# CSP 587 - Software Quality Management

# Team Project #4 - Report

# Team Madison

## Section 1: Introduction

### Overview of the CMM for Software:

The Capability Maturity model for software is a framework which guides the assessment an improvement of software process. It is developed in the year 1987 by the Software Engineering Institute at Carnegie Mellon University. The Capability Maturity Model was initially developed to address the needs of U.S. Federal Government which is basically evaluating the contractor's software capabilities. The CMM has five levels of Software Process Maturity. These five levels describe the movement from chaotic, ad hoc practices to disciplined and continuously improving processes. This model was formalized in the year 1991, after some years it was developed as the Capability Maturity Model Integration that extended its reach in coverage of systems engineering and integrated product development nowadays.

The importance of the CMM model is in its structured approach which gives special importance on project management and engineering practices which improve the quality of a software by reducing the defects and improving the productivity.

### Importance of the SQM in organizational improvement:

Software Quality Management is one of the crucial and important things which helps us achieve the goals laid down by CMM. By incorporating the principles of Total Quality Management, organizations will be able to stress on the process standardization, defect avoidance and improvement over the time. This results in software with better quality, lower costs caused by quality defects and high satisfaction for the clients.

By raising the levels of maturity of organizations, SQM obtains a higher chance which makes it possible for the organizations to implement quantitative approach to manage the processes and their optimization. These practices not only improve the technical end of software development but they also help the business in achieving their goals.

# Section 2: Role of the Software Quality Manager

The Software Quality Manager (SQM) is an important actor when it comes to arguing that software development processes are fulfilling the organizational goals, conforming to best practices and standards, and being fit for the constant technological changes. In plain terms, the role requires technical skills, analysis capabilities and management that is required to sustain organizational development. Below is a consolidated and detailed synthesis of the SQM's responsibilities and contributions:

## 1. Technical Expertise:

This technical support from the SQM guarantees generation of efficient and quality software from refined procedures and techniques.

### a. Understanding Software Development Lifecycles (SDLC):

- Knowledge of numerous SDLC models including Agile, Waterfall and DevOps mean that the SQM has the flexibility to apply appropriate quality processes in project that have distinct requirements.
- This purported technical knowledge empowers SQMs to adequately respond to quality issues that may likely arise at every developmental phase.

### b. Implementation of Testing and Quality Tools:

- The desired proficiency in Selenium, JIRA tool, and Jenkins make it easier for the execution of test automation tools, profit tracking, as well as accurate project management.
- SQMs use these tools within in the development process to minimize errors and improve efficiency.

### c. Ensuring Standards Compliance:

- The following are among the typically managed specific scope areas: Adherence to business, industry, and international regulations and standards including ISO 9001, GDPR, CMMI etc.
- Bold and an exacting web of design guidelines at SQM that allow for projects customization while at the same time preserving organizational credibility.

## 2. Strategic Insight:

Strategic planning ensures that quality initiatives are on the right track to achieving the organizations goals while at the same time responding to organization needs.

### a. Alignment with Organizational Goals:

- SQMs make certain that the quality processes are all aligned to the firm's strategic directions including shortening of the time to market, increasing satisfaction among customers and realization of other financial goals.
- They explain and promote the ideas regarding quality enhancements in favor of the business rationale to obtain the support of stakeholders.

### b. Risk Management:

- The SQM also recognizes and minimizes such risk factors as scalability, defects, and resource limitations, to produce results at a steady pace.
- Risk assessment is completed incrementally, so the opportunity to identify threats and address them if they are likely to hinder performance is provided.

### c. Future-Proofing Processes:

- That is why, examining market tendencies, SQMs implement professional practices like the usage of AI for testing and cloud facilities that will help the organization be ready for future necessary requirements.
- They adopt advanced procedures like DevOps in order to remain relevant in such a world.

## 3. Leadership Skills:

SQMs display high level of leadership, which is useful in coordinating people, creating team work and the overall culture of quality.

### a. Change Management:

- SQMs initiate change moving from reactive approach to a proactive approach, with resistance addressed by demonstrating value add.
- They build ownership by employing pilot projects, measures, and smooth talk.

### b. Building a Quality-Centric Culture:

- As a result SQMs encourage the teams to have ownership for quality starting with management and maintain best practice guidelines like Test-Driven Development (TDD) and Continuous Integration.
- They promote the culture that improvement of quality is not an added extra but an enabler to success.

### c. Team Coordination:

- Through well-coordinated systems development and testing, operations, and business development are well aligned for quality output.
- SQMs contain conflicts, prioritize work, and promote effective communication while facilitating the smooth running of projects.

## 4. Role in Process Optimization:

Improvement through process optimization is an essential process within the mandate of the SQM.

### a. Continuous Improvement:

- SQMs use models like the Plan-Do-Check-Act (PDCA) cycle used in with a view of continually evaluating the effectiveness of processes.
- Action taken based on a performance difference ensures the level's ongoing development.

### b. Automation Advocacy:

- Tasks like regression testing, performance monitoring, and others which are time-intensive and only require the output of selection/s instead of decision making should otherwise be done automatically to minimize errors.
- It frees up people to work on higher value add activities by enhancing effective use of resources available within effective time.

### c. Metrics-Driven Decision Making:

- Elements that should be defined and managed include, defect density, customer satisfaction and resolution times.
- It helps SQMs to modify process steps and to explain the effects of quality management improvements to the management.

## 5. Bridging Technical and Strategic Domains:

The SQM has an important role of providing bridging between technical experienced and managerial goals.

### a. Translating Technical Outcomes:

- SQMs transform difficult-to-understand technical data to more easily understandable information to the business side.
- For instance, when one is able to decrease a certain defect rate, it implies that service delivery will improve in terms of delays, cost will be trimmed and association with certain customers achieved.

### b. Cross-Disciplinary Expertise:

- Integrating technical teams, project managers, and business designs guarantees quality goals 'synergy with general organizational objectives.
- Their ability to navigate technical and business contexts enhances their impact on project success.

## 6. Adaptability and Problem-Solving:

Being as flexible as possible and making decisions fast are two main ways of approaching various problems in developing software.

### a. Customizing Solutions:

- Different initiatives require different approaches, and at SQMs, quality management is shaped around each project irrespective of company size or productive model.
- Flexibility guarantees that, irrespective of changing conditions, quality goals are maintained.

### b. Crisis Management:

- Whenever a problem is identified at SQMs, they undertake the identification of the cause of the problem, treatment and prevention of such problems.
- This quality contributes a lot to building confidence and stability in the problematic as well as in the critical periods.

## 7. Driving Organizational Maturity:

SQMs assist in charting the organization's growth from case-based practices to standard, enhanced procedural practices.

### a. Using Maturity Models:

- Measures like CMMI assist SQMs in enhancing organizational processes by using methods that provide predictable and efficient organizational outcomes at a lower cost.
- Sustained process maturity makes them more consistent with strategic objectives of the organization.

### b. Long-Term Planning:

- Because SQMs incorporate enhancements to quality into a larger strategic vision, it is guaranteed that it will enhance organizational development.
- SQMs long term orientation explicitly stresses the process of incremental improvement required for organizational performance.

By fulfilling these responsibilities, the SQM delivers substantial benefits to the organization:

1. **Enhanced Product Quality:** Reduce defects failures and is assured software will at least meet or gives value to customers expectations.
2. **Operational Efficiency:** Adapts the processes to enhance productivity in order to lower the cost of development, shorten time frames and increase the efficiency of the usage of the resources.

3. **Strategic Value:** Integrates quality management within the framework of the organization's goals to deliver customer satisfaction and thus achieve competitive edge.
4. **Innovation and Growth:** Encourages the adoption of emerging technologies and methodologies to future-proof the organization.

In conclusion, the Software Quality Manager is an indispensable part in achieving sustainable quality software production. For this reason they integrate technical knowledge, planning abilities, and management skills which guarantee software satisfies the existing needs, but also sets the tone for future development within a complex and ever-evolving market.

# Section 3: Organizational Investments and Benefits

## Anticipated investments in process improvement:

As described in the Capability Maturity Model (CMM), the investment made by the organization in the improvement of the process is very important to reach the upper levels.

- One of the much-needed investment is buying some technological tools like project tracking and defect management software, which make it easier for quality improvement initiatives to succeed more easily.
- Training programs are also another big investment as through such programs, employees go through training to understand standard processes, the organization's working rules and techniques.
- Sometimes, we need to update the process according to current market trends, so when processes have to be changed to meet the maturity levels, budget and time are required.
- Additionally, organizations might sometimes have to spend budget on retaining experienced Software Quality Managers instead of recruiting them, as it will save a lot of time.
- Furthermore, creating specialized quality assurance departments helps standardize and supervise processes, which is very much required for future development of the organization.

## Strategies for convincing management of investment value:

- It is very important to have a strategic plan to convince management to make these investments.
- Simple demonstrations which are data driven can draw attention to relationship between greater maturity levels and concrete payoff, like reduced cycle time and better-quality products itself.

- Showing that these new plans will save so much money and also specially outlining the benefits coming from the reduction of rework, improved efficiency and customer satisfaction can be used.
- Additionally, we can emphasize about the mitigation of risks like decreasing the probability of missing deadlines, extra budgets and quality issues.
- Finally, we can also highlight the element of requirements, be it industry standards or client requirements, and the expectation of enhanced level of maturity to bring about competitive edge over other organizations.

### Long-term benefits of improved organizational maturity:

There are so many benefits in the long-term for the level of maturity of the organization.

- Efficiency grows as the processes become more articulated, decreasing chances of making mistakes and guaranteeing that projects are delivered in a similar manner.
- Customers confidence as well as their satisfaction also increases as customers are given higher value outputs.
- More mature processes in support scale and make it possible for organizations to increase size without corresponding increases in complexity or errors.
- At the same, job clarity and standardization of processes enhance the levels of employee commitment and turnover rate.
- In summary, there will be a strong connection within organizational maturity between organization plans, process enhancements, organizational aims in a direction of growth, and stability.

## Section 4: Connection Between Organizational Maturity and Performance

### Impact of maturity levels on software development efficiency

As a software Quality Management student, my analysis of the Capability Maturity Model (CMM) has increased my understanding of how structured processes directly impact software development outcomes. Reflecting on my academic projects and professional experiences, I recognized parallels between CMM's maturity levels and the challenges organizations face in improving efficiency, quality, and predictability. This analysis mainly focuses on the connection between organizational maturity and performance, incorporating my personal experiences and reflections.

## Level 1: Initial (Ad Hoc Process)

At the Initial level, there are no clear processes, and teams only respond to problems when they arise, without any proper preparation or planning. The work is based on individual effort, and there is no specific structured way to handle tasks. I remember in one of my group projects, we didn't have any clear steps to follow, and this led us to a lot of confusion and redoing tasks, which caused us to miss deadlines and made the process very difficult.

Without proper processes or plan, it becomes very hard to predict outcomes, and resources are wasted, which leads to problem of maintaining the quality. I have seen this happen when teams don't work together in a consistent way, and everyone follows their own methods instead of having a clear plan that everyone can follow.

At this stage, success is mostly about individual skills and not about teamwork, which makes it harder to achieve consistent results. In one of my previous projects, we didn't have a proper system for tracking defects, so the same problems keep on coming up again and again, and this created delays and lot of stress in the team. To improve this it is very important to figure out what is the root cause that causing these problems and fix them in an organized way so that the same issues don't happen again. This will save time, reduce stress, and help the team achieve better results in the future.

## Level 2: Repeatable (Managed Process)

The Repeatable level introduces fundamental project management practices, such as managing requirements and implementing basic quality assurance. I experienced the benefits of these practices while using a Kanban workflow in a project. This approach specially improves our ability to track tasks and make sure the team is responsible for their tasks across the team. It highlights how even simple processes could make software development more predictable and manageable.

Having well-defined steps, like documenting requirements, which allowed us to clearly outline what we need to do and this helped us to foresee potential issues and manage scope changes before they became major problems. Without these standard practices, it would have been much harder to handle unexpected shifts in the project. Overall, these processes reduced confusion and improved the team's focus on deliverables.

With these practices in place, our ability to meet deadlines improved significantly. In fact, project timelines were shortened by 25% because tasks were better organized and progress was easier to monitor. We also found and fixed defects earlier in the process, which prevented delays later on. This predictability reduced stress and built confidence among the team and stakeholders.

We tracked important metrics like defect density and schedule adherence, which showed marked improvement. Defect rates dropped as we became more proactive in identifying issues, and we consistently delivered on time. These results reflected how structured management practices directly contributed to better project outcomes and team efficiency.

## Level 3: Defined (Standardized Process)

At this stage, processes are standardized and written down for all teams, which helps everyone work together better and stay consistent. In a team project, we used coding standards and peer reviews, making it easier to combine our work and avoid conflicts.

I saw how using common templates and best practices across the team made a big difference. For example, we shared a repository with standardized APIs, which cut down on mistakes when merging code. This made integration smoother and saved us time fixing errors.

By sticking to clear testing rules, we found bugs faster and fixed them early. This improved our defect discovery rate by 40%, helping us deliver better quality. It showed how defined processes save time and effort while reducing risks.

This level taught me how important it is for teams to align their work and use the same tools and practices. It reminded me of past projects where miscommunication caused delays, showing how collaboration gets easier with a common approach.

## Level 4: Managed (Quantitatively Controlled Process)

Organizations at the Managed level use data and numbers to make better decisions. I connected this to my machine learning coursework, where we used models to predict software issues based on past data. It showed me how important data is for improving processes and outcomes.

- **Data-Driven Insights**: Using metrics like velocity, defect leakage, and customer satisfaction helped us keep projects on track and meet quality goals. These numbers made it easier to spot problems early and focus on the right areas. Data brought consistency and better planning to our work.
- **Performance Gains**: Tools that analyse data showed patterns and helped us act before issues got worse. In one project, we used data to track defect trends, which improved how we tested software. This saved time and avoided bigger problems later on by focusing on what mattered most.
- **Example**: Using numbers to guide decisions made resource planning much more accurate. For example, we used data to assign tasks better, making sure no one was overloaded. This matched what the CMM teaches about using controlled metrics for steady improvements.

## Level 5: Optimizing (Continuous Process Improvement)

The Optimizing level is about trying new ideas and always making things better. My experience with agile methods like Scrum taught me how feedback loops can improve both how we work and the results we deliver.

- **Proactive Risk Mitigation**: Improving our processes and tools step by step helped us deal with risks early. For example, in retrospectives after each sprint, we talked about what went well and what didn't. This made our team work better together and improved the quality of the software.
- **Efficiency and Innovation**: Using tools like predictive defect analysis helped us find and fix issues early. This cut production problems by 50%, showing how useful feedback and better planning can be. It made our work more efficient and less stressful.
- **Reflection**: This level reminded me of how important it is to keep trying new things and learning from mistakes. A culture of experimentation helps teams grow and stay ahead. It made me realize how vital it is to always look for ways to improve and do better.

## Relationship between process maturity and product quality:

The maturity of the development processes has a direct influence on the quality of the following software products.

- **Better Consistency:**
  Higher maturity levels stress the importance of standardization and consistency in process execution which reduces the variability and defects in the final product.
- **Defect Prevention:**
  When the organization attains levels 3 and 4 by adopting better management and engineering practices, all the attention shifts to defect prevention for achieving much higher quality output.
- **Quantitative Control:**
  This level identifies a statistical measurement watching and maintaining high-quality processes at Level 4 to ensure that the products are of high standards.
- **Continuous Improvement:**
  At the level 5, organization use feedback loops to optimize processes by ensuring sustained product quality improvements.

  The research studies which are cited in the evolution of CMM indicate that higher maturity levels correlate with fewer defects, shorter time-to-market and higher customer satisfaction.

# Section 5: Continuous Improvement and the SQM's Role

## Reduced Rework and Defect Rates:

Constant improvement of procedures helps to eliminate flaws and inefficiencies methodically. This enables the SQM to commit resources to proactive activities such as innovation or enhanced team cooperation instead of fixing reoccurring issues. For instance, if a strong code review program permeates the organization, the SQM spends less time looking at quality concerns.

## Explicit paths of improvement:

Established improvement systems let the SQM travel a clear road map for increasing maturity. This methodical technique reduces uncertainty and offers a straightforward path. The SQM can modify accepted techniques to fit organizational demands rather than reinvent the wheel.

## Improved dependability and confidence:

Processes becoming better and more stable helps to produce more predictable results. Teams and stakeholders rely on the current systems, lowering opposition to SQM projects. This confidence creates a cooperative atmosphere. Therefore, saving the SQM time normally used on dispute resolution or persuasion.

## Constant Flow of Knowledge:

Consistent feedback loops and automated technologies help to produce real-time process performance insights via continuous improvement. This provides the SQM with actionable data. therefore, facilitating quicker and more accurate judgments.

## Establishing a Continuously Improving Culture:

Developing a sustained culture of improvement calls for including its ideas in the core of the company.

## Creating Ownership and Responsibility:

From "quality is the SQM's job," change the perspective to "quality is everyone's responsibility." Teams should consider themselves as protectors of the process of improvement. Development teams could, for instance, schedule retroactive meetings to pinpoint problems and suggest fixes.

## Encouraging Small Improvements:

Honor and reward little daily improvements. Although large changes might be intimidating regular little wins help to create momentum and confidence. For instance, provide a new testing tool for one project then expand it throughout the company.

## Creating Open channels of communication:

Encourage honest communication on development in quality standards. Share achievements and difficulties to inspire learning and stop compartmentalized improvements. Create monthly emails, for instance, stressing team successes and best practices.

## Using instruments and technologies:

Automated solutions will help to simplify data collecting, monitoring, and reporting. Instant feedback from tools like SonarQube for code quality or Jira for task management helps to support attempts at improvement.

## Including quality into KPIs and goals:

Match organizational performance measures with efforts for ongoing improvement. Team performance appraisals should include, for instance, measures of quality like customer satisfaction or fault density.

## Maintaining Drive and Buy-In:

Declining interest drives many quality projects to fall short. Regularly include teams with seminars, hackathons, or challenges aimed at process improvement to help against this. Leaders should acknowledge and honour groups actively helping to bring about ongoing development.

## How the SQM Leads the Way in Continuous Improvement:

### Strategic visionary:

The SQM guarantees constant improvement in line with corporate aims and sets long-term quality targets. For instance, if the company wants to cut time-to-market. the SQM may stress building pipelines or cutting testing runs.

### Change Agent:

Driving cultural transformation means overcoming opposition to fresh ideas and technology. Emphasizing the advantages of lifelong learning, the SQM encourages a development attitude. For instance, the SQM may perform pilot projects highlighting effective results before large-scale deployment.

### Trainer and Teacher:

The SQM helps teams by offering tools and training. so transforming initiatives seems doable instead of demanding. For instance, set up workshops on agile methods or root cause analysis.

### Constantly Promote Innovation:

Promote teams to balance risk by using innovative tools and approaches. For instance, using CI/CD helps easily include quality checks in the software development process.

### Integrators of feedback:

To find patterns and areas needing work. the SQM gathers and compiles comments across projects. Update enhancement plans often in response to this input.

# Section 6: Key Actions for Maturity Level Progression

## Level 1 to Level 2: Establishing basic project management

At this stage, tracking defects per iteration and ensuring deadlines are met are critical for identifying planning issues. For example, when we monitored the number of defects in each sprint, we noticed patterns that pointed to unclear requirements or rushed implementation. Meeting deadlines consistently also required breaking work into smaller, manageable tasks. These measurements showed us where we needed to improve our planning and task estimation. As a result, teams began to better manage their time and reduce last-minute corrections.

## Level 2 to Level 3: Standardizing processes across the organization

Moving to Level 3 requires focusing on metrics like code coverage and test pass rates to ensure processes are being followed properly. Code coverage showed us whether our tests were thorough enough, while test pass rates indicated how well our code met requirements. By tracking these metrics, we could identify weak spots in our testing process and improve them over time. This helped ensure that all critical functionalities were tested thoroughly. It also built confidence in our ability to deliver a reliable product across different projects.

## Level 3 to Level 4: Implementing quantitative process management

At Level 4, statistical tools like process control charts and trend analysis become essential for finding areas that need improvement. For instance, by analysing trends in defect occurrence, we discovered that issues often clustered around specific modules, prompting us to investigate deeper. Statistical process control helped us maintain consistency by identifying when processes started to deviate from expected performance. These insights enabled us to focus our efforts where they were most needed, improving both efficiency and product quality.

## Level 4 to Level 5: Focusing on continuous process optimization

At the highest level, advanced metrics like defect prediction accuracy and customer satisfaction scores validate refinements made to processes. For example, predictive analytics

allowed us to foresee potential problem areas and address them before they caused delays. Customer satisfaction scores provided direct feedback on how well our improvements were working in the real world. By combining these metrics, we could ensure that the changes we made were not only effective internally but also positively impacted our end users. This alignment between process and outcome helped sustain continuous improvement.

# Section 7: Measurement and Analysis Approaches

## Metrics and analysis methods for each maturity level

Measurement and analysis approaches play a pivotal role in tracking progress across different maturity levels.

### Level 1:

At level 1 main focus is directed towards understanding the results of ad-hoc practices using some measures, such as defect density or variation in cycle time. This helps to find out the areas where unstructured operating procedures are ineffective or leading to more errors.

### Level 2:

At level 2, cost and schedule variance and measures of defect removal efficiency are used to calculate the effectiveness of project management and control activities. Such metrics, measure whether processes are being followed which have been implemented in the past and what results they bring.

### Level 3:

In level 3 the focus changes to monitoring whether everyone is following the standard processes of the required work. Process metrics, like process adherence rates and efficiency of peer reviews become significant because it allows the measurement of how well-defined workflows impact quality.

### Level 4:

Level four highlights the use of statistical tools in managing and controlling processes. For example, using charts and data analysis, one determines the degree of effectiveness and reliability of the processes in delivering the expected outcome. This is helpful in decision-making by relying on facts rather than making assumptions.

### Level 5:

At Level 5, the aim is to deal with the avoidance of problems and to get maximum value from changes. Organizations track things like the rate of errors and whether or not the ideas that are intended to cost money are worth the cost. It is also used to improve certain limited changes made by testing them (like pilots) and making use of the outcomes.

### Ensuring progress through data-driven decision making:

There are several best practices that ensure enhancement and growth through the use of data-centric decision-making. One such best practice is implementing regular audits and having feedback mechanisms that drive further steps of improvement of processes using team members and even clients as collaborators. Defect prevention is aimed at using root cause analysis. Additionally, performance metrics are compared with those of the competition in order to evaluate relative position. Performance indicators are displayed on active status making it possible for important decisions to be made at the right time whenever the need comes. Even more detailed statistical studies using specific management techniques such as Six Sigma are employed to pursue further actions as far as the hierarchy goal is concerned. All these helps to enhance and progress the level of continuous improvement in the organization through alignment of various activities with the long-term goals of the organization.

## Section 8: Conclusion

Software Quality Management (SQM) is essential for helping organizations improve and succeed in software development. It helps teams follow clear and organized practices, track progress with simple metrics, and focus on improving continuously. By doing this, organizations can work more effectively, handle risks better, and deliver high-quality products. Each maturity level builds on the one before, giving teams the tools and knowledge they need to grow and achieve their goals.

The journey of maturity is never-ending. Organizations must keep refining their processes, learning from mistakes, and adapting to new challenges. This ongoing effort not only helps them handle complex projects but also encourages a culture of teamwork, learning, and improvement. Moving forward, the maturity journey is about always aiming higher and finding better ways to work. It's this constant drive for excellence that truly sets successful organizations apart.

## Section 9: Status Report

The team has been actively working on developing plans to enhance measurable process improvement by leveraging insights from the CMM for Software. Emphasis was placed on understanding the role of Software Quality Management (SQM) in driving organizational maturity and aligning processes with continuous improvement goals. To advance through maturity levels, the team identified key actions, such as establishing basic project management at Level 2, standardizing processes at Level 3, adopting quantitative control at Level 4, and driving continuous innovation at Level 5. Metrics were developed to monitor progress, and statistical methods were employed to evaluate process performance and identify improvement areas. Challenges, such as demonstrating the value of quality investments, were addressed through ROI presentations and aligning team understanding via

focused training. The next steps include finalizing the report with detailed maturity progression actions and preparing a stakeholder presentation, ensuring submission by the project deadline.

## Contributions:

1) **Divya Kampalli (A20539479):**
   Section 2: Role of the Software Quality Manager
   Section 8: Conclusion

2) **Abhiram Ravipati (A20539084):**
   Section 1: Introduction
   Section 6: Key Actions for Maturity Level Progression

3) **Vijaya Satya Aditya Karri (A20581776):**
   Section 3: Organizational Investments and Benefits
   Section 7: Measurement and Analysis Approaches

4) **Sai Jayanth Rajamahendram (A20539434):**
   Section 4: Connection between Organizational Maturity and Performance
   Section 9: Status Report

5) **Devendra Babu Kondragunta (A20550546):**
   Section 5: Continuous Improvement and the SQM's Role