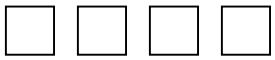
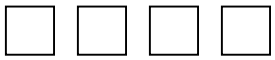


Software Quality Improvement for Financial Services after the Merger

A Practical Guide to Achieving Short Term Returns and Long Term Organizational Gains

An MKS White Paper
By Linda Newsom-Ray
Senior Consultant
MKS Professional Services



Objective

To provide senior IT managers and executives in financial organizations with insights into implementing quality and best practice initiatives, particularly when coping with the aftermath of mergers or acquisitions.

It will help managers and executives to:

- Understand some common issues created when IT-dependent organizations merge
- Effectively assess the role of large-scale quality and benchmarking initiatives, such as Six Sigma[i], TQM[ii] and CMM[iii] in the changing organization
- Consider some simple ways of implementing quality initiatives at a pragmatic level, which may provide a faster return on investment.

Change as the only constant

Mergers & acquisitions

The Financial services sector is, arguably, more prone to mergers and acquisitions than most other industries. One can hardly think of any financial institution of any standing that was not formerly two or more separate organizations in the not-so-distant past.

With increasing market pressures on growing the business to attract market recognition and retain market share, expansion by merger has become the most common method of fighting off the competition. If you can't beat them, buy them.

Against this backdrop of expansion and the resulting organizational change, there is the unalterable fact that all financial organizations are 100% dependent on their IT systems in order to stay in business.

Given these two factors, it follows that such organizations, and particularly their IT departments, must be fully equipped and ready for the next sweeping change. This is not easy when, commonly, financial organizations are littered with elderly 'legacy' systems that are, in fact, still critical to the business.

Mergers inevitably create duplication of some underpinning customer-facing technology. It can take years to migrate such systems or to construct a viable solution for the new, enlarged organization. Where there are especially entrenched positions and complex systems, it can take years just to obtain agreement to the decisions among the stakeholders.

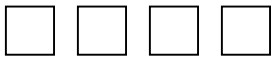
Moving towards more direct customer-access systems

In addition to the complexities caused by mergers and acquisitions, there is an inexorable movement towards 24/7 financial services and customer-intervention banking.

> *'Data from eMarketer indicates that that the number of online banking consumers in the U.S. will steadily grow from 2.8 million in 2000 to 18.3 million by 2004.'*[iv]

> *'Two thirds of online consumers believe that, five years from now, they will conduct most of their financial transactions on the Internet."*[v]

While this is a tremendous opportunity for acquiring new business and potentially lower servicing costs, there are also severe risks.



It is currently not feasible for banking and other financial systems to have long periods of downtime without adversely affecting the company's image. It has never been acceptable for accounts to be mismanaged, but today's technology makes any errors more obvious, more quickly. The resulting adverse publicity can seriously dent market confidence; customers voting with their feet can make a huge impact on the company's bottom line.

Quality and service – the sole differentiators

With regulatory limits on what can be offered to whom and market place factors driving narrower margins, it is increasingly difficult to be an extraordinary provider of financial services. Long gone are the days of unswerving customer loyalty. Customers are becoming more and more adept at swapping providers to obtain the best return on their savings products and pay the least amount on their borrowings.

With less and less tolerance for any downtime or errors, customer loyalty may only be as strong as the next click away from your website to a better one. So what are financial organizations doing to improve customer retention?

It is certainly important to recognize that the quality of service does make a difference. To this end, many financial organizations have implemented positive policies regarding their customer-facing processes and staff. With 24/7 banking, however, it is the systems that are directly customer facing: thus, the responsibility for excellent service is no longer the sole province of the branch network or customer service staff. So it follows that the systems development, management and operational staff all need to be wholly aware of what constitutes providing quality service.

Increasingly, financial organizations are turning to major quality initiatives such as Six Sigma and TQM to provide some answers. These are designed to sweep away bad old practices and to bring in fresh new ways of ensuring that the customer's interests always come first. Ever more financial organizations are focusing their IT improvement initiatives, by utilizing well-respected benchmarking models, such as the Software CMM and, more recently, CMM Integrated. There are certainly some major success stories with all of these models, including claims of billions of dollars saved, enormous returns on investment and huge improvements in product quality.

Some of the obvious success stories are:

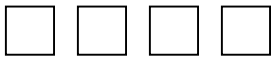
- Six Sigma: Motorola[vi] and General Electric[vii],
- TQM/Malcolm Baldridge Award and Software CMM: Boeing[viii]; and
- Software CMM: Raytheon[ix]

What can be achieved by implementing Quality initiatives?

Six Sigma, CMM and TQM are just some of the many different flavors of best practice, all of them stemming from the pioneering work of Dr Walter A Shewart, Dr W Edwards Deming and Dr Joseph M Juran in the 1940s, 1950s and 1960s.

All of the models promote the concept of reducing errors by analyzing the processes and systematically eradicating the causes of injected errors through various useful and effective techniques.

The results achieved by other high profile companies provide a tremendously compelling reason for wanting to implement these models in your own organization. There are many good



practices, which are offered up as a way of life, as part of these quality initiatives – and good reasons for wanting to adopt them.

Successful quality initiatives:

- Can deliver huge improvements in the quality of the software; greatly reduce costs of software maintenance; help the IT department to deliver software better, faster and cheaper.
- Enable management to have greater visibility into the software processes, so that there are fewer surprises e.g. less likelihood of late delivery, cost-overruns, unstable systems/releases.
- Offer better options for making informed choices when business priorities change.

Of course, practically every financial organization would choose to go this route if success could be guaranteed. The fact is, however, that although large-scale quality initiatives have many good qualities, longevity is not normally one of them.

It is quite difficult to find companies that will publicly admit to it, but it is generally accepted within the software process improvements industry that at least 70% of large-scale quality initiatives fail, normally within two years, having delivered an average of 5% of the anticipated ROI. The Juran Institute estimates that 80% of all companies that tackled TQM in the 1980s failed.[x] More recently (1999) a DACS report demonstrates similar findings.[xi]

Why do large-scale quality initiatives fail?

A number of reasons have emerged as to why these initiatives fail so frequently:

- **Organizational flux**
  Six Sigma and TQM both involve training the whole organization from top to bottom; both business and IT staff. There is normally little return on investment within 2 years of starting these initiatives because of the high up-front costs. Where the organization has just trained its entire staff and then it merges with another organization that has no such background, the investment will likely be lost. There is no guarantee that the new management will have the same belief in the quality initiatives. Major organizational changes will wash away the knowledge and create high staff turnover when it is least desired. Even when no mergers occur in this period, senior management attention is frequently drawn elsewhere and funding for the quality initiative is drastically reduced or completely removed.

- **Resistance to change**
  Six Sigma and TQM both demand exceptional drive and long-term commitment (not just funding) from management in order to implement them successfully. They also demand total acceptance by the whole workforce, including the CEO and all senior management, to the extent that they change their working practices forever and not just for the first two weeks after they have been trained.

  The majority of the failures are caused by the human factors not being taken into account. There should be no surprises in this:

  "Culture is nothing more than the values, beliefs, and norms of a group of people. There is nothing more difficult to take in hand, more perilous to conduct, or more

uncertain in its success than to take the lead in the introduction of a new order of things."

Niccolo Machiavelli, The Prince

Mergers create enormous levels of organizational change and can result in high stress levels among those staff that do not feel in control of how things will shape up in the new organization. Bear in mind that some people will suffer from 'change fatigue'; i.e. they are unable to cope with taking on board fresh concepts for implementing quality improvements, whilst they are having to change their whole thinking on what their role is (or is not) going to be. Of course they will try to hold on to the past – it is the only thing they have. These people need to grieve for the death of the old organization – and they will need help to overcome the loss, by constant communication of how the future will be, and by being involved in making it happen. Acceptance of change needs to happen in the heart and soul, not just in the head.

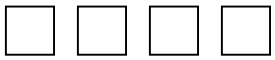
There are many, many books on the subject of introducing change and overcoming resistance: some of them useful.[xii] The bottom line is that people will not change their behavior unless they see – and are emotionally drawn to – a good reason for doing so – the 'What's in it for me?' syndrome. Moreover, where people can see that the top of the organization is not providing good role models, why should the rest of the organization behave like good citizens? A fish rots from the head down.[xiii]

- **Defining the solution before the requirements are agreed**
  If someone in the organization thinks that Six Sigma is the solution, they should seriously consider what set of problems they intend it to address before moving forward. Buying a friend a stretch limousine may seem a wonderful gift until they need to park it at the mall. There is no such thing as a 'one size fits all' in terms of quality initiatives. For every success story like Motorola or Boeing there are at least two expensive failures.

  One should carefully examine the type of companies that achieve success with their quality initiatives. Often, there is a particular need for precision software engineering, e.g. for maximum safety within space shuttles. Typically, these organizations have robust software engineering practices that have taken years to develop and their project delivery cycles are stable and rather lengthy. Their staff have specialized domain knowledge and tend to be very loyal and long serving. All of these factors point to greater stability and less churn, which means that quality initiatives can thrive.

  These are not the characteristics of most financial organizations today. Pulling out all the stops to develop the systems behind new web-based products, for a marketing campaign that will ensure that maximum capacity is needed immediately, is a very different business proposition from launching a space shuttle.

  The financial services processes need to be fast and flexible, and capable of delivering good quality. So if implementing Six Sigma isn't going to help for another two years – even if it is successful – what will?



Getting started with an incremental quality initiative

Simple steps to help make it work

At this point, it should be noted that managing development projects is quite difficult. Managing software process improvement projects is even more difficult, even under normal organizational conditions, because of the nebulous concepts involved and a general lack of ability to either estimate accurately or to identify scope creep.

So, trying to get any kind of quality initiative to 'stick' when the organization is undergoing massive change, with all the resulting uncertainty, is next to impossible. The only way of tackling it is through the KISS principle (Keep It Simple, Stupid).

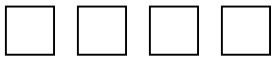
It is perfectly possible to use the principles of the quality models to provide an objective basis for improvement without investing hugely in training everyone at once. Models such as the Software CMM, or the newer CMM Integrated are excellent as an objective yardstick, for measuring organizational process maturity. Using these models can be very helpful in terms of prioritizing where improvements should start.

It is important, however, to make sure that goals are set for the relatively short-term and have SMART[xiv] objectives that can demonstrably be achieved.

The first thing to do when faced with massive organizational change is to define the parameters of what is within the scope of the project and what is outside – in other words, the who, what and where of the people, processes and systems. The quicker this can be captured the quicker the duplication and waste can be eradicated. In any merger situation there will be redundant systems, redundant processes – that is, ones that are not facilitating a quality result – and, less pleasant to deal with, redundant jobs.

Next: define what is hurting. In the early days of a merger this is tricky as practically everything will feel as though it's hurting. Normally, it is a good idea to start by asking if there is a tendency for projects to overrun on dates and costs and under-perform on quality and/or delivered functionality. Looking critically at the cost of maintenance for each system is also a good indicator of quality (e.g. well-structured systems) and whether project and development practices are good or poor.

Once the causes of the pain are found, proper steps can be planned to define the desired values and how they will be achieved. Having an established quality improvement initiative demonstrates that there is a plan for the future and it will have the effect of providing some stability in what can be uncertain and, for some people, dispiriting times.



Early days

Living through a merger is like being caught up in a moving time-zone picture: there is a patch of sunshine – although one has to keep running to stay in it – and the rest of the world is in darkness. The future is unknown and the past is gone. During this period of uncertainty, people will gather around the coffee machine or water cooler and chatter, incessantly, about the situation. Be aware that 'the grapevine' is not only a source of misinformation and the subsequent generation of fear, but it will travel much faster than any formal organizational information and communication flows. It is therefore extremely important to put in place a policy of continuous communication of as much information as can be given at every stage. Even a statement of 'we do not yet have the full picture' will at least help to reassure staff that they are not being plotted against by bosses who know what the future will hold, but are not telling them.

Setting up a specific web site, dealing with the communication of plans and allowing for Q&A, can be very cost-effective and can be changed quickly as the plans develop.

In the early stages, it will be apparent to all staff that there is a plethora of systems providing more or less the same functionality: so some systems will have to be scrapped. Uncertainties hanging over the technical future add to the loss of morale because systems that are now declared surplus were once built and proudly maintained by staff whose historical contribution to the organization is being visibly eradicated.

Productivity can be close to zero, so it is important to put something positive in place quickly. This will normally mean putting together some sort of blueprint for the future and making sure that the resource planning allows for acquisition of new skills. For this to happen, a clear technical strategy needs to be in place based upon knowledge of the perceived needs of the enlarged organization. This will take time and require input from a wide cross-functional group. Anything that builds bridges across 'the divide' can only be useful at this stage to avoid some of the negative feelings that can occur.

Decide on what 'success' means for the organization – and measure it

To give a relatively quick and easy route out of the maelstrom created by mergers, it is recommended to hold one or more Goal / Question / Metric workshop(s), or some similar mechanism that will enable the group to determine, for example:

- How the organization should be prepared for the future
- What business needs must be met
- What measures must be in place to improve:
  - time to market
  - code quality
  - system flexibility, etc.
- How to put together rapid response teams that are willing to work cross-functionally with the rest of the business
- How to reduce maintenance costs

It should be self-evident that the less effort spent on finding and removing bugs, the more money and resources will be available to position the organization for the future. In the new scheme of things, the 'techies' will need skill-sets that include analysis, architecture design and communication skills for Joint or Rapid Application Development, as well as technical ability and domain knowledge. Putting together a simple skills matrix, i.e. the required skills versus who has them, can be a helpful way of spotting the gaps.

Be very careful to avoid implementing a full-scale metrics program at this stage: they are notoriously difficult to make successful, even in with 'normal' levels of organizational change. Focus only on collecting metrics that are well-understood; that measure processes, not people; and when you know exactly what you intend to use them for. Collecting metrics for the sake of it has no benefits whatever and will create resentment.

What you need to make the right things happen

Looking to the future also involves analyzing the present and learning from the past. What is needed here is:

- The right vision and guidance
- Clearly defined business objectives and priorities
- Clearly defined technical strategy
- Clearly defined quality strategies
- The will and the resources to make it succeed
- The whole-hearted participation of the people who plan to stay in the company

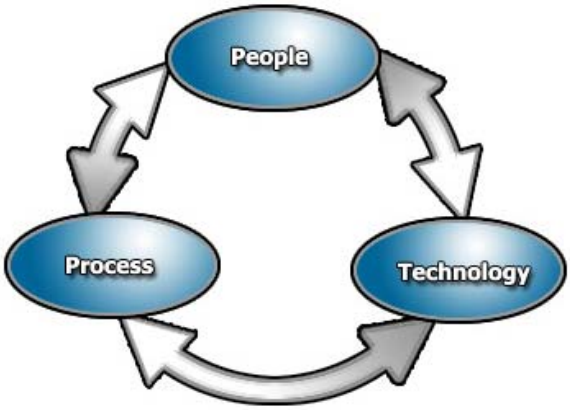
Design action plans that prioritize and focus on results – and track, track, track. The Deming cycle of 'Plan, Do, Check, Act' is a useful aide-memoire and has now been incorporated into another well-respected benchmarking standard, ISO 9001: 2000.[xv] Ensure that as much as possible of this planning work is co-operative, using team-based activities. Using workshops across the enlarged organization to define and build towards common goals will reduce the tendency for attitudes of 'them and us'.

People, process and technology

'The underlying premise of software process management is that the quality of a software product is largely determined by the quality of the process used to develop and maintain it.'[xvi]

It is well recognized that neither technology nor people alone will make high quality software systems. Well-defined and consistently applied supporting processes must also be present.

This interrelationship is normally represented thus:



Diag: *The people, process and technology triangle where the process and the underpinning technology support the people developing the software.*

Each of the three factors is discussed below with particular emphasis on implementing quality initiatives against a background of constant organizational change.

People

Dealing with uncertainty

If all that is offered in the new organization is uncertainty, people will leave - usually the best people leave first. With any merger situation, or indeed any major organizational upheaval, people fear downsizing or losing out on job-leveling (i.e., ironing out the inconsistencies in rewards for similar roles). Suspicion and resentment can be rife and territorial battles flare up with little provocation. The 'not invented here' attitude can be a killer to any initiative. This is the base level of Maslow's Hierarchy of Needs[xvii] where survival is key.

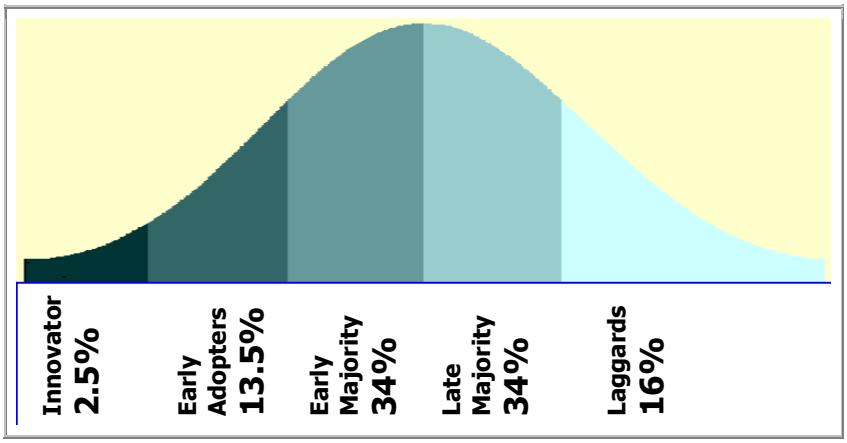
It is essential to get people to believe that they have a future in order to progress from this stage. People cannot think at the 'self-actualization' level if they live in fear of not being able to support their family.

'Take hope from the heart of man, and you make him a beast of prey.'

Ouida

Dealing with resistance

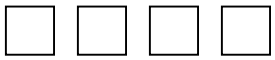
It is likely that there will be people whose skills are no longer required and who are unable or unwilling to move on once their usefulness has been exhausted. Such people can be destructive to any improvement or change management initiative: they are the 'laggards' that Dr Everett M Rogers[xviii] identifies in his Adopter Categories. Sometimes the only way of dealing with them is to move them out of the organization.



Diag: *The 'Adopter Categories', Dr Everett M Rogers*

To overcome resistance, it is vital to enlist the help of people who view change in a positive light. The best-suited candidates for supporting quality initiatives and making things happen are at the Innovator and Early Adopter end of the bell curve. These people need to be nurtured, as they are the ones who can effect the required changes in organizational attitudes and aspirations.

It is only possible to make an organization extraordinary by capitalizing on their extraordinary people. Extraordinary organizations will succeed where others fail. The average never wins:



- *'To succeed we have to surprise people. We have to attract and addict them. Attention is all.'*[xix]

There is a checklist of 'Principles of Managing Organizational Change' in Appendix A, which provides further guidance.

Technology

Setting business strategy

The technology that underpins today's customer-facing 24/7 financial services may still have some roots in its legacy systems. These are usually many years old, written in 3rd generation languages, rather inflexible to change and require highly specialized staff to maintain the code.

When mergers happen, the new and enlarged organization may find itself with two or more systems essentially addressing the same business requirements. A way forward must be found, whether this is migration, re-engineering, re-writing, outsourcing or some combination of these.

Defining and gaining agreement to that plan can be an uphill struggle. Decisions should always be based on a rational analysis of the ability to meet the business needs in the future and the capacity, flexibility, and potential for longevity that each of the systems provides. Any decisions that are based on feelings, partial information or supposition will quickly fuel suspicion and claims of favoritism among aggrieved staff. One such mechanism for analyzing and defining the best way forward is Constructive Evaluation[xx]. Using the principles of Quality By Design, the process involves identifying critical business goals, then gradually and iteratively drilling down to define what needs to be in place to meet the defined goals. In the hands of an experienced practitioner there can be no doubts regarding 'fair play'.

Similarly, some best guess must be made as to the main business functions the technology must support now and in the future. Assuming a range of up to 3 years is reasonable. Shorter periods run the risk of purely tactical moves that damage the overall systems architecture, while longer periods are increasingly difficult to predict.

Strategy needs to be based on current knowledge and the most likely scenarios. It is probably a good assumption that the next two years will consist of almost constant change and that there will be another significant change after that. It is good practice to assess the most likely directions and then put in contingency plans and agreed checkpoints to mitigate the risk of potentially driving everyone down the wrong route. Whatever else happens, the need to have flexible systems (and flexible staff and processes) will become ever more of a driving factor.

Adding value over a period of time

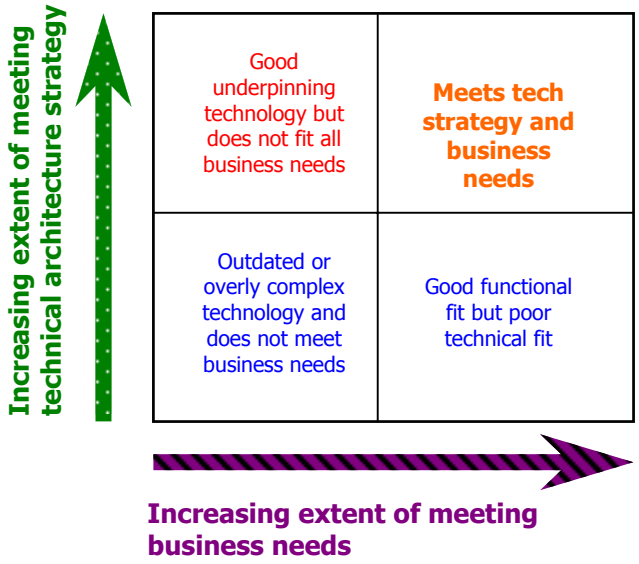
Given the huge investment in most systems that support financial organizations, it is normally not viable to start completely from fresh with critical systems. If the old systems are still supporting at least some of the critical business functionality then the code has to be re-used or recovered in some way. It is, therefore, essential to discover what systems are in place. An audit must be performed to determine what systems are where, what applications are running on each system, and what hardware, operating systems, databases, and languages are being supported.

To move forward from here one must apply a policy of continuously removing or transitioning the systems that:

- Are a positive risk to the business (zap)
- No longer support the business's future (scrap)
- Need a GUI front end to give the look and feel that is wanted (wrap)
- Can, if transitioned to better technology, provide useful data/services (tap).

Once the physical audit has revealed what physical systems are present, the next step is to perform a physical to functional mapping, to determine to what extent the systems are supporting the required future business goals.

Mapping the systems against the following simple matrix can help spot obvious candidates for zapping, scrapping, wrapping or tapping.

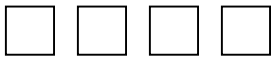


Diag: *'Value to business' matrix; aim to zap or scrap the systems in lower left quadrant; consider wrap or tap for lower right quadrant. Systems in top left quadrant may, dependent on scalability, be candidates for further development, to enhance business fit.*

It is also important to recognize the current levels of quality, including maintainability as measured by cyclomatic complexity, visibility of structure, etc. Unsatisfactory ratings against these measures could have a bearing on the need to replace sooner rather than later. Extensive examination of the systems against objective acceptance criteria, known as the 'ilities', (as in portability, flexibility, security, capacity, etc.) may also prove beneficial in determining the need to replace early.[xxi]

Define detailed mapping of existing/near future systems, against the blueprint

- The 'ilities' can help describe the increased requirements of the systems as they may be used in the enlarged organization or in a more customer focused way. Consider Constructive Evaluation Methodology[xxii] for this detailed work.



- It is essential to define the future as clearly as one can even if it is prone to change. If it is documented and well thought out, it will form a solid basis for determining the risks involved in moving both towards and away from the blueprint when circumstances change.[xxiii]

There is a checklist of 'Defining Technology Futures' in Appendix B.

Process

Coping with differing process maturity across different sites

The third corner of the triangle is process. There is no possibility of building truly great systems, except by accident, unless the people who are designing and engineering them are working to well-defined, agreed processes.

With mergers and massive changes in financial organizations, the likelihood is not that there are no development and project management standards, but rather too many disparate ones. There may also be different IDEs across various sites, which add to the complexity of developing a single project with multiple methods.

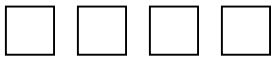
In addition to this, the speed of changeover between developing the old legacy systems and suddenly having to put together brand new systems with collaborative development techniques may mean that there are no defined processes for constructing the newer types of systems. This can mean that trouble is building up for later when these newer systems themselves advance to legacy status. Analyzing the code and re-engineering it to a cleaner design may be required to make these systems maintainable and capable of being deployed to meet the needs of the expanded organization.

As with technology and people, the only way through this is to find out what methods and processes are in use across the organization. Some analysis is needed to determine the level of use, the maturity of the processes, and the degree of success as evidenced by the visibility of the project process and the quality of the resulting software products. Only when this health check across the organization is done can any decisions be made as to the next steps. Suspicion and competition can be rife between previously separate organizational areas. It is, therefore, essential that only objective measurements be used to determine the relative maturity and value of the processes in use.

Using an objective yardstick, like the CMM, can prove beneficial in such cases, as there can be no arguments if one part of the organization has already been assessed as a Level 3 and another part has no formal project management or software engineering processes in place.

If none of the component parts of the newly enlarged organization have a history of using CMM as a benchmark, then it is probably not a good idea to start full-scale assessments at this time. Full-scale assessments can be a major investment and the organizational flux experienced post-merger will not form a good basis for the training and process development that will be required.

The most cost-effective approach to understanding the process maturity at each site is to organize a series of lightweight process reviews. This will avoid over spending on too-sophisticated mechanisms. With a knowledgeable software CMM practitioner, for example, a good 'feel' for the maturity of a development site can be obtained in a low number of hours, given some astute questioning and access to project and process documentation. Once this understanding has been obtained and documented, then the CMM can be used as a mechanism to prioritize the areas for potential improvement.



Financial organizations, in common with other software-dependent industries, frequently fall into the trap of aiming for a particular CMM level as their main goal. Whilst this is admirable in itself, it should not be the only aim: any improvement should be based on real business goals, not on some arbitrary notion of 'goodness'. It should be noted that CMM initiatives are also highly prone to failure: the main reason for failure – apart from the normal factors to do with resistance to change – is a lack of focus on meaningful business objectives.

Major CMM-based initiatives should not be attempted unless it is known what measurable (SMART) results are required to be achieved: for example – fewer defects into production, better predictability of project timeliness and delivery of functionality.

Aiming purely for a particular level means that too much emphasis is placed on the assessment itself, rather than improving the way things are done: once the assessment is over, the staff may just heave a sigh of relief and heave the process documentation out at the same time. If the right business goals are set, however, then the CMM level will be attained as a useful by-product – and the staff and management will want to continue working in such as way as to continue to reap the business benefits.

Should an attempt be made to bind together all the development groups? Or is it best to allow the differences to co-exist?

There will almost certainly be some differences in methods of project management and software engineering methods used across previously disparate organizations.

Performing an initial review of the processes to discover the differences and then mapping some common terms for each site can be a useful first step, as a basis to understand how each site operates. Allowing differences in processes to allow for different styles of project, is something to be encouraged – provided that there are consistent and visible control points for projects, which can be understood and used when developers and project managers from different sites work on the same team. The results of the initial process reviews can form the basis for defining the terminology and processes to be used on the projects that follow.

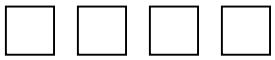
Different development units will probably employ different IDEs. This is acceptable, provided that the quality processes are recognizable and they are being used to ensure that the right functionality is being delivered to the anticipated level of quality.

The most benefit can be obtained by ensuring that source code re-use is possible. To do this requires full visibility across the IT organization via consistent enabling technology. The most obvious candidates for supporting technology would be for the critical processes such as: requirements management; workflow through the project with quality reviews at each stage of deliverable; full software configuration management throughout the development and testing phases; fault-fixing; and deployment. Rewarding developers every time they re-use code can also be an efficient and cost-effective way of reducing maintenance overhead in the longer term.

Judging by metrics

Cost

Any attempt to understand the relative cost of development and maintenance, versus the quantity of deliverable functionality, may be regarded with some suspicion, and thus fraught with problems. On the other hand, it will not be easy to determine what is to be done with a particular development site, until one understands it in objective terms.



One needs to determine, in bald terms, what it costs to develop systems in this site, as opposed to that site. Any comparison is only meaningful if the size (including complexity) of the development is taken into account. Descriptions of effort or duration are not verifiable as a measure of size.

Installing Function Point or Feature Point analysis is likely to be too time-consuming and expensive to be of benefit in the shorter term. Instead, consider a simple mechanism, using self-defined size/complexity attributes, which can be easily trained and used. The same mechanism can be used for estimating all project size metrics, which will also be of benefit in the longer term.

Right quality products and projects

The following are some easy ways of finding out whether the projects are delivering what is wanted:

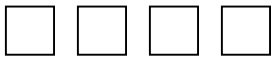
- Judge by the quality of the resulting code; number of defects per KLOC, measured at intervals: e.g. on implementation, and at 6 months, 12 months, 18 months
- Judge by project timeliness, and whether the project is on-budget
- Examine those teams that are consistently delivering late and/or over-budget – and find out why this happens
- Examine those who are consistently good – and find out why: reward and promote their achievements

Define the goals for process quality and process adherence

Simple mechanisms, like defining a set of artifacts to be used in each project, can make a big difference to the quality of software products. If there are no consistent artifacts across the organization, consider implementing some or all of these: quality plans, risk management plans, configuration management plans, test strategy plans, standardized progress reports, standardized time-sheets, requirements change request forms, quality assurance and customer sign-off forms for deliverable work-products or documents, and post implementation reviews. Consider implementing formal peer reviews of critical work-products, as they can be a cost-effective way of reducing defects early in the lifecycle. Even rapid delivery projects can benefit from extremely concise versions of all these control-mechanisms.

Alternatively, consider implementing a process management and workflow system to ensure that each work product (i.e., software or management artifact) conforms with the required flow for that type of project. For example, a workflow rule can be put in place to make sure that Quality Assurance reviews a risk management plan before it goes out to a project sponsor.

Process improvements and quality initiatives in general can generate enormous rewards both in tangible forms (such as a reduction in maintenance costs) and in intangible ways (staff satisfaction). The only way to reap the benefits, however, is to make sure that the improvements are sustainable in the long term. This requires planning, persistence and a proactive interest in the processes to make sure that the improvements become – and remain – a way of life, or 'the way we do things around here'.



How MKS solutions support quality initiatives

For organizations serious about software quality and process improvement, MKS's Integrity Solution provides a range of software configuration management technologies enabling companies to manage change to software code, systems, and their development organizations more effectively & with high return on investment. Here are just a few of the ways the MKS Integrity Solution can support an organization's software quality improvement process:

Code comprehension – a mapping of relationships and patterns to better comprehend and identify a large or complex code base developed by many hands and over many years.

Impact analysis – detailed analysis of impact of change prior to its execution to prevent downstream breakages or failures

Process assessment – professional services staff can streamline process improvement with an assessment of current software development activities, documentation & tool automation of working processes, and recommendations for incremental and manageable process improvement

Graphical process mapping – visual mapping of the software development process within the workflow tool, enabling teams to clearly document process and automate it via technology

Issue escalation – for priority or urgent issues

Adjustable role-based workflow and security – process, workflow and security is established based on the hierarchical organization and dynamics of the team. Can be applied on a micro or macro level.

Integration with Microsoft Project – ties the development process and developer activity to overall management of the software project(s)

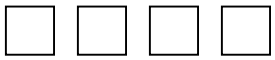
Version control and audit trails – manage change at the file, project and sub project levels for granular auditability

Global development group support – through a federated server architecture, diverse software development teams in various geographic locations can collaborate on change in real time.

Build control – enables repeatable and reliable builds in multi-platform environment

Conclusions

In situations where an organization is dealing with the aftermath of a merger or is going through rapid transition and organizational churn, it is not recommended to commence large-scale quality initiatives such as Six Sigma or TQM. Such initiatives are unlikely to survive long enough to provide a genuine ROI. For most organizations, there is too much risk, too high a financial investment in the required level of training without sufficient focus on results. It is not completely unheard of for these initiatives to survive in such conditions, but it is quite rare and demands an exceptional level of leadership and commitment.



There can be similar issues with wholesale adoption of benchmarking methods to achieve 'a level' for the sake of it: sensitive and selective use of the models, aimed at making improvements to reach particular business goals, is an approach which is much more likely to succeed.

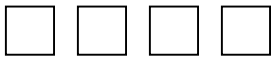
To reap rewards from quality initiatives during the early stages of a merger, ensure that you have clear and visible mechanisms in place for focusing on quick wins with SMART objectives and verifiable results in each of the following areas:

- Technology – make it future-proof
- People – ensure that their skills are honed
- Processes – make them visible and consistent

Consider combining your processes with underpinning technology so that there is more likelihood of synergy, compliance and more predictable and better quality results.

Remember that, until the post-merger integration process is well under way, there will be suspicion as to the 'hidden agenda' of any attempt at measuring software quality, team productivity or maturity of processes. Overcome this by ensuring that there is frequent, clear communication as to future plans, successes and failures; opting for team-based workshops 'across the divide'; and ensuring that all measurement or assessment of quality is done as objectively as possible.

Above all, keep it simple, keep it forward-looking and focus on positive, achievable results.



Appendix A

Principles for managing organizational change – a quick checklist

- Expect & allow for a period of mourning for the old

- Analyze the past
  - Take forward what is good
  - Throw away the no-longer relevant
  - Balance-sheet AND a people-based approach
  - Cost reductions you need to achieve

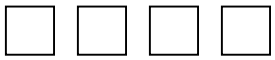
- Define business goals for the skill-sets you need in the future
  - Define what skills are most appropriate for the future needs: technical and interpersonal
  - Consider People-CMM[xxiv] as a basis for understanding the maturity of the processes relating to human capital, as evidenced across different parts of the organization

- Audit the skill-sets that you have now
  - What technical skill-sets do people possess?
  - Are these skills relevant to the new organization?
  - What personal/communication skills do they have?
  - Are they willing/able to move forward?

- Define the gap & make action plans to move forward
  - Use the skills audit to form a skills-matrix, against the technologies that you intend to employ in the future
  - Gaps in knowledge will demonstrate where to train people, or whether their combined skills and experience demonstrate that they should move on

- Encourage everyone to think and act as a leader
  - Help everyone to gain the most from group synergy
  - Help ensure that the action plans are tracked and actively progressed
  - Review progress in light of changing organizational needs

- Communicate early and frequently
  - Intentions, concepts
  - Plans; even communicating a lack of firm plans can help
  - Successes and failures
  - Be as open as possible, to avoid negativity created by uncertainty



Appendix B
Defining technology futures – a quick checklist

- Perform and audit of, and compile an inventory, of the systems you have now:
  - Where are they?
  - What environments are they running?
  - What applications are they running?
  - What business functionality are they supporting?

- Business function to systems audit: to what extent are the systems supporting the business goals now?
  - Critical goals
  - Highly desirable
  - Nice-to-haves

  Use the audit as basis for making some hard decisions. Are there any obviously redundant systems, e.g. that are candidates for 'tapping' for their useful data, then scrapping?

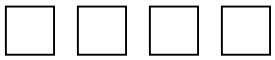
- Define blueprint for future systems
  - What business needs must be met in the future?
  - Define the architectural components that are needed - what and where
  - Define the technical characteristics of your preferred future technical solutions – how systems should be in the future
  - Keep options open and checkpoint regularly against progress and continuing business needs

- Future-proofing
  - Technology type
  - Scalability
  - Flexibility

- Define detailed mapping of existing/near future systems, against the blueprint
  - Consider Constructive Evaluation Methodology for detailed work.

- Look at the 'ilities' in increasing levels of detail, for critical systems
  - Portability (for moving to bigger, more modern tech base)
  - Flexibility (for coping with future change)
  - Maintainability (for when those core staff have left)
  - Auditability (obvious for financial institutions)
  - Security (24/7 availability to hackers)
  - Capacity (essential if expanding customer base)

- Define migration strategy
  - Buy afresh
  - Build afresh
  - Enhance existing system(s)
  - Combine existing systems



Bibliography

Author

The author has 20 years experience in IT, working as a developer, project manager, quality assurance and process improvements specialist: approximately 15 years have been with large banks and financial organizations in the UK, mainland Europe and the USA. All of these organizations have: (a) implemented one or more of: Six Sigma, TQM and CMM and (b) been the result of mergers or were actively engaged in dealing with the aftermath of merging. The author helps organizations to implement practical solutions to improve software quality.

References

[i] Six Sigma: Motorola developed Six Sigma in the early 1980s and have derived enormous benefits from it (also see ref vi below)

[ii] TQM: Total Quality Management, most notable proponent of which is Phil Crosby 'Quality is Free'; McGraw Hill, 1979; and 'Quality is still free', McGraw Hill, 1995

[iii] CMM: Software CMM originally published by the Software Engineering Institute 1994; there are other models, such as the more recent CMM-Integrated, published August 2001 and People CMM (also referenced later). In practice, most references to 'CMM' still mean the Software CMM, as that has had most up-take to date.

[iv] Cendant Media Center Trends, see : http://www.cendant.com/media/trends_information/trends_information.cgi/Financial+Services/59

[v] Also useful articles on this site regarding profitability of internet banking: http://www.aba.com/About+ABA/_TECH_InternetBanking.htm

[vi] http://www.motorola.com/mediacenter/news/detail/0,1958,467_243_23,00.html

[vii] At GE, Six Sigma is 'the way we work around here' http://www.ge.com/sixsigma/

[viii] http://www.boeing.com/news/feature/baldrige/14mar_s84617.pdf and for CMM: http://www.boeing.com/news/releases/2002/q1/nr_020225m.htm

[ix] http://www.raytheon.com/press/1999/jan/seilev5.html

[x] http://www.juran.com/research/articles/article004.html

[xi] 'A Business Case for Software Process Improvement Revised - Measuring Return on Investment from Software Engineering and Management', an updated DACS State-of–the-Art Report; Thomas McGibbon, Sept 30, 1999; http://www.dacs.dtic.mil

[xii] Some of the ones that the author has found useful are:
'Why Change Doesn't Work – why initiatives go wrong and how to try again and succeed'; Harvey Robbins and Michael Finley; Orion Business Paperbacks, 1998
'Fast Forward – the best ideas on Managing Business Change'; edited with an introduction and epilogue by James Champy and Nitin Nohria; Harvard Business Review book 1996

[xiii] 'Funky Business – Talent Makes Capital Dance'; Jonas Ridderstrale and Kjell Nordstrom, ft.com Bookhouse Publishing, 2000

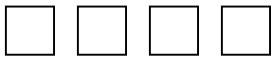
[xiv] SMART: Specific, Measurable, Achievable, Realistic and Time-Bound

[xv] Lloyd's Register Quality Assurance http://www.lrqa.co.uk/Default.htm?trainingservices/pdca.htm~mainFrame

[xvi] 'The Capability Maturity Model', Paulk *et al*, CMU SEI, Addison Wesley, 1994

[xvii] Abraham H Maslow: 'Motivation and Human Personality', 1954

[xviii] 'Diffusion of Innovations'; Dr Everett M Rogers, The Free Press, 1995



[xix] 'Funky Business' *ibid* please see reference xiii above

[xx] 'Constructive Evaluation Method', based on the principles of Quality By design. Developed by Professor Vic Stenning, *et al,* as a practical, iterative, method of ensuring that software and systems are developed according to rigorous principles, to meet their stated objectives. Contact Prof Stenning for consultancy and advice on using the method. Downloadable copy of method from http://www.anshar.com

[xxi] Software Quality Engineering principles, as proposed by Tim Kasse, one of the contributing authors of the Software CMM who continues to provide inspiration and guidance on software process improvements: http://www.kasseinitiatives.com

[xxii] 'Constructive Evaluation Method', *ibid* please see reference xx

[xxiii] For a useful discussion of developing a technical architecture, see: 'Re-shaping IT for Business Flexibility', Mark Behrsin, Geoff Mason, Trevor Sharpe; IBM McGraw Hill Series, 1994

[xxiv] 'The People Capability Maturity Model': Curtis, B., Hefley, W.E., & Miller S; CMU Software Engineering Institute, 1995. Drs Curtis and Hefley both provide consultancy and assessment services around the People CMM and also the Software and Integrated CMMs. See the SEI website for further details - http://www.sei.cmu.edu