

## Quality Solution Delivery Framework for Risk and Project Management

Author Farhat Yamin

Contact No: +92-315-5179707

**Abstract:** This study is presenting a *Quality Solution Delivery Framework* which is a blend of Project Management and Risk management methodologies. All core phases of project management either utilized by the traditional approach or modern framework like agile, are envisioned into this framework. Besides taking care to include the core features, Risk Management phase has also been blended in each of the phase to form a best framework which is compliant of Operations Clause of ISO 9001:2015 along to take care of Project and Risk Management features simultaneously. Qualitative methodology has been adopted to collect the data which was transformed into the information base to come up with the analytical views. To cope up with limited time constraint, two sample projects of same nature were taken into research account. One was observed from initial stage till design phase while other was taken from built till sign off.

The formulized framework was compared with modern vastly used development life cycle methods like waterfall, prototyping and spiral. And revealed that all features of these SDLC are amalgamated and settled appropriately in suggested model to get paramount results.

**1. Introduction:** As the current business era is based on Solution delivery performance management, so based on this imprint, Information technological companies become more profound of attaining high rate of project success. Grabbing business with satisfying interested parties, is getting much more important to exhale into the corporate sector. Organizing the functional and operational area within limited resources become a big challenge. In this scenario, project management and risk management are the two helping factors to retain the business skin safer.

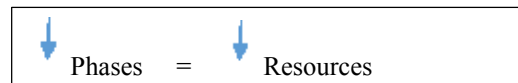
First pillar of Quality Solution Delivery Framework is a Project Management, refer the discipline of initiating, planning, executing, controlling and closing the work of team to achieve specific goals and meeting specific success criteria at the specified time (en.wikipedia.org/wiki/Project\_management). A goal with set scope of work and under limited resources which provide an output in the form of Solution delivery or a product is considered to a project. An IT based goal which is expected to deliver a software is said to be a software delivery project and also called an IT project. All IT projects are required to get through a specified sequence of activities. These activities are based on small chunks of a project major activities also called Work break down structure(WBS) and every WBS is produced a result or may be an input of another WBS. This sequence of such activities is called the Software Development Life Cycle (SDLC).

The second pillar of this study is the Risk Management can be described as it involves understanding, analyzing and addressing risk to make sure organizations achieve their objectives. So it must

be proportionate to the complexity and type of organization involved. Risk Management is an integrated and joined up approach to managing risk across an organization\Project and its extended networks. ([www.theirm.org/the-risk-profession/risk-management](http://www.theirm.org/the-risk-profession/risk-management)). So it is an immense need to the day to take care of the risk factor and implement highly proficient risk management process which could help to achieve the goal ultimately.

Every SDLC and model yet introduced is treating Risk Management and Project management as two separate entities. Risk Management is usual address in one phase either planning or when the risk occurs really. But the whole process has never been addressed to discuss as an embedded part of the project management. Whereas this study is elaborating both concept under one umbrella where all phases of both concepts are rationally addressed. A complete Solution delivery life cycle has been pin down to get optimum results.

This life cycle is consisting of 7 phases instead going through two life cycles comprises of multiple phases. The rational is



Based on this approach, more factors have been covered in less phases. And the risk impact and rating eventually reduced as the project goes on.

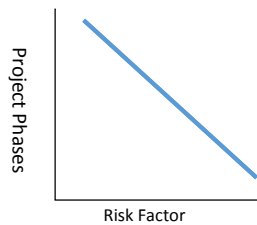


Figure 1: Project Phases and Risk Factor Marginal Dimensional Relationship



Figure 2: Quality Solution Delivery Framework

## 2. Purpose:

Introduction of a Modern Quality Solution Delivery Framework of blended flavor of Risk and Project Management Processes

**3. Limitations:** The study is focused on the IT Sector only which involves in Software Development Life Cycle. To complement this study, qualitative research technique is adopted in which IT project Manager and Product Managers are visited to gather the data. Due to the time constraint, selected sample projects are investigated during the planning and design phase only.

## 4. Quality Solution Delivery Framework (SDF)

**Primer:** Quality Solution Delivery Framework is an intermingled approach of Project and Risk Management Processes. According to the market survey, normally both phases are consisting of 13-18 Phases. But this standard is covering all aspects in 6-8 phases. Every phases of this model is fairly addresses the both components to get the optimum success results.

SDF is consisting of two components:-

5.1. A life cycle model addresses the phases of a software project that included envisioning, planning, build, stabilize, deploy and post deploy activities. Life cycle model is used to provide structure to avoid the problems of the undisciplined and haphazard approach towards software.

4.1 Following diagram is self-explanatory of this flow:-

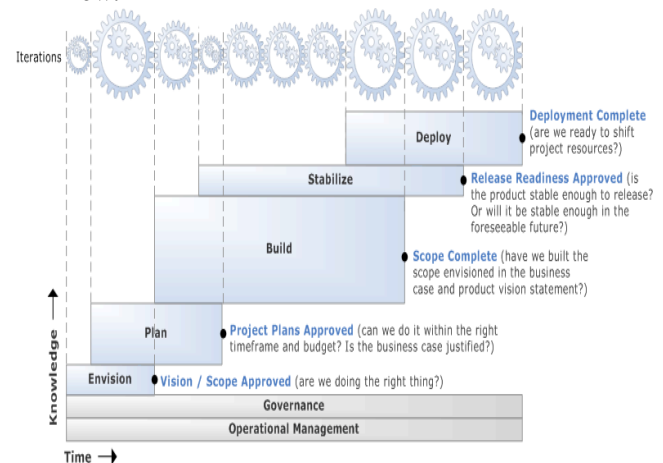


Figure 3: Contribution of Phases in QSDF

## 4.2 Risk Management Approach:



Figure 4: Risk Management Approach for QSDF

## 5. Quality Solution Delivery Framework Activities (Phases wise):

Code	Description
A	Accept - Necessary and desirable practice that is acceptable as written.
T	Tailor - Necessary and desirable practice that requires some adjustment to be used in environment.
O	Optional - Practice may be useful for some, but not all, projects in this environment.

S No	Organizational Track / Work Stream / Activity	tpeccA	Optional/Tailor
<b>1.</b>	<b>Envisioning</b>		
1.1	Establish Project Process		
1.1.1	Select Project Process Template	A	
1.1.2	Tailor Project Process		T
1.1.3	Review Project Process	A	
1.1.4	Establish Measurement Plan	A	
1.1.5	Establish Project Data Management Plan	A	
1.1.6	Monitor Measurements and Process Assets	A	
1.2.1	Capture Product Vision		
1.2.2	Write Vision Document	A	
1.2.3	Define Personas	A	
1.2.4	Develop Lifestyle Snapshot	A	
1.2.5	Review Product Vision	A	
1.2.6	Risk Identification	A	
1.2.7	Risk Analysis	A	
1.2.7	Risk Impact\Severity\Probability\Exposure	A	T
<b>2</b>	<b>Planning</b>		
2.1	Baseline Configuration Management		
2.1.1	Review Configuration Management Guidelines	A	
2.1.2	Create Configuration Management Plan	A	T
2.1.3	Establish Configuration Management Access Control Policy	A	
2.1.4	Establish Configuration Management System and Records	A	
2.1.5	Risk configuration and mitigation Planning (Risk Response)	A	
2.2	Create a Scenario		
2.2.1	Brainstorm\Prioritize Scenarios	A	
2.2.2	Storyboard \ Validate Scenarios		O
2.2.3	Write User Acceptance Tests	A	
2.3	Create a Quality of Solution Requirement		
2.3.1	Brainstorm and Prioritize Quality of Solution Requirements	A	
2.3.2	Write and Validate Quality of Solution Requirements	A	
2.3.3	Write User Acceptance Test Cases	A	
2.4	Create Product Requirements		
2.4.1	Develop User Interface Flow Model		O
2.4.2	Develop prototype		O
2.4.3	Develop a Domain Model	A	
2.4.4	Develop Functional Requirements	A	
2.4.5	Define Interface Requirements	A	
2.4.6	Define Security Requirements		O
2.4.7	Define Operational Requirements	A	
2.4.8	Allocate Product Component Requirements	A	
2.4.9	Prioritize and validate Functionality	A	
2.5	Design: Create Solution Architect		
2.5.1	Create Alternative Application Partitioning Designs	A	
2.5.2	Design System Architecture and Deployment	A	

2.5.3	Select Architecture	A	
2.5	Plan Project		
2.5.1	Determine Risk Sources and Categories	A	
2.5.2	Define Risk Parameters	A	T
2.5.3	Define Risk Management Strategy	A	T
2.5.4	Define Risk Response Parameters	A	
2.5.5	Plan Project Resources and skill set	A	
2.5.6	Form Project Team	A	
2.5.7	Establish Project Team Charter		O
2.5.8	Define Project Roles and Responsibilities	A	
2.5.9	Define Project Life Cycle	A	
2.5.10	Define Project Communication Plan	A	T
2.5.11	Identify Project Stakeholders	A	
2.5.12	Define Budget and Schedule	A	
2.5.13	Plan Project Stakeholders Involvement	A	
2.5.14	Review Project Plan	A	
2.5.15	Obtain Project Commitments	A	
2.6	Create a Database Project		
2.6.1	Create Database Project	A	
2.6.2	Import Existing Database		O
2.6.3	Configure Build and Deploy Settings	A	
2.6.4	Modify Generated Scripts		O
2.6.5	Add Database Project to Source Control	A	
2.7	Establish Environments		
2.7.1	Plan Product Integration	A	
2.7.2	Establish Unit Test Environment	A	
2.7.3	Establish Integration Environment	A	
2.7.4	Verify Integration Environment	A	
2.7.5	Select Project Coding Guidelines	A	
<b>3</b>	<b>Build</b>		
3.1	Manage Change Requests		
3.1.1	Prioritize Change Requests	A	
	Execute Risk Monitoring Activities as Planned	A	
3.2	Plan an Iteration		
3.2.1	Iteration Analysis		O
3.2.2	Plan Knowledge and Skills	A	
3.2.3	Define Communication Plan	A	
3.2.4	Iteration Plan Review		O
3.2.5	Conduct Previous Iteration Retrospective		O
3.3	Test a Customer Requirement		
3.3.1	Select and Run Customer Requirement Tests	A	
3.3.2	Conduct Exploratory Testing		O
3.4	Fix a Bug		
3.4.1	Reproduce the Bug	A	
3.4.2	Find Source of Bug and Assign	A	
3.4.3	Write or Update a Unit Test for Bug Fix	A	
3.4.4	Perform Unit Test for Bug Fix	A	
3.4.5	Refactor Code for Bug Fix		O
3.4.6	Bug Code Review	A	
3.4.7	Integrate Bug Changes	A	
3.5	Implement a Development Task		
3.5.1	Write Code	A	
3.5.2	Perform Code Analysis	A	
3.5.3	Write or Update Unit Test	A	
3.5.4	Perform Unit Test	A	
3.5.5	Refactor Code		O
3.5.6	Prepare for Code Review	A	
3.5.7	Code Review	A	
3.5.8	Integrate Changes	A	
3.6	Analysis		
3.6.1	Design and Development Analysis	A	
3.6.2	User Experience Work Breakdown Analysis	A	
3.6.3	User Education Work Breakdown Analysis	A	
3.7	Verify a Product Requirement		

<b>4</b>	<b>Stabilize</b>		
4.1	Manage Change Requests		
4.1.1	Analyze & Review Change Request	A	
4.2	Risk Monitoring and Control		
4.2.1	Risk assessment review	A	
4.2.2	Evaluate Risk Mitigation Factors and improve	A	
4.3	Plan an Iteration		
4.3.1	Select Iteration Backlog	A	T
4.3.2	Plan Iteration Resources	A	
4.3.3	Form Iteration Team(s)		O
4.3.4	Define Iteration Roles and Responsibilities		O
4.3.5	Identify Iteration Stakeholders		O
4.3.6	Plan Iteration Stakeholder Involvement		O
4.3.7	Estimate Iteration	A	
4.3.8	Define Iteration Budget and Schedule	A	
4.3.9	Obtain Iteration Commitments	A	
4.4	Test a Customer Requirement		
4.4.1	Develop Customer Requirement Tests	A	
4.5	Fix a Bug		
4.5.1	Code Fix	A	
4.5.2	Prepare for Bug Code Review	A	
4.6	Build a Project		
4.6.1	Start a Build	A	
4.6.2	Verify a Build	A	
4.6.3	Fix a Build	A	
4.6.4	Accept a Build	A	
4.7	Close a Bug		
4.7.1	Verify and Bug	A	
4.8	Verify a Product Requirement		
4.8.1	Develop Tests	A	
4.8.2	Select and Run Tests	A	
4.8.3	Monitor and review of risk occurred	A	
<b>5</b>	<b>Deploy</b>		
5.1	Develop Documentation		T
5.1.1	Create Documentation Plan	A	
5.1.2	Write and Edit the Documentation	A	
5.1.3	Technical Review of Documentation	A	
5.1.4	Test Documentation	A	
5.1.5	Risk Logging and Categorization	A	
5.2	Release a Product		
5.2.1	Establish a User Acceptance Test Environment	A	
5.2.2	Establish Validation Guidelines	A	
5.2.3	Select Release Candidate	A	
5.2.4	Create Rollout and Deployment Plan	A	
5.2.5	User Acceptance Testing	A	
5.2.6	Analyze User Acceptance Test Results	A	
5.2.7	Accept product for Release	A	
5.2.8	Execute Rollout Plan	A	
5.2.9	Create Release Notes	A	
5.2.10	Package Product	A	
5.3	Governance		
	Perform Quality Assurance		
5.3.1	Audit	A	
5.3.2	Identify Inconsistencies	A	
5.3.3	Analyze Review Minutes	A	
5.3.4	Review Design and Code Guidelines	A	
5.3.5	Track Corrective Actions to Closure	A	
5.4	Miscellaneous Activities		
5.4.1	Track Milestone Review	A	
5.4.2	Conduct Project Retrospective	A	
<b>6</b>	<b>Post Deployment: Operational Management</b>		
6.1	Miscellaneous Activities		
6.1.1	Team Meeting(s)	A	
6.1.2	Collect and Store Data	A	
6.1.3	Manage Stakeholder Involvement	A	
6.1.4	Perform Risk Review and Derive Lesson Learned	A	

6.2	Manage Change Requests	
6.2.1	Create Change Request	A
6.2.2	Track Change Requests	A
6.3	Manage Issues	
6.3.1	Issue Log Review	A
6.3.2	Analyze Issue	A
6.3.3	Take Corrective Action	A
6.3.4	Monitor Corrective Action	A
6.3.5	Review Action and Verify Resolution	A
6.4	Manage Risk	
6.4.1	Identify Risks	A
6.4.2	Analyze Risks	A
6.4.3	Select Risks for Mitigation	A
6.4.5	Prioritize Risks for Mitigation	A
6.4.6	Plan Mitigation Action	A
6.4.7	Implement Mitigation Action	A
6.4.8	Monitor Risks	A
6.4.9	Implement Contingency Plan	A

## 6. Project Team Model:

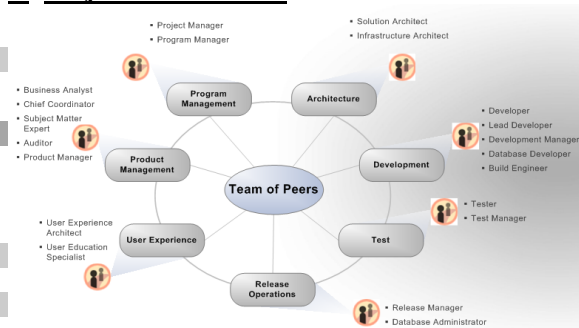


Figure 5: Team Management for QSDF

**7. Expected Risk Categories and Source:** This section presents a list of all the possible risks which may pose threat to a project at any stage. A Project Manager can refer to this section of document in order to identify the risks which are applicable to the project. Following are high level categories for which Risks have been identified:

- Product Size Risk:** PS is the Risk associated with the overall size of the software to be built or modified
- Business Impact Risks (BIR):** BIR are the risks associated with constraints imposed by the management or the marketplace
- Customer Related Risks (CRR) :** CRR are the risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner
- Process Risks:** PR are the risk associated with the degree to which the software process has been defined and is followed by the development organization
- Technology Risk :** TR are the risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system



6. **Development Environment Risks:** DE are the risks associated with the availability and quality of the tools to be used to build the product
7. **Risks Associated With Staff Size and Experience** associated with the human resources, their availability and their experience and performance requirements
8. **Risks Associated with Cost:** RAC describing the risks associated with the allocated cost of the project
9. **Risks Associated With Schedule :** Such risk associated with the schedule of the project

## **8. Lifecycle Models for Comparative Analysis with SDF**

*Table 1: Mapping of Market Driven SDLC and QSDF*

SDF Phases Mapped	RMA	Prototyping Model Phases	Waterfall Model Phases	Prototyping Model Phases
Envision	Risk Assessment	Requirements Definition/ Collection	Concept / Analysis	Plan
Planning	Risk Management Planning	Design	Requirements Specification	Risk Analysis
Build	Risk Response	Prototype Creation / Modification / Refinement	Design Testing and Implementation	Risk Analysis Engineering
Stabilize	Risk Monitoring Risk Logging and Categorization	Assessment	Coding and Development	Engineering
Deploy	Risk Review and Lesson Learned	System Implementation	Maintenance	Customer Evaluation

## **9. Mapping of Risk Assessment Approach and Landmark Development Life Cycle**

*Table 2: Mapping of Risk Management Approach and QSDF*

Phase	SDF	RMA
1	Envisioning	Risk Assessment
2	Planning	Risk Management Planning
3	Design	Risk Response
4	Build	Risk Monitoring
5	Stabilize and Deploy	Risk Logging and Categorization
6	Post Deployment maintenance	Risk Review and Lesson Learned

## **10. Conclusion and Recommendations**

Project and Risk Management are the traditional concepts presented by many project visionaries. Both concepts are critical to manage to overwhelm the chances of a project failure. Although these concepts

are considered as two different magnitudes and outfitted as separate entities usually. But keeping these concepts critical and considering both as collective prerequisite for project success, Quality Solution Delivery Framework is introduced for IT projects specifically. This framework is a blend of both concepts which ensures the Software development life cycle with parallel input exertions of Risk Management Approach. All planning, implementing, controlling and monitoring activities are taken care of both concepts simultaneously. Also this framework captivating the post-delivery activities to get the lesson learned and ensure the continual improvements.

The main role to perform both approach are recommended to do by a product owner, project manager or a Risk and Project Coordinator. Things are recommended to get done using checklists, Forms and pictorial view. Less\short set of documentation would be proven a smart approach to cope up with the timelines, cost and efforts constraints. But few of ISO 9001:2015 requirements for the compliance of Operations clause is kept as necessary to address.

Logging part of the product and Risk items is an important activity for this framework. This should be done in way that data can be helpful to obtain desired output to assist in decision making and lesson learned. Visual Studio Team Foundation Server features have been studied and analyzed to fulfil the requirements of QSDF. It is also analyzed that this product is currently utilized by IT companies with success ration and derive good performance results. Although it depends on the business size and nature that which tool could be suitable to fulfil their requirement.

SDF can be considered a smart approach to adopt and ensure the project success which obviously leading towards business growth. A project can be managed and run through the less time and effort input to drive the high level of output. It can be considered as an attractive way to get and retain customer and surely a business.

A software development entity can get benefit out of it by adopting the project management and risk assessment approach under single umbrella.