Configuration Management: The silver bullet for cost and schedule control[1]

Joseph Kasser DSc CEng CM MIEE
Systems Engineering and Evaluation Centre
University of South Australia
School of Electrical and Information Engineering F2-37
Mawson Lakes Campus
South Australia 5095
Telephone: +61 (08) 830 23941, Fax: +61 (08) 830 24723
Email: joseph.kasser@unisa.edu.au

Abstract. **This paper views the System Life Cycle from the perspective of Information Systems, the application of Knowledge Management and modern Quality theory. It shows how to reduce cost and schedule overruns by means of explicit emphasis on Configuration Management within the Cataract Approach to Build Planning in an integrated management-technical environment can reduce cost and schedule overruns and minimise project failures.**

I. INTRODUCTION

This paper shows how effective configuration control may be used to minimise the by project failures and cost and schedule overruns endemic in the current systems and software acquisition paradigm. Data from the USA [1] and UK [2] show that the problem is an international one. Efforts to overcome the problem have reacted to the effects of poorly articulated and changing user requirements during the development process and have focussed on changing the production process from the waterfall approach to some type of rapid, spiral or other methodology. However, from an information systems and Knowledge Management perspective the root cause of the cost and schedule overruns and project failures is not because the requirements change, it is a combination of

- the poor management of changing requirements during the System Life Cycle (SLC), and
- the lack of information needed for making informed decisions about the impact of the proposed changes.

The system and software development life cycle (SDLC) section of the SLC has evolved several methodologies since the early days of the Waterfall model. One of them, the Spiral Model [3] explicit emphasis on Risk Management. However, even with Risk Management and the current emphasis on process standards and capability maturity measurement, the developer working within the current production paradigm, cannot answer two simple questions posed by the customer during the SDLC [4] namely:

- "What do you mean, you can't tell me how much of my project has been completed?" [5].
- "What do you mean you can't tell me if my project is in trouble?" [6].

A. The key to effective control of the SLC

The key to effective control of the SLC is the Anticipatory Testing approach [7]. It can provide answers to the two questions, and thus by inference, cost-effective control of the SDLC. The Anticipatory Testing concept is a control and information system paradigm rather than a production paradigm. It views the SLC from the perspective of Information Systems, the application of Knowledge Management and modern Quality theory. It has explicit emphasis on Configuration Management and building Quality into the process. Reference [7] states that *the process, product and organisation represent three tightly coupled dimensions* and must not be considered independently. In addition, every one of the dimensions changes over time.

From the Anticipatory Testing perspective, the SLC is a time-ordered sequence of activities and can be considered as a series-parallel set of phased Builds (mini waterfalls or cataracts) [8] in a multi-threaded environment under the control of the Configuration Control Board (CCB). There are three threads of work in the SLC, management, development and test, sometimes known by other names including Quality. Reference [9] provides a description of the Cataract methodology that extends the Spiral Model by explicitly emphasising configuration control.

[1] This work was partially funded by the DSTO Centre of Expertise Contract

0-7803-7385-5/02/$17.00 ©2002 IEEE.

II. CHANGES

Reference [10] states that there are two types of changes during the SLC

- **Planned changes**. These tend to be changes that add capability.
- **Unplanned**. These tend to be changes that repair defects, but unplanned additional requirements are also a fact of life.

A. Change requests

The process for dealing with both types of change is the same and begins with a change request. The change requests are processed via the CCB. Requests for planned changes tend to be processed well before the change is to be implemented. Requests for unplanned changes however, need to be categorised by priority. Typical categories my be "routine", "urgent", or "do by yesterday" or their equivalents. A typical "do by yesterday" change request is the result of an analysis of a Discrepancy Report (DR) reporting that the system crashes.

B. The generic process for handling a change request

The generic process for handling a change request is shown in Figure 1. Some source generates a change request, which is logged and assigned an identification number. The impact of the requested change on the product and process (Builds) shown in Figure 2, is assessed and a decision made as to whether to accept or reject the request. The source is then notified of the decision, if the change request is accepted, then, if the configuration control process is fully operational:

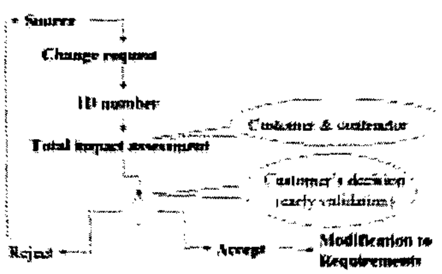

Fig. 1. The generic change request process

- From the product perspective, the affected requirements and all subsequent project documentation must be changed to reflect the new situation. This is done by adding, deleting or modifying (a combination of adding and deleting) requirements. The change may affect the capability of components at various levels of the design.

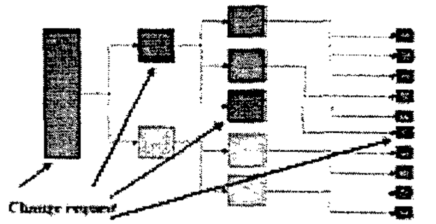

Fig. 2. The impact of a change request

- From the process perspective, the Build Plan must be changed to show when and where the change will be implemented by changing the affected elements of the Work Breakdown Structure (WBS). The cost and schedule impact will then be seen.
- The Systems Engineering Management Plan (SEMP) and Operations Concept Document (OCD) must be modified as appropriate.

However, in most current instances the configuration control process is defective and one or more of the steps listed above do not take place.

III. THE CONFIGURATION CONTROL PROCESS

Conventional configuration control tends to be limited to products that are either in process of construction or have been completed. However, the key to effective control of the production process is effective configuration control and informed decisions about the impact of any change request **on both the product** (cost and capability) **and the process** (cost and schedule). Thus the impact of the requested change on the process as represented by the Build Plan and WBS also needs to be assessed as well as the impact on the functionality or capability of the product under development.

IV. IMPACT ASSESSMENTS

A. The purpose of the impact assessment

The purpose must be to determine the cost, feasibility and risk of each requested change to both the product and process. The elements of the impact assessment are the

- Determination if the request has been rejected before and if those reasons are still applicable [9].
- Determination if the request has been accepted but not yet implemented [9].
- Determination if a conflict or contradiction exists with other requirements and resolve it.
- Determination if the requirement/change is really needed.

- Determination of the change in the total project risk on the schedule.
- Determination if the change is feasible.
- Estimation of the cost to implement the requirement/change.
- Determination of the cost drivers for the change in the design.
- Performing sensitivity analyses on the cost drivers. The results of a typical sensitivity analysis are shown in Figure 3.
- Discussing the cost drivers the results of the

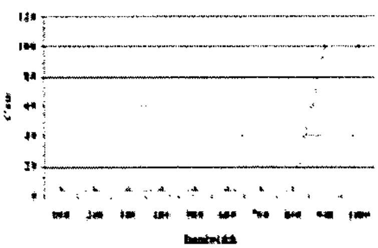

Fig. 3. Typical Sensitivity Analysis Graph

sensitivity analysis with the customer and determining if the cost drivers are really necessary. For example, in a given system, the requirement was for a communications bandwidth of 900 units. If the requirement had been accepted without the analysis, the cost to implement would have been estimated as 100 units of money. However, when presented with the graph shown in Figure 3, the customer was able to state that the bandwidth requirement of 900 had been based on building in excess capability on a needed bandwidth of 450. Thus lowering the bandwidth requirement to 800 would still provide more than the needed capability, thus reducing the costs by about 95%.

- Documenting the decisions in the system requirement repository.

B. Typical impact assessment questions

Typical impact assessment questions that may be used to guide the assessment are

- Why do we need the change?
- What if we don't accept it?
- What are the alternatives?
- What will the modified system do?
- How will the change contribute to mission objectives?
- How will the change impact existing and planned adjacent systems?
- What are the risks and their probability of occurrence?
- What are the required resources required for implementing the change?
- How long will it take to implement the change?

C. The analysis of the change request

The CCB will assign incoming change requests to an Integrated Product Process Team (IPPT). The IPPT will analyse the change and perform the impact assessment to the appropriate depth to minimise risks. The analysis phase of the impact assessment can be considered as the traditional design phase of the SLC. If the change request is considered as a requirement, then the process of meeting the requirement is the design of alternative solutions and the choice of the optimal solution, namely the analysis and synthesis functions of the SLC. The analysis is a problem solving and fixing exercise as illustrated in the feedback loop shown in Figure 4.

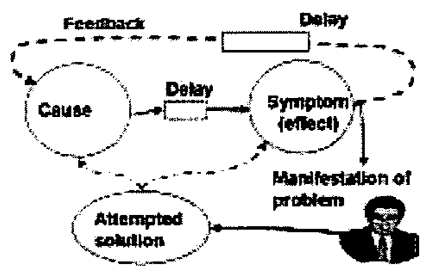

Fig. 4. Problem solving feedback loop

The problem is examined and analysed which results in one or more solutions being proposed. If the root cause of the problem is not found, a solution may not work or may only work for a short period of time. In addition, even if the implemented solution works it may introduce further problems that only show up after some period of time. Consider the implications of the time delay in the problem solving feedback loop. Any action has an effect in the present and in the future. Group these effects in time as:

- **First order** - noticeable effect within a second or less.
- **Second order** - noticeable effect within a minute or less.
- **Third order** - noticeable effect within an hour or less.
- **Fourth order** - noticeable effect within a day or less.

- **Fifth order** - noticeable effect within a week or less.
- **Sixth order** - noticeable effect within a month or less.
- **Seventh order** - noticeable effect within a year or less.
- **Eighth order** - noticeable effect within a decade or less.
- **Ninth order** - noticeable effect within a century or less.
- **Tenth order** – noticeable effect after a century or more.

The analysis of the requested change has to consider all of the above as applicable. While the higher order effects may not be applicable in a computer based system, they are applicable in long-lived systems such as those that affect the environment (Dams, power plants, etc.)

V. THE DECISION

When the impact analysis is complete, the project manager makes the final decision on the effect of a change. Two factors affect the decision, namely the recommendations of the IPPT (based on the result of the analysis) on the impact of the requested change on the cost, schedule and risk, and the willingness of the customer to accept the impact. The knowledge needed to perform both the analysis and make the decision is represented in the Quality System Elements (QSE) stored in a Framework for Requirements Engineering in a Digital Integrated Environment [4]. The information in the QSE includes but is not limited to the items shown in Table 1.

VI. THE SUITE OF TOOLS THAT COMPRISE THE SILVER BULLET

The suite of tools that comprise the Silver Bullet needed for configuration control in the Anticipatory Testing environment are a combination of several existing different and usually unconnected tools (eg. Requirements Management, Project Management, WBS, Configuration Control, and Cost Estimation, etc.), used in today's management and engineering work streams of the SLC. In addition the tools must also provide the capability to perform

- impact assessments of the effects of proposed changes before implementation,
- trade studies to compare the costs, risks, and capabilities of alternative designs.

VII. THE CONFIGURATION CONTROL BOARD

The CCB controls change. The acronym can also mean Change Control Board. The CCB must control all changes in the SDLC, those affecting both the process and the product. Thus project management must be a function of the CCB something that is not typically done in the current project management paradigm.

The elements of a typical CCB organisation are shown in Figure 5. Note that the project manager may be within the contractor or customer organisation depending on the time and place.

TABLE 1 THE FREDIE QUALITY SYSTEM ELEMENTS (QSE)

QSE	Purpose of individual QSE
Unique identification number	The key to tracking.
Requirement	The imperative statement containing both the required functionality and its corresponding Quality criteria or other form of representation.
Traceability to source(s) -	The previous level in the production sequence.
Traceability to implementation	The next level in the production sequence. Thus requirements are linked to design elements, which are linked to code elements, and so on.
Priority	Knowing the priority allows the high priority items to be assigned to early Builds, and simplifies the analysis of the effect of budget cuts.
Estimated cost and schedule	These feed into the management plan and are refined as the project passes through the SDLC.
The level of confidence in the cost and schedule estimates	These should improve as the project passes through the SDLC.
Rationale for requirement	The extrinsic information and other reasons for the requirement.
Planned verification methodology(s)	Developing this at the same time as the requirement avoids accepting requirements that are either impossible to verify or too expensive to verify.
Risk	Any risk factors associated with the requirement.
Keywords	Allow for searches through the database when assessing the impact of changes.
Production parameters	The Work Breakdown Structure (WBS) elements in the Builds in which the requirements are scheduled to be implemented.

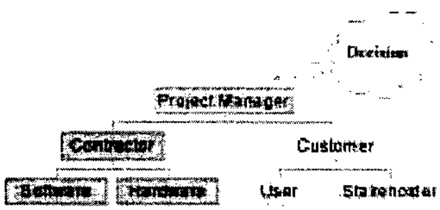

Fig. 5. Elements of a typical CCB

A. CCB composition

The customer may have a single representative or several, representing the user and stakeholders on the CCB. In government acquisitions or maintenance contracts the customer representative on the CCB could be the Contracting Officer's Technical Representative (COTR). The COTR knows the total contract budget and feeds the needs of the users and stakeholders to the CCB. The contractor can estimate the risk, cost and schedule impacts. The project manager then makes the decisions within the scope of the contract.

B. CCB decisions

All changes affect the cost of production. The factors that the CCB considers in making the decisions (accept or reject the change request) are different depending on the type of contract. In a cost-plus arrangement, the costs will be passed on to the customer in the form of additional or reduced costs depending on the nature of the change. In a firm fixed price (or design to cost) contract, if a requested change adds cost, then the contractor will absorb the cost, or the price will be increased, or some functionality will have to be removed to keep the costs fixed. This will probably be a lower priority requirement than the one being changed. In this situation, the CCB must ask the customer to decide which requirement to remove. This situation is one of the reasons for the "priority" element in the QSE.

VIII. IMPROVING THE CCB

The CCB for a single project may be improved in several ways including the following.

A. Adding test and evaluation capability

In many instances in government contracts, the government employs a separate contractor to perform the test and evaluation (T&E) or independent validation and verification (IV&V) of the development contractor's work. A more effective CCB is shown in Figure 6. Bringing the T&E representative onto the CCB at the start of the project means that

- The effect on the change on the T&E process will be part of the impact assessment.
- T&E (or IV&V) are brought into the project at the start, unlike in most of the current situations when they are brought in only after the defects have been designed into the system. This difference, in itself, is a major cost-reducing element, by allowing interdependent in-process testing and early detection of defects, ie., the Anticipatory Testing concept [7].

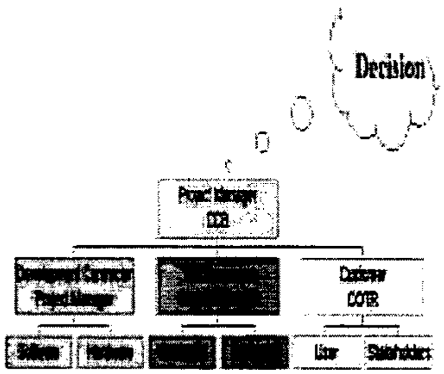

Fig. 6. A more effective CCB

B. The Customer single point interface

Figure 6 shows the COTR as a single interface to the customer. The contractor thus has only one customer to satisfy so there is a clean interface. This does not mean that the users and stakeholders do not need to be satisfied. The task of satisfying them remains wholly within the customer's organisation, thus truly providing only one person with the authority to authorise changes. Users and stakeholders may still be present in the CCB as part of the IPPT.

IX. LIFE CYCLE IMPLICATIONS

A. The recursive life cycle

Reference [9] shows that the change request process shown in Figure 1 is the same as the process for accepting the initial set of requirements at the start of the SLC. Thus the only difference between a requirement at the start-up phase and a change sometime later in the entire SLC is that a start-up is a transition from no system to some system, while a change is a transition from some system configuration to a different system configuration. Thus from this perspective the traditional design process is the impact assessment of the cost, schedule and risk of meeting the customer's need of a number of designs and the choice of the optimal design to meet the initial set of requirements. As such, the life cycle phases of the SLC are recursive. For example the feasibility analysis performed at the start of the SLC is functionally identical to the design processes later on, the difference being in the amount of information available at the time, the depth of the analysis, and the terminology used.

B. *The type and place of the IPPT in the SLC*

There are two types of IPPT in the SLC, permanent and temporary. Each contains the appropriate knowledge and skills necessary to complete its assignments. There are also two places for an IPPT in the SLC. The CCB is permanent IPPT and the impact assessments of change requests are performed by temporary IPPTs.

X. SUMMARY

By viewing the SLC from this perspective

- The CCB must perform the project management function on the process as well as on the product.
- The place of the IPPT in the SLC is easily recognised.
- The impact assessment activity (design) is generic throughout the SLC.

XI. CONCLUSIONS

Configuration management is not a silver bullet. However, when used as an integral part of the Cataract Methodology within an integrated process-product environment and the appropriate suite of tools, it does provide better control of, and information about, the state of, the SLC than other management approaches.

REFERENCES

[1] CHAOS, The Standish Group, *Chaos*, 1995. http://www.pm2go.com/sample_research/chaos_1994_4.asp last accessed January 16, 2002.

[2] OASIG, The performance of information technology and the role of human and organizational factors. Report to the Economic and Social Research Council, UK, 1996, available at http://www.shef.ac.uk/~iwp/publications/reports/itperf.html, last accessed January 16, 2002.

[3] Boehm B., "A Spiral Model of Software Development and Enhancement," *IEEE Computer*, May 1988.

[4] Kasser J.E. "A Framework for Requirements Engineering in a Digital Integrated Environment", *the Systems Engineering Test and Evaluation (SETE) Conference*, Brisbane, Australia, 2000.

[5] Kasser J.E., "What Do You Mean, You Can't Tell Me How Much of My Project Has Been Completed?", *INCOSE 7th International Symposium*, Los Angeles, CA, 1997.

[6] Kasser J.E., Williams V.R., "What Do You Mean You Can't Tell Me If My Project Is in Trouble?", *First European Conference on Software Metrics (FESMA 98)*, Antwerp, Belgium, 1998.

[7] Kasser J.E., *Applying Total Quality Management to Systems Engineering*, Artech House, 1995.

[8] Denzler D., Kasser J.E., "Designing Budget Tolerant Systems", *First International Symposium on "Reducing the Cost of Spacecraft Ground Systems and Operations"*, Rutherford Appleton Laboratory, England, 1995.

[9] Kasser J.E., "The Cataract Methodology for Systems and Software Acquisition", submitted to *the Systems Engineering Test and Evaluation (SETE) Conference*, Sydney, Australia, 2002.

[10] Donaldson S.E., Siegel S.G. *Cultivating successful Software Development*, Prentice Hall PRT, 1997.

AUTHOR

Joseph Kasser has been a practising systems engineer for over 30 years. As well as being a Certified Manager, he has a doctorate in Engineering Management (Systems Engineering). He is the author of "*Applying Total Quality Management to Systems Engineering*", Artech House, 1995, and many conference papers. Dr. Kasser performs research into improving the acquisition process. He is a recipient of NASA's Manned Space Flight Awareness Award for quality and technical excellence (Silver Snoopy), for performing and directing systems engineering and has many other awards (certificates and plaques) as well as letters of commendation from previous employers and satisfied customers. Dr. Kasser also teaches systems and software engineering subjects both in the classroom and internationally via distance education.