposits. He has successfully managed many manual and automated testing projects across the Cards and Banking domains. He has been instrumental in establishing and stabilising Managed Services assignments for many global clients.



RAJESH AS

Senior Project Manager – Banking Practice
rajesh.as@sqs.com

Rajesh AS is a Finance and Banking expert with over 18 years of experience and a PRINCE2® certified IT professional. Part of SQS testing team since 2005, he has handled testing projects for various global clients in the areas of Loans, Private Banking, Cash Management and Core Banking in both offshore and onsite models. He has proven strengths in process assessment and has been instrumental in setting up and establishing testing practices for many global clients.



BALAJI CG

Associate Projects Manager – Banking Practice
balaji.cg@thinksoftglobal.com

Balaji CG is an IT graduate with over 11 years of experience in IT Services. He has helped to set up test practices for clients and successfully managed many manual and automated testing projects across banking domains. He is currently the UAT Practice Manager for a global US bank.

Contents

Management summary	5
Introduction.....	5
Market – current status and outlook	6
Aligning requirement and software development methodologies	6
Requirement development – best practices	7
Managing change – in a fast changing world	9
Market tools and their benefits	10
The requirements maze – end-to-end requirement tracking & validation	11
Metrics that can drive success	12
Requirement repositories – why & how to build them.....	13
Conclusion	14
References	14

Management summary

“Software defects found in production cost four times as much to fix, as defects found during system testing; and two hundred times as much to fix, as defects identified during requirements definition.”

Gartner 2013 [1]

Solidified requirements are the foundation for the success of product development. Ideally, requirements are not expected to change except for market-induced/regulatory changes. However, this happens only in Utopia, and in reality, require-

ments keep changing until the product is put into operation. This means requirements have to be clearly defined and documented, and well managed to ensure changes are kept to a minimum and are continuously integrated with other components of the software development life cycle (SDLC).

This whitepaper intends to showcase SQS BFSI's Requirement Development & Management Framework (“SQS RDMF”), a collation of time-tested requirement documentation & management best practices, and demonstrate how it helps organisations to manage requirements better and deliver business needs in time and on budget.

Introduction

Requirements definition & management has become more challenging with the faster pace of technology advances. According to the Standish Group's Chaos report [2], 73% of projects were either cancelled or failed to meet expectations due to insufficient definition and analysis of requirements.

Irrespective of the maturity of the SDLC processes, the following basic issues still prevail in the industry and derail the time, cost and quality of any product development.

- No structured process for requirement development
- Gaps in requirement elicitation and analysis
- Diluted ownership for requirements

- Frequent changes to requirements resulting in time-consuming reworks and budget overruns
- Requirements are not well managed through the project life cycle

“SQS RDMF” provides a detailed methodology to develop and manage requirements throughout the SDLC by:

- Establishing and following a requirement development process suitable for different development methodologies
- Integrating best practices by which requirements are gathered, elicited, analysed, documented and finalised.

Market – current status and outlook

The market for new and enriched software is ever-increasing. The drivers for this market are changes in technology, new markets, new products and services, market demand and legal requirements, etc. The industry has now matured through many lessons learnt around SDLC. Important among them is managing requirements by using a defined process and integrated & collaborative tools. There are organisations that provide requirement management services either as a separate service or part of overall project management.

Industry is extensively using general documentary software such as Microsoft for requirement definition and management. But major software development organisations and IT departments of other organisations have started using requirement management software. Gartner estimates that demand for such software will grow between 5% and 7% annually by the end of 2016 [1].

Aligning requirement and software development methodologies

Though the importance of any phase in SDLC cannot be overemphasised, requirement is the phase that should be nurtured to ensure correct and clean requirement flows into subsequent phases. Requirement definition may be categorised as follows for planning and control purposes.

- a. A requirements development process (RDP) that describes how to
 - Identify appropriate stakeholders
 - Gather and document requirements
 - Analyse and prioritise requirements
 - Review and finalise requirements

It addresses the project purpose from business strategy, process and technology perspectives.

- b. A requirements management process (RMP) that describes how to
 - Manage and control requirement changes
 - Track requirements status
 - Trace requirements across SDLC phases
 - Apply configuration management

It helps the project to build the right product.

Both RDP and RMP should be bespoke to the needs of the organisation, product and development methodology followed. For example, in a waterfall model, RDP would be completed and RMP would be initiated along with the software design phase and continued until the software is put into production. In the case of an agile model, both RDP and RMP would be initiated and applied iteratively until the entire product is put into production.

It is imperative that any RDMF should be thought through and flexible for customisation. Such customisation should consider the following aspects for proper alignment:

- Development methodology (Waterfall, agile, etc.)
- In-house or 3rd party development
- Off-the-shelf or bespoke product
- Regulatory compliance
- Templates
- Level of documentation and detail

Requirement development – best practices

Software projects suffer most when requirements change, setting off chain reactions of revision and rework which ultimately impact the schedule and the cost.

As per the Standish Group’s Chaos report [2], which surveyed 9,236 IT projects, the top 3 causes of project failure were lack of user input, incomplete requirements and changing requirements. An industry analysis shows that 82% of rework is due to changes in requirements and 40% of overall project

costs are due to rework. This backs up the need to instigate a proper requirements framework, enforcing effective RDP to avoid requirement changes due to process ineffectiveness.

A thorough RDP, RMP and supporting technology/ tools should be in place and should be properly integrated for best results.

The RDMF framework proposes the “EDC cycle” for requirement development as set out in Figure 1.

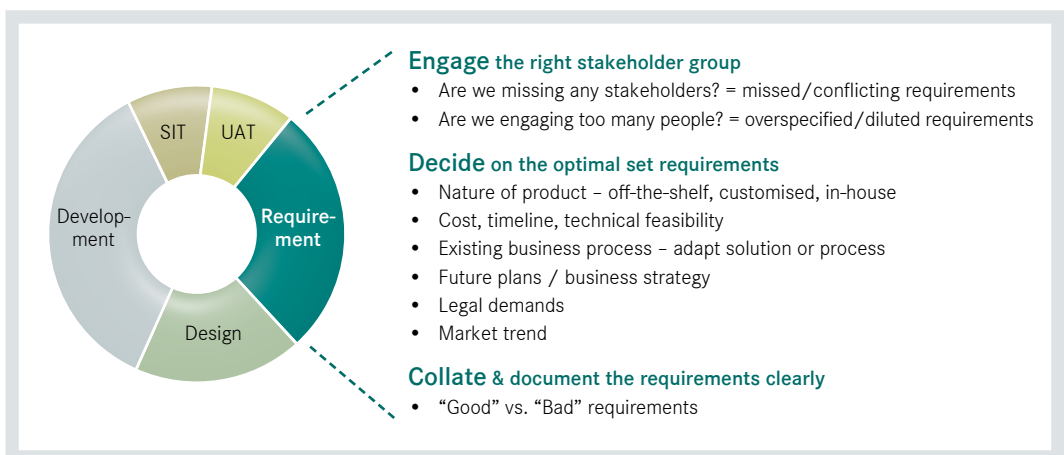


Figure 1: The EDC cycle for requirement development

Engage the right stakeholder group: There is useful guidance available on stakeholder engagement in sources such as the BABOK [3].

Decide on optimal requirements: The RDMF framework recommends the Goal Decomposition approach (Figure 2) which is based on the principle of breaking the overall project goals down into manageable objectives. Requirements are derived from objectives in terms of conditions or capabilities

that should be met in order to achieve the objective. This helps to ensure that requirements are aligned with overall project goals and facilitates the prioritisation of requirements that are critical to a project’s success.

Collate “Good” requirements: The success of requirements gathering depends on adequate and appropriate collation and documentation of finalised requirements (Figure 3).

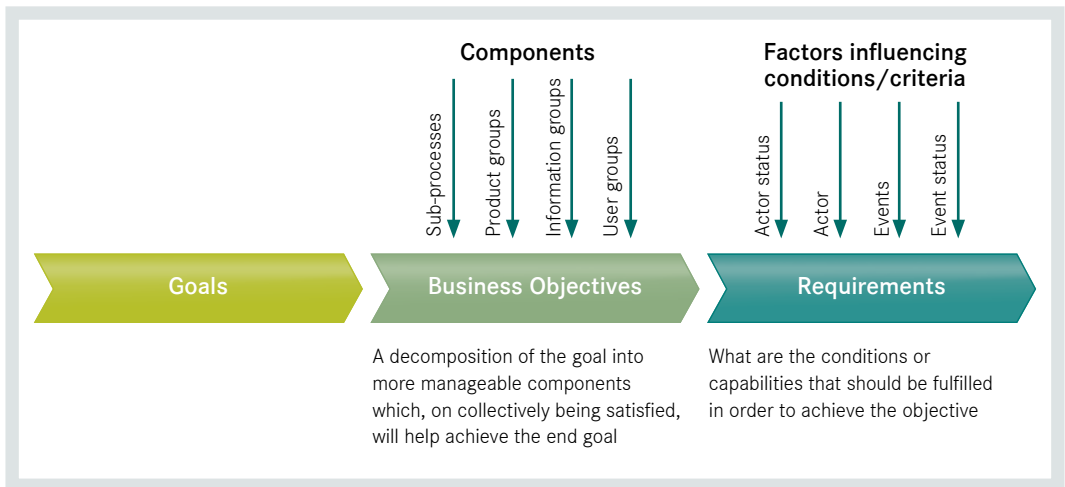


Figure 2: Goal Decomposition approach

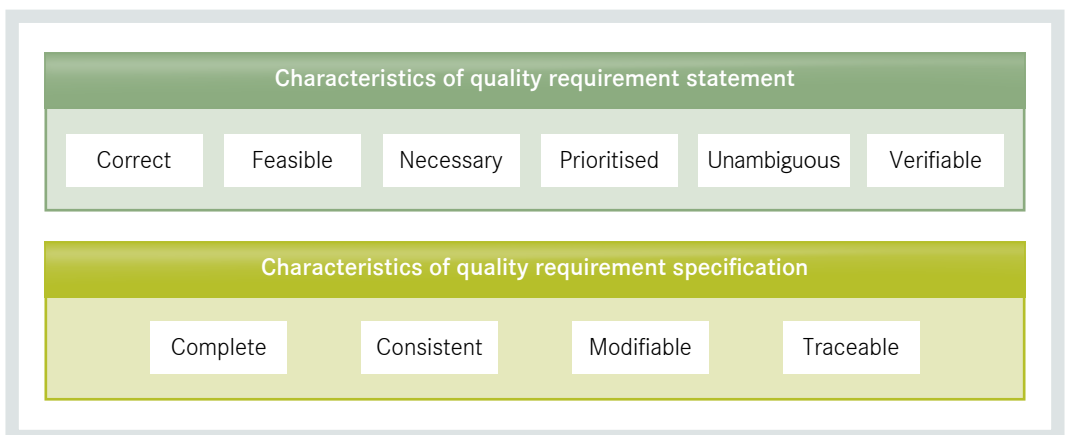


Figure 3: “Good” requirements

Managing change – in a fast changing world

Change control in requirements management is the process that reviews, regulates and manages any changes required to the requirements baseline. The Change Control Process (CCP) runs from project inception through to completion and needs to be followed irrespective of the nature of change – big/small, simple/complex (Figure 4). The key benefit of this process is that it allows for changes to be considered in an integrated fashion while reducing risk, which often arises from changes being made without consideration of the overall project scope/objectives, timelines and budget.

Changes may be requested by any authorised project stakeholder. Changes should be recorded in written form and entered into the change management system. Every documented change request needs to be reviewed, analysed for impact and either approved, deferred or rejected. When required, the CCP includes a change control board. All change requests, approved or rejected, will be updated in the change log and formally communicated to all stakeholders. Approved change requests will be implemented after reviewing the requests in terms of cost, time and quality.

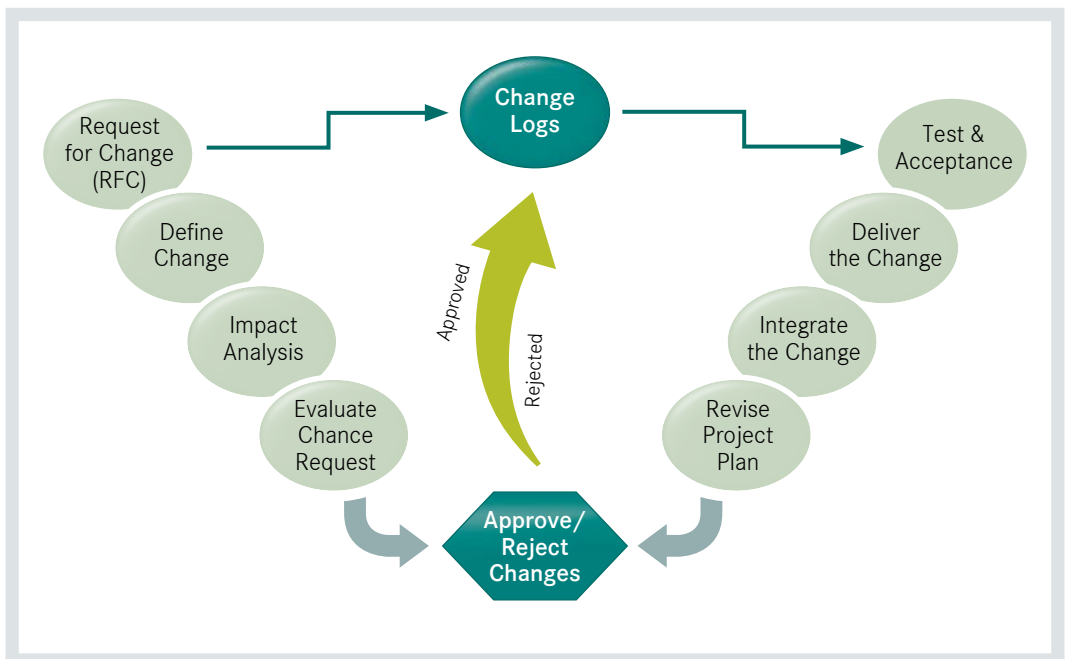


Figure 4: Change Control Process

Market tools and their benefits

Some of the common challenges in requirement management via the document-based approach are:

- Distributed information
- Reusability
- Tracking changes and version maintenance
- Change communication
- Managing scope
- Review process (emails, multiple versions of documents, meetings, etc.)

Using requirement management (RM) tools will enable requirements to be captured, maintained and approved in one central location. Requirements can be seen in real time and instant feedback can be provided. The mechanism of tracking clarifications, feedback and taking action by changing/deleting a requirement is robust in most RM tools (Table 1).

Stakeholders will be able to provide instant feedback by analysing upstream and downstream impacts, saving time and effort by avoiding rework in the development and testing phases. The RM tool will also enable stakeholders to assess the

scope of the release by looking at the requirements status and employing logical grouping to ensure a meaningful release.

There are predefined workflows where the tool can capture the following information:

- Requirement source
- Requirement assigned to
- Requirement changes
- Target for completion
- Target for review etc...

Automatic emails and reminders are triggered to stakeholders regarding the progress of every stage. There are a whole host of software tools on the market that enable requirements to be captured and managed in a structured and collaborative fashion. Most of these tools are capable of handling the entire application development life cycle (ADLM) and are chosen based on requirements. Examples of such tools have been provided below. Readers are encouraged to refer to the Gartner Magic Quadrant [1].

Levels	Tools	Remarks
Enterprise level	<ul style="list-style-type: none"> • IBM Rational DOORS • Borland Caliber 	Targeted at large, enterprise-level implementations; very expensive and loaded with features for companies
Mid-market level	<ul style="list-style-type: none"> • Accompa • Jama 	Good balance between the feature set of the “enterprise-level” tools listed above and the ease-of-use of the “entry-level” tools listed below
Entry level	<ul style="list-style-type: none"> • Atlassian JIRA • FogBugz 	Affordable and can be used to manage requirements in a structured fashion – especially within smaller organisations

Table 1: Market tools

The requirements maze – end-to-end requirement tracking & validation

Tracking the requirement status means that owners can be assigned, and requirements monitored and controlled at different phases of the software life cycle from inception to delivery, including statuses that are in-progress, yet to start, deferred and obsolete.

The end-to-end requirement traceability process defines the tracking of the life of requirements both horizontally and vertically, in forward and backward directions throughout the SDLC, to ensure that all requirements can be traced back to their source as well as forward to the delivered functionality, providing robust impact analysis. The key benefit of this process is that it ensures that all the agreed requirements are delivered without any extraneous effort; all changes to the requirements during the course of the projects are tracked meticulously, capturing details of such along with the impacted requirements.

Horizontal traceability can be established by developing a relation from ‘Release >> Business

Requirements >> Functional Requirements >> Test Cases >> Test Results >> Defects’. This will help to address any orphan requirements. Vertical traceability can be created by establishing the dependency between the requirements which enables identification of the impacted requirements in case of changes.

SQS RDMF provides a unique solution by combining both the processes of requirements status tracking and requirement traceability (both vertical & horizontal) (see Figure 5). An easy visual way of tracking any ‘unacknowledged change’ can be demonstrated with an example. Once a relationship is established between a business requirement (RQ-1) owned by a product manager and a functional requirement (FR-1) owned by a development manager, then any changes to either RQ-1 or FR-1 will automatically generate a red suspect trace, indicating a change. If the two stakeholders review and agree the change, then one of them can clear the trace which will automatically convert the status to green.

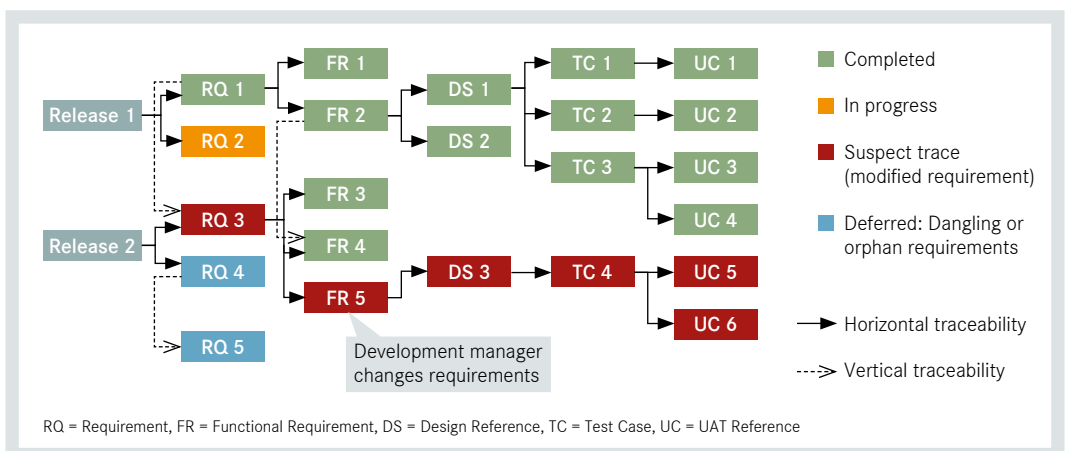


Figure 5: End-to-end requirement tracking process

Metrics that can drive success

The metrics for requirement management fall under two categories:

- Process metrics: used to measure the effectiveness or efficiency of the requirement management processes
- Product metrics: used to measure the quality of requirements

Metrics planning is a 2-step process (Figure 6):

1. Deciding on the problem to be solved and hence what to measure
2. Deciding how to use the metrics

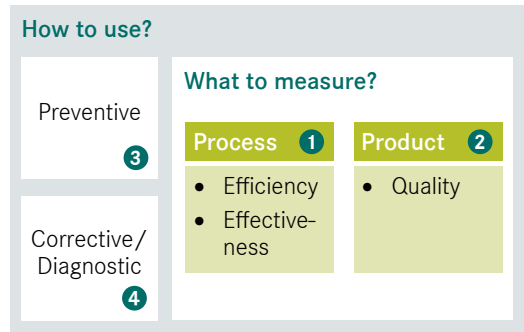


Figure 6: Metrics planning

This paper provides measures that can be derived from the SQS RDMF which will help to achieve these objectives (Table 2).

	Measure	Representation	Measure & Reporting Frequency
Process	Change prevention 1	<ul style="list-style-type: none"> Change due to gap in requirement elicitation Change in business need 	After requirement change 4
	Effort per requirement 1	Person hours per requirement (from initial requirement discussion to final requirement sign off)	After requirement sign off 4
Product	Impact on project goal (e.g. reduce cycle time of a workflow) 2	<ul style="list-style-type: none"> High Impact Medium impact Low impact 	<ul style="list-style-type: none"> During requirement gathering for optimisation 3 During subsequent phases 3
	Requirement quality 2	<ul style="list-style-type: none"> Ambiguous Conflicting Missed Incomplete 	After requirement review 4
	Requirement optimisation (e.g. workflow 1 faster than workflow 2) 2	% Requirements that could have been optimised	After requirement review 4

Table 2: Measures

Requirement repositories – why & how to build them

A requirement repository is a method of storing approved requirements along with their attributes and is available to all stakeholders (Figure 7).

Objectives of a well-designed requirement repository:

- Cater to the requirement documentation & collaboration needs of an ongoing project (short term)
- Adapt the IT solution to evolving industry best practices without having to collate and document requirements from scratch
- Faster solution fitment by facilitating quick mapping of product features to requirements
- Easier change management – provide quick change impact view

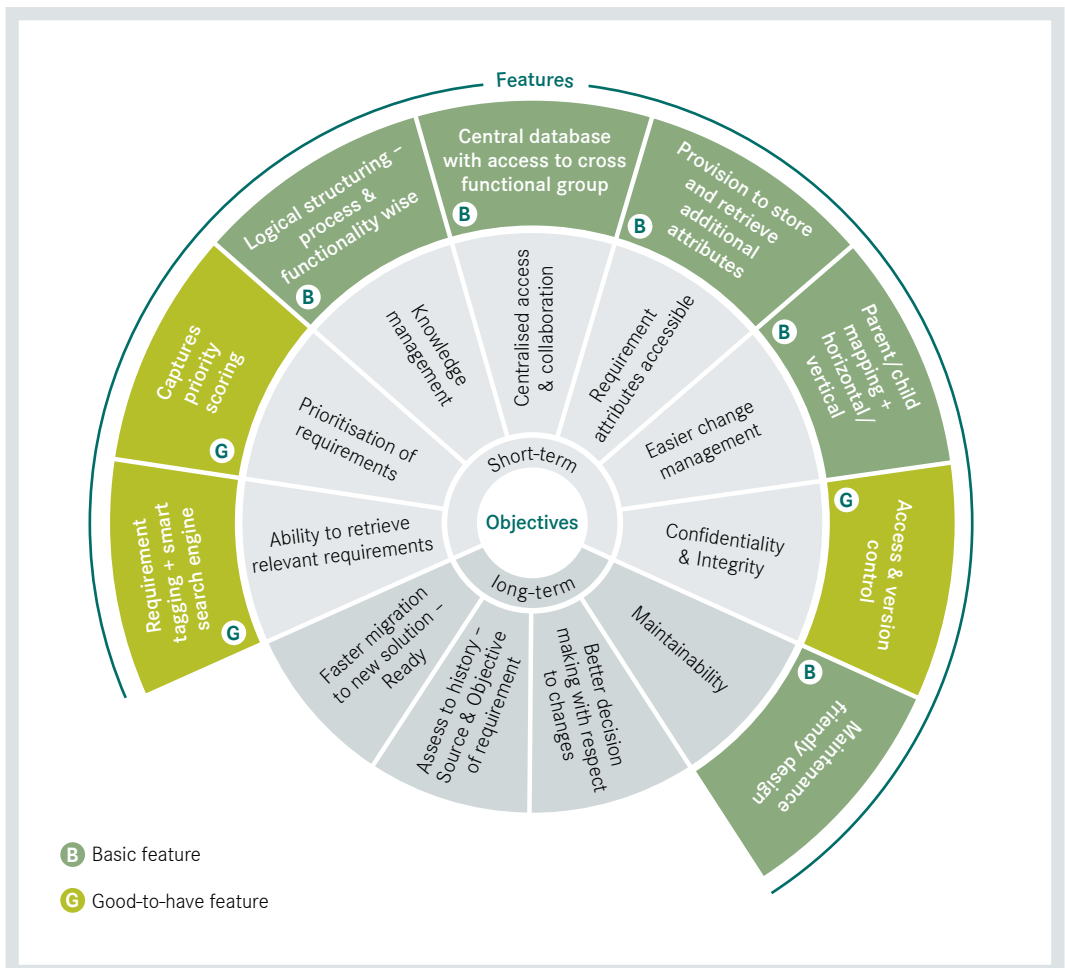


Figure 7: RDMF Repository – features vs. objectives

Conclusion

Requirements form the foundation of the software development process and a well-structured requirement development and management framework is imperative to the success of every IT project.

The SQS RDMF collates best practices on requirement development and management that help to build a strong foundation of requirements benefiting subsequent phases of the project and even subsequent product upgrades or fitment of new solutions.

The key principles are:

- Structured requirement development and management methodology, aligned with the SDLC model
- Focus on overall project goals and objectives while deriving requirements
- Smart documentation that ensures clear communication to all stakeholders
- Collaboration throughout the requirement gathering/change process
- Structured requirement change management
- Traceability across requirements that facilitates decisions and change management
- Metrics to drive best practices
- Repositories aiding faster maintenance, upgrades or migration

References

- [1] Gartner (2013). Magic Quadrant for Application Development Life Cycle Management.
- [2] The Standish Group (1995). Chaos Report.
- [3] K. Brennan (2009). A Guide to the Business Analysis Body of Knowledge. International Institute of Business Analysis (IIBA)

© SQS Software Quality Systems AG, Cologne 2015. All rights, in particular the rights to distribution, duplication, translation, reprint and reproduction by photomechanical or similar means, by photocopy, microfilm or other electronic processes, as well as the storage in data processing systems, even in the form of extracts, are reserved to SQS Software Quality Systems AG.

Irrespective of the care taken in preparing the text, graphics and programming sequences, no responsibility is taken for the correctness of the information in this publication.

All liability of the contributors, the editors, the editorial office or the publisher for any possible inaccuracies and their consequences is expressly excluded.

The common names, trade names, goods descriptions etc. mentioned in this publication may be registered brands or trademarks, even if this is not specifically stated, and as such may be subject to statutory provisions.

SQS Software Quality Systems AG
Phone: +49 2203 9154-0
Fax: +49 2203 9154-55
info@sqs.com | www.sqs.com