

Living Total Quality Management-(The Agile way)

Devyani Sharma, Saurabh Bagaria
Pitney Bowes Business Insight
Logix Techno Park, Tower - D
Sector-127, NOIDA- 201301(U.P.)
Phone 91-120-4026000
Fax 91-120-4026013/14
Devyani_sharma@mapinfo.com

Abstract:

Total Quality Management is used by organizations across the globe to ensure that every process from start to finish whether in product manufacturing, solution development or service distribution embraces quality as a guiding principle. In this paper we explore why TQM is such a powerful philosophy and how software development companies can benefit from TQM.

We shall discuss how TQM can help when following Agile Model and how it complements the agile philosophy.

The Paper will be divided in five sections:-

1. The Quality perspective
2. Agile Quality Management- a combined approach
3. Agile Quality Mindset
4. Challenges
5. Conclusion

Keywords:

TQM, Agile, Quality

1 Introduction

In mid 1940s, Dr. W. Edward Deming introduced the world with concept of TQM. Dr Deming's ideas had little impact on the American psychology but his management mantras were a huge success in Japan. As a result of which till date Japanese products are considered to of highest quality standards. Total Quality Management (TQM) is Intrinsic Quality Control, an integrative management philosophy aimed at continuously improving the performance of products, processes and services to achieve and exceed customer expectations every time.

T = Total = Everyone in the organization

Q= Quality = Customer satisfaction

M= management = people and processes

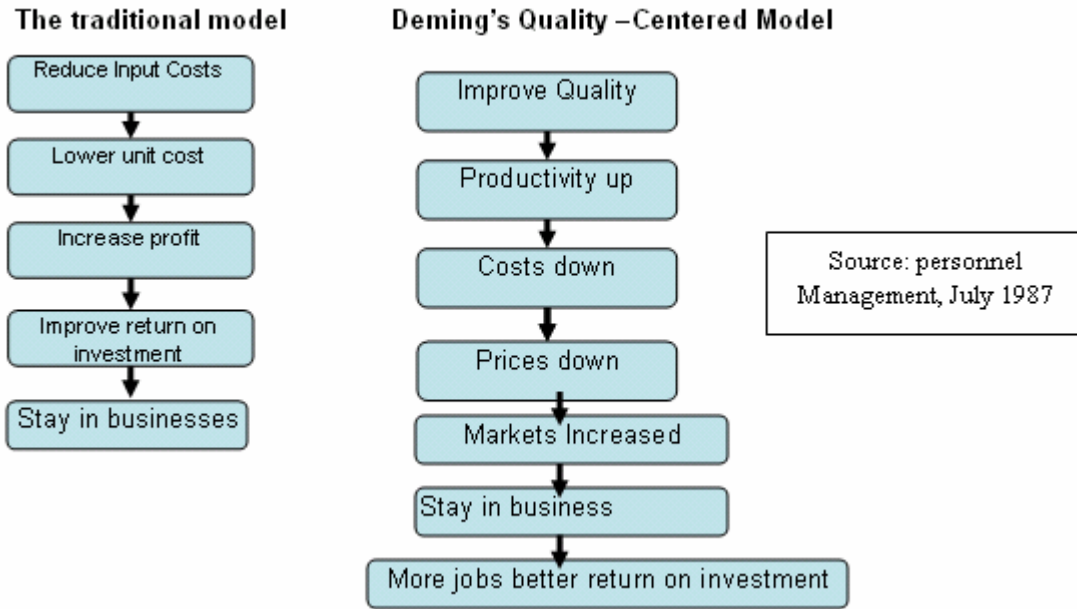


Figure:1

There are very few short term gains achieved by implementing TQM. The real benefits of TQM are realized in the long run and come into effect once TQM has become a standard at the workplace after it is running smoothly for sometime. In large organizations it may take several years before long term benefits are productivity, increased morale, reduced cost, and greater customer commitment.

TQM was originated in the manufacturing sector but it has been successfully adopted by almost every type of organization e.g. Churches, Highway maintenance, churches, Hotel management, schools and universities

Agile Methodologies:

On the other hand the agile methodology has made a huge success in the software industry. Agile software development is a conceptual framework for undertaking software engineering projects. In 1990s people began to realize that the traditional way of doing things had to be changed and that the multiplying market needs better software development methods, ones that would have following characteristics.

- More emphasis on Individuals and interactions than on processes and tools.
- Concentrate on the software rather than on documentation.
- A better relationship with customer and ensure their participation while software is being developed.
- Quicker response to change and easier way of implementing changes.

And this resulted in software methodologies that were quicker and easier to plan through changing business environment. One can find many different agile methodologies but they work on similar principles i.e.

- Software development is mainly design activity
- Individuals have a very big impact on the project
- Software development should not entirely rely on individuals rather it's the team which is responsible.
- Customers may not know the details of what they expect. Therefore they should have right to ask for change
- Change management process should be flexible

Although these methodologies had advantages there were a few problems uncovered as we progressed with them. There have been cases when the project started as small efforts and later went to become a very big project in such cases lack of extensive documentation made it difficult to take advantage of reuse opportunities. Also transitioning any new process or method into general use is a large and challenging task. Successful transition of any technology requires considerable resources, a long-term support program, and a measurement and analysis effort to gather and report results there is little such support available for the agile methodologies.

Total Quality Management in Agile - the combined approach

Studies have shown that about 50 percent of time is spent on avoidable rework rather than on value-added work, which is basically work that's done right the first time. Once a piece of software makes it into the field, the cost of fixing an error can be 100 times as high as it would have been during the development stage. Total Quality Management and Agile methodologies both work to solve these problems;

Total Quality Management way

- Building work processes that reduce errors to a minimum.
- Adding activities to find any leftover errors as early in the project life cycle as possible.

For instance, misunderstandings in deliverables and scope need to be resolved in the planning. Likewise, errors in business logic need to be caught in the analysis phase. The project manager should be aware that, in almost all cases, a good quality management process ends up taking more effort hours and up-front costs in the project. These costs are more than offset by a reduction in the time to correct errors in the testing phase, or worse, when the solution goes into production.

Agile Way

- Encourage communication
- Have more of customer involvement so that the risk of gap in customer expectations and actual product delivered is reduced.

Agile supports that most software development problems can be traced back to communication failures – someone didn't talk to the right person, or they talked to the right person at the wrong time, or they didn't talk to anyone at all. Here communication encouraged by using Special techniques. Very few projects fail because team was appraised about the status – rather when the information is hidden from the team that team fails to be aligned with the expectations. This means that everyone should be able to explain their point of view on any decision, to express their concerns, and to admit mistakes.

Comparison

Agile	TQM
Focus on Individuals & Interactions	Process is more important than
Involvement of Customer in all stages	Contract Negotiation before beginning with work
Fast responding to change for flexibility	Predefined plan is strictly followed through out the project
Quickly deliver a chunk of working software	Get the documentation approved first.

While TQM focuses heavily on processes and TQM way of doing things requires a lot of documentation the Agile way is quite contrast to it. Here the focus is on reducing documentation. TQM aims at reducing cost by appropriately documenting to minutest detail so that the scope of error is reduced. Agile aims to improving communication so that very little documentation is required and it aims at reducing cost by reducing time spent on documentation.

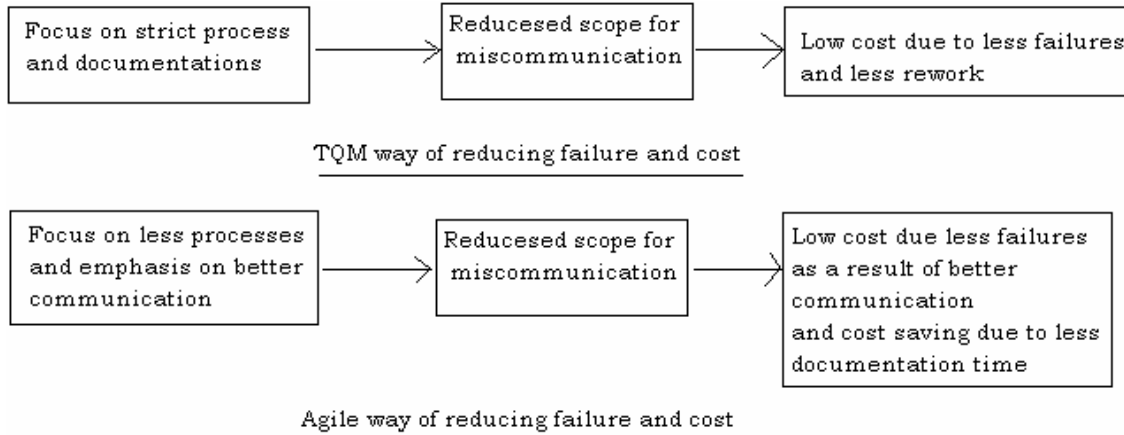


Figure 3

So we see that both techniques work towards on reducing the scope for miscommunication. There may be difference in the way they try to achieve this, but the intent is the same. The need of the hour is to find a path between the two methodologies so that the short comings of both are replaced by advantages of each other.

Need of the hour

We need a way where we do neither get ourselves over burdened with the overload of documentation nor should we forget to get bare essentials in place. Though documentation takes lot of time it is a one time investment which gives fruit in the long run and on the other hand over dose of it can lead to time wastage and project delays.

Business requires software to be

1. cost effective
2. Defect free
3. highly flexible(to adapt to ever changing business needs)

While TQM does answer to some extent to the Problem one and to a large extent to problem two, Agile has a solution to the problem one and problem three. Now if we want a solution that fits from all three angles to business perspective then we need to have a combined approach.

The approach that we suggest is a combined method in which Total quality is implemented in the small cycles of Agile. We call this approach TQM_(a) here we identify certain set of processes that are to be associated with project and we concentrate delivering the cost effective, defect free and flexible output using those processes. We need not concentrate on having process that span across the project rather, for each iteration we follow process most suitable.

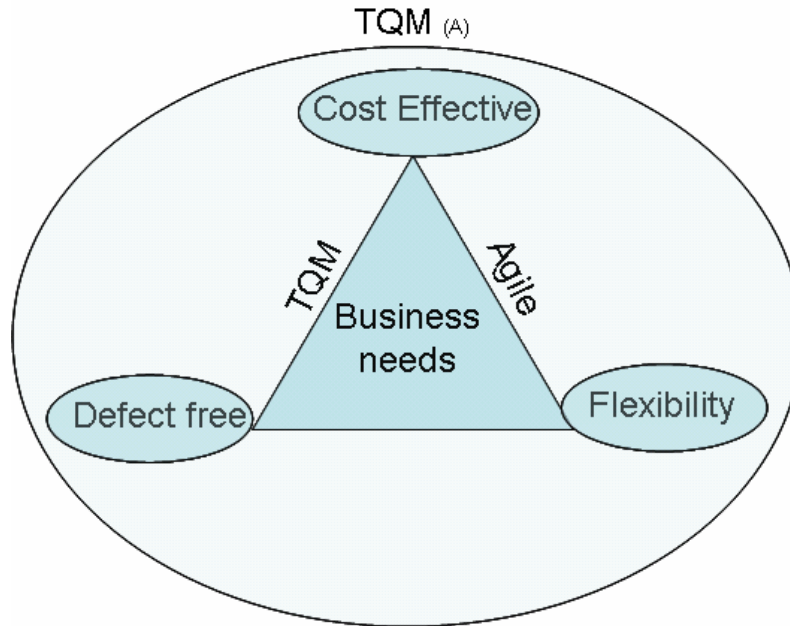


Figure 4

Example :-

We have a project which is estimated to have 50 modules so for this project If we implement it TQM way then there will be very heavy documentation phase that documents every minute detail and but if we go the agile way we would rather not do any documentation. But what if we do some basic documentation and we do have process control tools like

1. **Checklist** – to verify that nothing is being left out.
2. **Review** – to verify that whatever is being done is done in the right direction.
3. **Defect Reports** – to ensure that source for error is identified and eliminated.
4. **Entry Exit criteria defined for each cycle.**

Such process orientation may help us in handling the issues that were faced by following crash stories.

*You can also refer to the case study of CMMI and agile: a High Tech R&D Success Story by Niels Markert, ARD, Robyn Plouse, INTEL, Gene Miluk, SEI, for reference.

TQM(a) Mindset

“Quality is an attitude and not an event”

It's a great idea to perform regular quality control checks and have lot of documentation during a project. But they lead to a lot of extra cost and if one waits for checks to perform at the end of project or pushes them off onto the customer then definitely disaster can be seen coming.

Projects, view quality control as one or more events, or worse, something that is performed at the end of the project. To be effective, however, employees need to adopt an overall quality mind-set. Team members need to take ownership of their products and ensure that they are of the highest quality from the very beginning. If the team feels that it is becoming more of a burden for them then they may reduce the amount of documentation e.g. just make requirement docs but make them comprehensive enough that no other form of docs is needed for the team. The quality mindset says that even work you do should eliminate need of rework. Apparently lot of people are divulging from this. They tend to over document things creating

replicas of the same docs with little alterations. People should be open to critique and understand that the quality control process is in place so that the project results in quality deliverables that require a minimal amount of rework.

Many times we come across quality frameworks that talk of initiatives similar to ones mentioned below

1. Initiatives like quality day/ zero defect days to check results of everyone's work so far .
2. A dedicated team to thoroughly check the application when it is completed to find any errors that were missed up to that point.
3. Member of end user team (customers) to be responsible for the overall quality of the solution.

Question "What are your developers doing in between the quality days?"

Probable answer by the developers: - working hard to construct the solution, In fact, they may start thinking that quality is an event i.e. only on quality day/ zero defect days only they will pay attention to the quality of their work and on other days they may conveniently ignore it.

That could be cause of concern. Quality is a state of mind, and building quality into your solution needs to be an integral part of your process. In the framework mentioned above It appears that for people may consider quality control as scheduled series of events and the responsibility of a particular team

In fact, developers may produce lower quality work if they see the "quality day" as the time to catch all the errors. The developers might also rely on the inspection at the end of the project to catch errors, again putting the onus for quality on someone else. Finally, designating a customer to be in charge of overall quality appears to shift the responsibility as well.

Instead, the developers need to understand their responsibility for the quality of their work. At every step, It must be of high quality as they progress. The quality plan needs to reinforce this by having walkthroughs and testing inspections throughout the development process—not at arbitrary times in the project plan. Lastly, overall responsibility for quality must remain with the team that builds the solution, not with anybody else a These changes will ensure the team has a quality mind-set and takes responsibility for the quality of the application it is building.

Challenges

The goals and aims of TQM are worthy, the needs are certainly great, and numerous techniques and tools are available for implementing TQM. So why are quality, productivity, competitiveness, timeliness, and innovativeness not improving very much? Why are our companies experiencing the difficulties already noted-working on the wrong problems, becoming mired in irrelevant details, creating organizational chaos and low morale, overlooking creative options, making incremental or piecemeal changes at best, to list only a few?

Each organization has its own history, culture, record of performance, and leadership. Such a wide variation from one organization to another means that the assignment of setting up a Quality program is a unique problem that cannot be solved by force-fitting a "standard" model into the organization. Nor can a "standard" set of actions fit every organization

In any event, the implementation of TQM(a) will not involve significant, far-reaching changes in the organization. Rather it will cause a modification in people's attitudes, behaviors, actions, and habits and no one will be untouched. Responding to changes necessitated by TQM(a), people are likely to have reactions. The reason these thoughts and questions arise is that people become comfortable with a system.

Conclusion:-

While implementing TQM(a) management just does not need to investment heavily into process implementation. Rather the approach should be such that every member of the organization does every task with the aim of giving an error free output even if it means putting in little bit extra effort, or slowing the pace a bit. This higher quality, in turn, will lead to less rework and a more satisfied team and client. The basic value proposition for quality management is that you will save more cost and time over the life of your project than the cost and time required to set up and manage the quality management process. Some of the steps given below may be helpful in implementation but actually the TQM for each project will have a different approach to be followed. There is no standard method to be followed.

- Discussing the importance of quality with the team.
- Identifying simple metrics that can be collected early, with the hope that you can make one or may be two rounds of improvements before the work gets too far along.
- Ensuring that quality is not treated as an event and has become rather a way for life for the team.

References

Out of crisis

by W.Edward Demings

Total Quality- Success Through People

Ron Collard

Breakthrough Thinking in Total Quality Management

by Glen D. Hoffherr, John W. Moran and Gerald Nadler

International Journal of Quality & Reliability Management—Total quality management sustainability, Volume 19 Number 5 2002

by Mohamed Zairi (ed)

Perceptions of Success and Failure in TQM Initiatives

Kevin J. Dooley, Arizona State University

Do Organizations Consider Strategic Cost Management When They Outsource to China?

by [Tom Mochal](#)

Use quality management to catch errors early in the project life cycle

by [Tom Mochal](#)

Managing Project Quality

by Timothy J. Kloppenborg and Joseph A. Petrick

Small projects rely on quality activities instead of processes

by [Tom Mochal](#)

Quality management requires focus on processes, not just people

by [Tom Mochal](#)

Minimize rework as a part of your quality management process

by [Tom Mochal](#)

Quality is a mind-set, not an event by

[Tom Mochal](#)

Pin down the meaning of quality service before asking your team to provide it

by [Steven A. Watson](#)

'Value' replaces 'quality' in latest management strategy

by [gantthead](#)

Exploring the relationship between total quality management and information systems development

by Lillian Y. [Fok](#)

An Investigation of the Therac-25 Accidents by Nancy Leveson, University of Washington and Clark S. Turner, University of California, Irvine. AIRCRAFT ACCIDENT INVESTIGATION REPORT 96-5



Aircraft
Ministry of Transport [Japan]

Accident

Investigation

Commission

The top 4 things project managers do to destroy software quality

By Robert Bogue

<http://spectrum.ieee.org/sep05/1685/2>

<http://spectrum.ieee.org/sep05/1685/2>

Rod Welch **DIARY: March 28, 2000 TQM causes bumbling that dooms NASA's Mars program**

<http://www.dtic.mil/ndia/2005cmmi/thursday/miluk.pdf>

PureConferences

www.pureconferences.com

Pure Conference Solution Pvt Ltd., A-108B, Sector 58, NOIDA 201301, India; Tel: +91 120 4621080

Page 9 of 10

Authors biography and citations



Devyani Sharma

Devyani Sharma is CSTE, MCSD and MCA honors. Currently working as Quality Assurance Engineer with Pitney Bowes MapInfo, she has 3.5 years of experience working in the field of quality assurance. Prior to Piney Bowes MapInfo she has worked with Keane, Accenture Services Private Limited and Quark Systems Consulting. Her experience is mainly in Quality Assurance and Software Development In the year 2007 she was a speaker at the International Software Testing Conference, Bangalore and in the year 2006 she had submitted white papers on Importance of Test Data and Testing best practices to Accenture services private limited and a Proof of concept on automation using Rational functional tester. She has received JSE Super achiever award for performance in 2005-2006 by Accenture Services Private Limited.

Citations:

Importance of Test Data,2006, submitted to Accenture services Private limited.
Non IT professionals in Software Testing, 2007 Software testing Conference
Location based service, 2008GIS development.
<http://www.gisdevelopment.net/magazine/global/2008/june/44.htm>

Saurabh Bagaria

Over Ten years of experience in designing and developing solutions on varied platforms with exposure to all stages of project development life cycle. Specialization in creating technical architectures for solutions ranging from web based, desktop to embedded applications on leading technologies such as .Net, UML, Design Patterns, XML, LBS and SOA for various verticals. Expertise in Geographical Information Systems, Location Based Services based on ESRI, MapInfo, and Oracle GIS Technologies.

Currently a member of management team, help defining the business vision, building teams, people management, PMO management and lead innovation efforts

Citations:

A geographic information system to study trauma epidemiology in India
<http://www.traumamanagement.org/content/1/1/3>
(In list of Top 10 most accessed articles of all time)

“GramChitra - Taking GIS to Grassroots”

http://thinkcycle.org/tc-fileSystem/download/development_by_design_2002/publication:_gramchitra_-_taking_gis_to_grassroots/dyd02_GramChitra_latest11.doc.pdf

Location Based Services: A collaborative approach

<http://www.gisdevelopment.net/magazine/global/2008/june/44.htm>