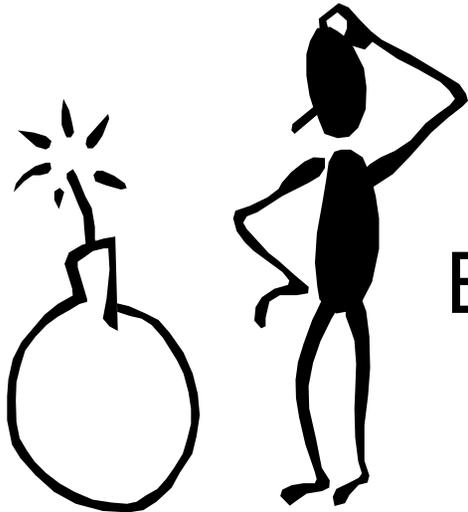# Strategic Quality - A Software Engineering Approach

M.Eng

Engineering Management

Dissertation

by

Roelof van Staden (9515747)

December 2004

Quotes

"A person who, seeing farther and probing deeper than other people, has energy enough to give effect to this extra vision."

George Bernard Shaw

"There is nothing either good or bad, but thinking makes it so."

William Shakespeare

"The problem with quality management is not what people don't know about it. The problem is what they think they do know."
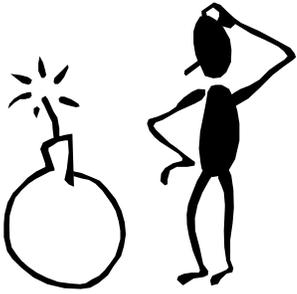
Philip Crosby

"The great end of life is not knowledge but action."

Thomas Henry Huxley

"Don't find fault. Find a remedy."

Henry Ford

Introduction

• A dilemma that numerous software engineering organizations face is to determine whether quality, and quality programmes should become part of the organizational strategy.

• Almost more important, how should the implementation of quality programmes in software engineering organizations be approached?

• Organizations realize that the quality of a service or product could make the difference between existing or closing down, hence an organized approach to quality is needed – a quality strategy.

• The software engineering industry faces it's own specific challenges, one of these being the quality of software systems.

• Numerous resources exist for identifying and measuring software quality, but lacks practical guidance in implementing quality approaches as part of an organizational strategy.

•The dissertation investigated the creation of a process framework for continuous improvement and measurement needed to align the strategic and quality goals of a software organization.

The Need for Software Quality

• A general view of quality is the totality of features and characteristics of a product or service to satisfy specified or implied needs.

• Juran states "fitness for purpose", and other authors mention "conformance to requirements" as elements of software quality.

• Statements about software quality imply that a link exists between software quality and user needs; i.e. the more closely the software matches the user requirements (needs), the higher the quality of the software.

• Conformance to requirements supports the concept that quality should be built-in during the design phases, because software quality is generally determined by the earlier phases of the software development process.

• In reality, software quality is more than just conformance to user requirements, it is about compromises and trade-offs to create software systems that provide tangible benefits to the organizations that utilize these systems.

The Need for Software Quality

• It is interesting to note that people generally spend about 50% of their time to write a program, and then the other 50% of their time trying to find the errors.

• In some programs that have been tested, more than 10 faults per 1000 lines are observed, which equates to 10 000 faults for a program that has 1 million lines of code.

• According to the Software QA/Test Resource Center the main reason that is being given for this unfortunate set of events is that software is generally not tested very well.

• Other conditions that are cited in the literature include:

- The complexity of modern software packages.

- The pressure to get the software product to market.

- The software industry does not accept any liability for errors.

- Poor work methodologies.

- Miscommunication or even no communication at all.

- Egos, because people like to believe that they are capable of many things.

- Poorly documented code.

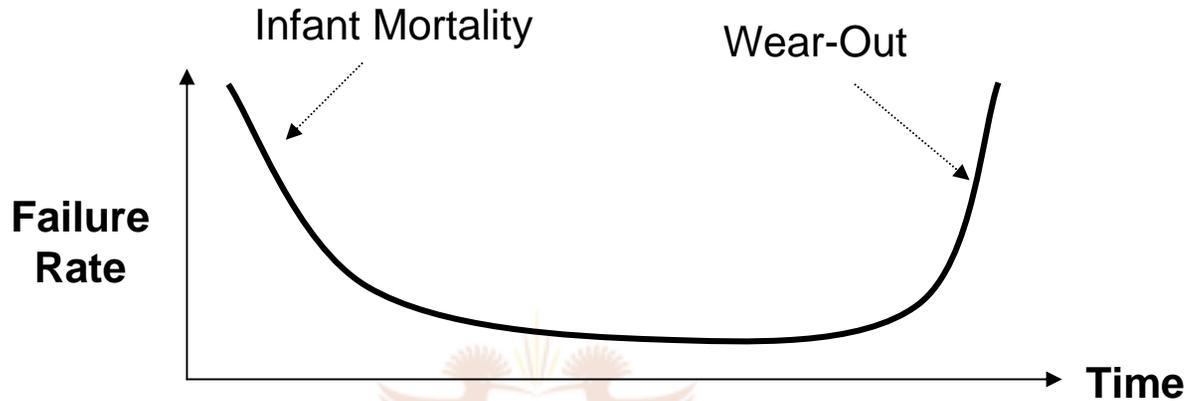- Software development tools that are flawed and contain errors.

Software Reliability

Infant Mortality

Wear-Out
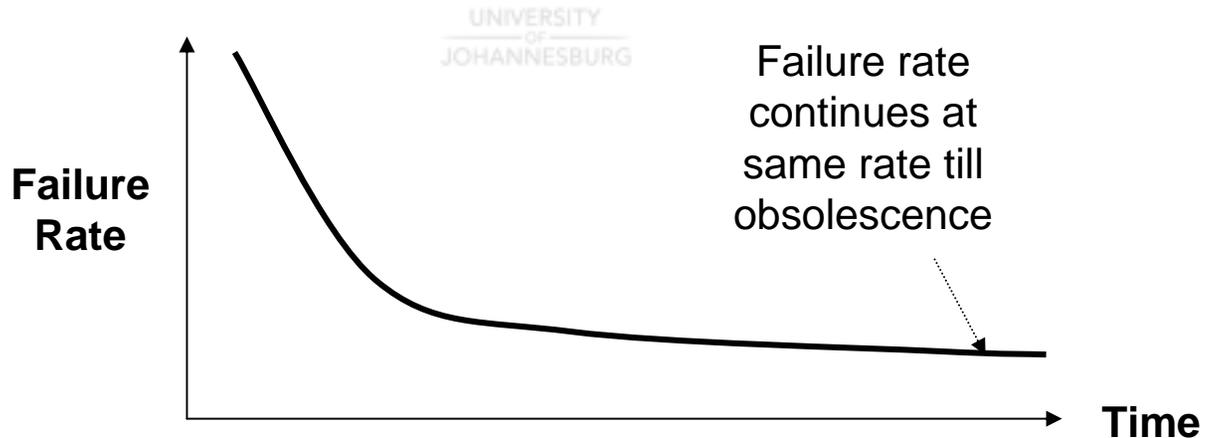
Failure Rate

Time

Figure 1: Hardware Failure Curve

Failure Rate

Failure rate continues at same rate till obsolescence

Time

Figure 2: Idealized Software Failure Curve

Software Reliability

Increased
Failure Rate
(Side Effects)

Actual Curve

**Failure
Rate**

Change

Idealized Curve

Time
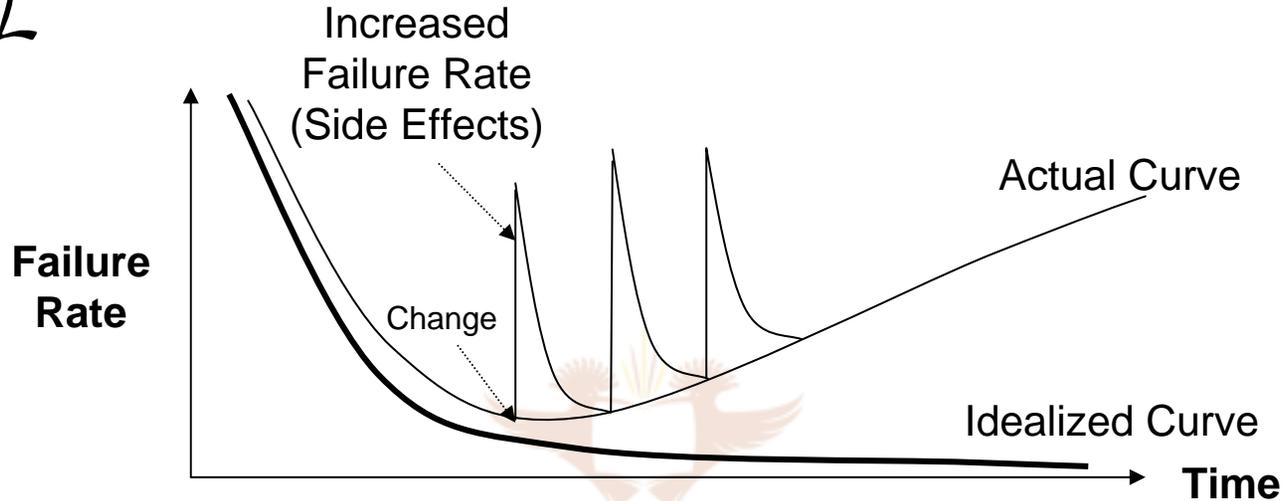
Figure 3: Actual Software Failure Curve

60 - 100x

Cost to
Change

1.5 – 6x

1x

Definition   **Development**   **After Release**
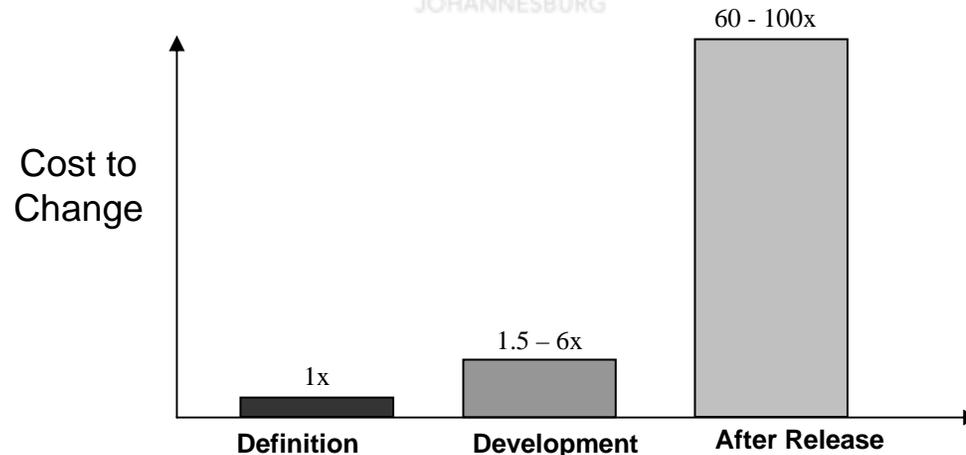
Figure 4: Cost of Changes to Software

Strategic Planning

• Strategic planning is a management function that is aimed at the improvement of the organizational functions, products and services.

• In order to produce a world-class quality strategy to enhance, strengthen, and improve the organization's competitive advantage, Kinlaw states that:

> *"…there are many tools that people need in order to undertake improvement projects in a careful and systematic way. They need problem-solving tools, such as flow-charts and cause-and-effect diagrams. They need to know to create run charts, histograms and Pareto charts. They need statistical tools to control their systems. They need to know to plan and collect data and how to use check sheets."*

• Together with all the tools and techniques a clear vision or overview of the process of improvement and measurement is needed. This means that software teams need an improvement and measurement model, and they require guidance in using such a model. A strategic framework is used to establish such an improvement and measurement model for software engineering organizations.

Strategic Planning

• In order to get everyone in the organization to change their behavior, to learn and to find ways to improve the way the organization functions, it is necessary to *develop a strategy, and to implement the strategy.*

• It needs to be understood that *a strategy's main aim is to improve the accomplishments and profitability of the organization in the short- and long-term on a sustainable basis*.

• An organizational strategy for a software engineering organization can be defined as a plan or tactics for improving current processes and methods.
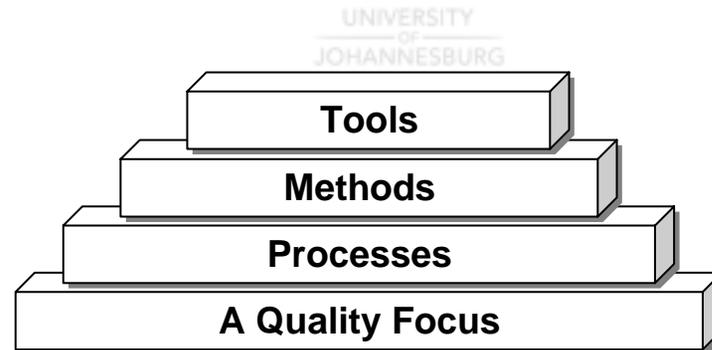
| Tools |
| Methods |
| Processes |
| A Quality Focus |

Figure 5: Software Engineering Layers

Strategic Software Quality Framework

Level	Description	Definition
1	Initial	*The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.*
2	Repeatable	*Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.*
3	Defined	*The software process for both management and engineering activities is documented, standardized, and integrated into an organization-wide software process. All projects use a documented and approved version of the organization's process for developing and maintaining software. This level includes all characteristics defined for level 2.*
4	Managed	*Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled using detailed measures. This level includes all characteristics defined for level 3.*
5	Optimizing	*Continuous process improvement is enabled by quantitative feedback from the process and from testing innovative ideas and technologies. This level includes all characteristics defined for level 4.*

Table 1: SEI Process Maturity Levels

Strategic Software Quality Framework

Level	Definition	Key Performance Area
2	Repeatable	*Software configuration management* *Software quality assurance* *Software subcontract management* *Software project tracking and oversight* *Software project planning* *Requirements management*
3	Defined	*Peer reviews* *Intergroup coordination* *Software product engineering* *Integrated software management* *Training program* *Organization process definition* *Organization process focus*
4	Managed	*Software quality management* *Quantitative process management*
5	Optimizing	*Process change management* *Technology change management* *Defect prevention*

Table 2: SEI Process Maturity Key Performance Areas

Strategic Software Quality Framework

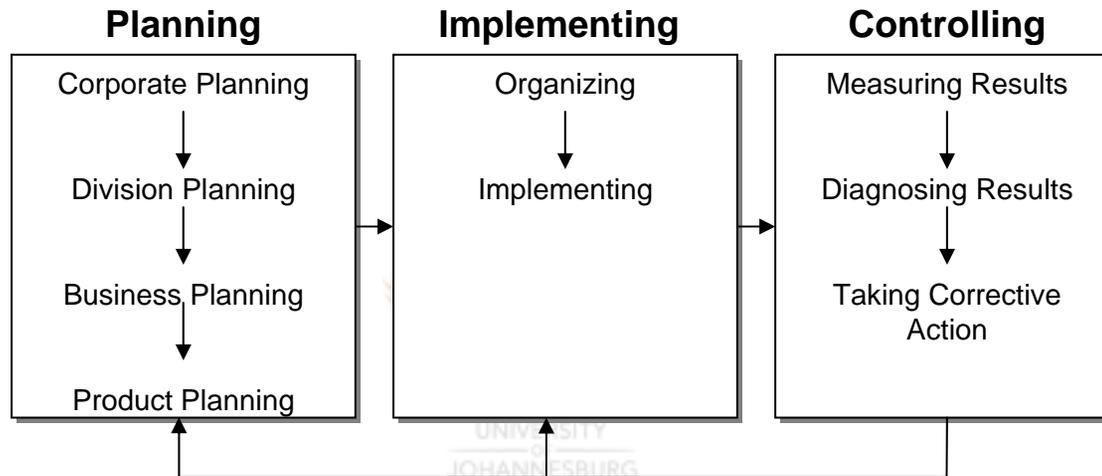| Planning	Implementing	Controlling
Corporate Planning	Organizing	Measuring Results
↓	↓	↓
Division Planning	Implementing	Diagnosing Results
↓		↓
Business Planning		Taking Corrective Action
↓		
Product Planning		

Figure 6: Strategic Planning Process

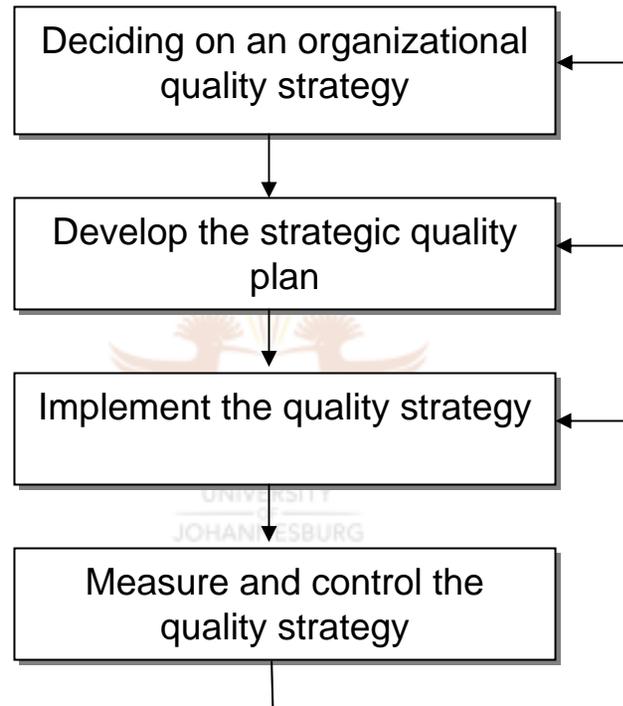# Strategic Software Quality Framework

Figure 7: Basic Quality Process Framework
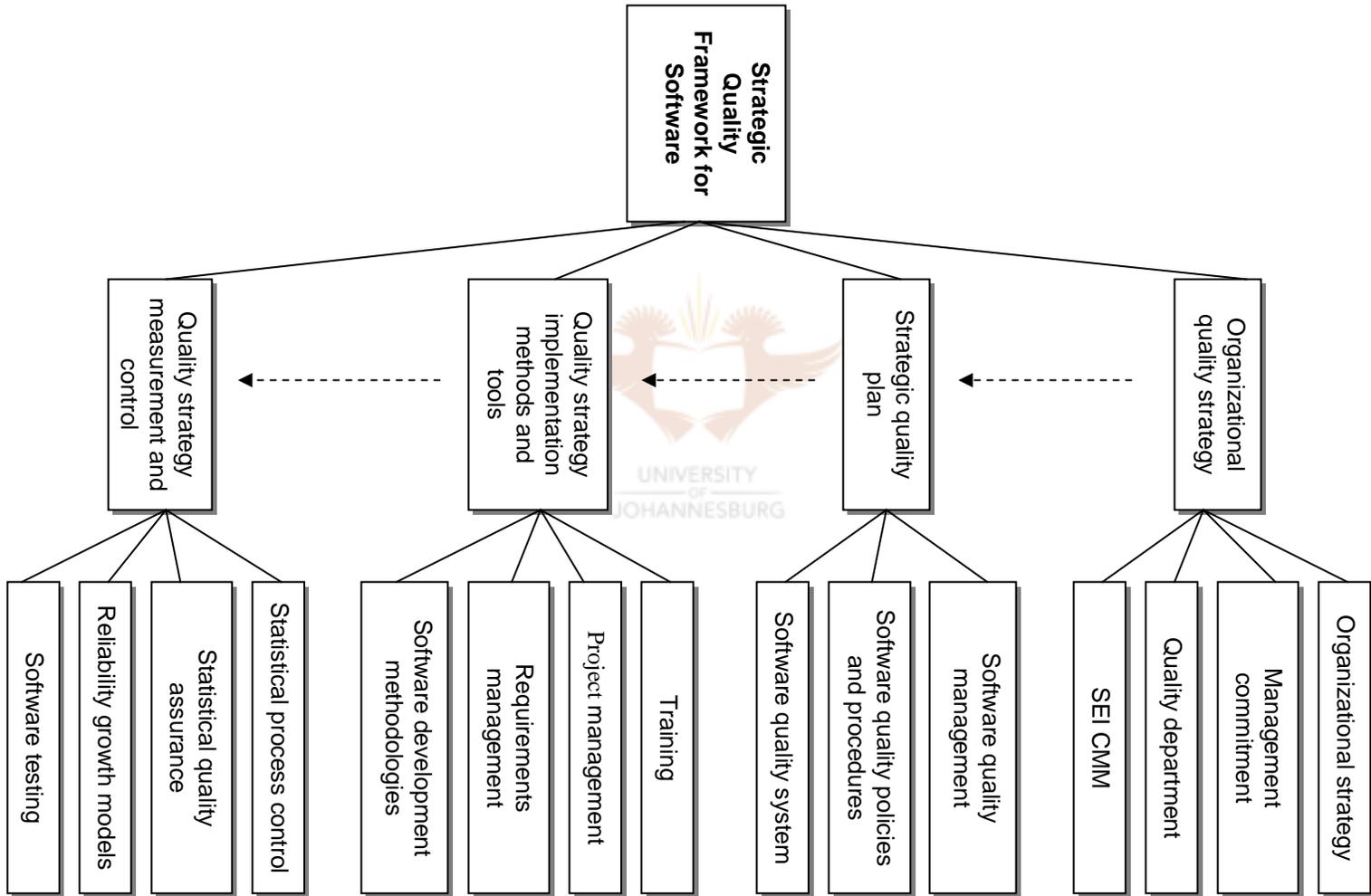
Strategic Software Quality Framework

Figure 8: Expanded Quality Process Framework

Conclusion

Quality assurance is an essential activity for software engineering organizations that produce software products and systems. Software developers and consultants generally agree that high-quality software is an important goal. In order to produce software systems of a high quality, a strategic quality framework as created by the author is essential to coordinate all the activities necessary in the development of the software system.

The strategic software quality framework is a guideline for the improvement of the quality of software engineering products and services. The framework should be adapted to suit the organization where it is applied, adding or even removing quality functions and processes that will not add value to the overall objectives of the organization. The framework features a measurement and control feedback loop, and the process of examining quality data should be performed on a regular basis to ensure that the framework is indeed improving the quality of products and services.

The topics of quality and strategy and the links between the two offer a wealth of research opportunities. The effect of organizational strategies on quality efforts should be investigated further, in an attempt to measure and define the benefits that are gained from adopting such practices. The software engineering industry is a fast paced environment, and moving towards an approach of improved quality is exciting and challenging. This may just lead to a change in the way organizations operate in the future.