SOFTWARE VERIFICATION RESEARCH CENTRE

SCHOOL OF INFORMATION TECHNOLOGY

THE UNIVERSITY OF QUEENSLAND

Queensland 4072
Australia

TECHNICAL REPORT

No. 00-40

Extending the Integrated Capability Maturity Model (CMMI) for Safety-related Applications

Neil Robinson, Peter Lindsay, Adrian Pitman

December 2000

Phone: +61 7 3365 1003
Fax: +61 7 3365 1533

Note: Most SVRC technical reports are available via anonymous FTP, from svrc.it.uq.edu.au in the directory /pub/SVRC/techreports. Abstracts and compressed postscript files are available via http://svrc.it.uq.edu.au.

Extending the Integrated Capability Maturity Model (CMMI) for Safety-related Applications

Neil Robinson [a], Peter Lindsay [a], Adrian Pitman [b]

[a] Software Verification Research Centre, The University of Queensland, Queensland 4072, Australia
[b] Defence Materiel Organisation (DMO), Department of Defence, Australia

Abstract. The recently released CMMI offers a Capability Maturity Model integrated for software and systems engineering. The Australian Defence Force intends to use CMMI to assess suppliers of software intensive systems. A key aim is to identify the strengths and weaknesses of system and software suppliers, and to address identified weaknesses early in the acquisition process. This paper describes an extension of CMMI to explicitly deal with safety engineering, for use in assessment of suppliers of safety-related systems.

INTRODUCTION

Very few large software intensive development projects are successful first time. Many development projects across the world have suffered dramatic failures, at great financial cost to suppliers, their customers and the public. When safety is added to the equation, the consequences of failure are not just financial loss and inconvenience. In the worst cases, failure can mean loss of human life (Leveson 1995).

One approach to increasing project success rates is to use suppliers that can demonstrate a mature capability in system and software development. The Software Engineering Institute, Carnegie Mellon University, recently released CMMI: the Capability Maturity Model – Integrated for Systems Engineering/Software Engineering (CMMI 2000a). CMMI aims to provide guidance to organisations on how to improve their processes and their ability to manage development, acquisition and maintenance of products and services.

CMMI can also be used as a basis for assessment of organisations. With this in mind, the Australian Government's Defence Materiel Organisation (DMO), part of the Australian Defence Force, will adopt CMMI and its associated assessment methodology, SCAMPI (CMMI 2000b), as a key tool in their acquisition processes.

In the USA, the general approach to using CMMI-type models in acquisition has been to insist that suppliers attain an overall capability maturity level. This is often a prerequisite of contracts, for example. In Australia, the DMO intends to use CMMI instead as a risk-management tool, in which the capability of systems suppliers is assessed with respect to certain engineering processes. The aim is to address any identified weaknesses as part of the acquisition project programme by introducing project-risk mitigations, such as budgeting for process improvement in a specific areas.

However, the DMO recognises that CMMI is a generically structured framework, which requires amplification for specialised areas of software and systems engineering, such as safety engineering. Developing safety-critical systems is a high-risk activity and requires specialised skills and experience within an organisation. Hence, the DMO has initiated a technical study on Safety Capability Assessment, aimed at producing a safety extension to CMMI. The technical study is part of the 'DefSafe' project being undertaken by the Software Verification Research Centre at the University of Queensland, aimed at improving safety-critical system acquisition practices in the DMO.

The Australian Defence Force operates and maintains a diverse range of safety-critical and safety-related systems. A number of different system and software safety standards are used in acquisition projects, depending on the nature of the system and the acquisition. Such standards include US Mil-Std 882C, UK MoD standards 00-55 and 00-56, NATO STANAG 4404, Australian Def(Aust) 5679, and avionics software standard RTCA/DO-178B. The emerging civil functional-safety standard IEC 61508 is also expected to exert increasing influence on future acquisitions. See (Wabenhorst 1999) for an overview of the different safety standards.

This paper describes how we are extending CMMI to support capability assessment of proposed suppliers of safety-critical systems to the DMO. The extension is informed by the above standards and intended to operate with each of them individually or when taken in combination.

OVERVIEW OF THE STRUCTURE OF CMMI

We begin with an overview of CMMI before discussing how it is being extended. CMMI is presented in a document suite which contains training materials, an assessment method and the process model itself. The process model is available in two representations: 'staged' and 'continuous'.

The staged model is aimed at assessing the capability maturity level of an organisation as a whole. There are five maturity levels ranging from

"Performed" through to "Optimizing". Each maturity level identifies a number of process areas that the organisation should be implementing at that level. If the organisation can show that it is implementing the processes in a manner sufficient to fulfil the goals of those process areas then it can claim to be at that maturity level.

By contrast, the continuous model defines 'capability levels' for each process area. Capability levels range from "Not performed", through "Performed" and up to "Optimizing". The model can be used to develop a profile of the organisation's capabilities in different process areas. The model has been developed in such a way that key dependencies between process areas are respected. For example, it would be difficult to perform "Requirements Management" without developing requirements first.

While perceived as having a generally positive impact on suppliers' process improvement, capability maturity ratings are a coarse instrument and can be misleading (O'Connell 2000). In keeping with its purpose of using capability assessment as a risk-management tool, the DMO has decided to use the continuous model as the basis for its assessments.

The process areas in the CMMI continuous model are divided into four categories:

- Process Management
- Project Management
- Engineering
- Support

Under each of these categories are collections of Process Areas. A Process Area is a cluster of related Practices performed collectively to achieve a set of Goals (CMMI, 2000). For example, in the Engineering category the Process Areas are:

- Requirements Management
- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation.

Process areas contain:

- **Specific and Generic Goals** – These describe the purpose of the process area.
- **Specific and Generic Practices** – These describe activities which are considered important in achieving the goal that the practice is mapped to.
- **Sub-practices** – These provide additional guidance on meeting the Practices.
- **Introductory Notes**
- **Typical Work Products**
- **Generic Practice Elaborations**
- **Discipline Amplifications** – These provide additional guidance for particular disciplines. At present, discipline amplifications exist for system engineering and software engineering.

A SAFETY PROCESS MODEL

The challenge in extending CMMI for safety engineering was to find the optimum method of integrating the safety processes with the existing CMMI. Two fundamentally different approaches were investigated:

- A 'distributed' approach, in which the elements of safety management/engineering are distributed across the existing CMMI model
- A 'stand-alone' approach, whereby the CMMI model is extended with Process Areas specific to safety management/engineering goals.

In order to compare the two approaches we first developed a hierarchical 'safety process model' embodying the key elements of safety processes extracted from the different system and software safety standards in use at the DMO. The CMMI structure and the safety lifecycle of (IEC 61508, 1998) also influenced our model.

The structure of the safety process model is shown in Figure 1. High-level safety process elements were broken down hierarchically into finer elements (see Figure 2 for an example) and guidance was provided for bottom-level elements. For example, against the element which states that a systematic approach should be used for hazard identification, guidance is provided on suitable techniques, e.g. Hazard and Operability Studies (HAZOP) or Functional Failure Analysis (FFA).

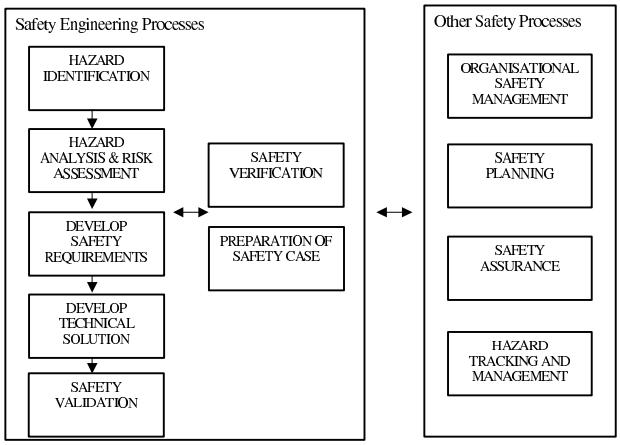

Figure 1 - Structure of Safety Process Model

- *Hazard Identification*
- …
  - *Identify all possible hazards*
  - *Use a systematic approach that includes consideration of all the phases of the system lifecycle*
  - *Use an appropriate model of the system as the basis for analysis*
    - *Involve appropriate people / organisations*

    …

Figure 2 - Example extract from Safety Process Model

In areas where the safety standards differ (for example in setting integrity levels), generic high-level guidance only was provided. This is in common with much of CMMI, which has been designed to be useable in a very wide range of organisations and industries.

INTEGRATING THE SAFETY PROCESS MODEL WITH CMMI

A 'representative subset' of the Safety Process Model was then mapped onto the two possible CMMI extension approaches.

For the distributed approach, the elements of the Safety Process Model were distributed across the CMMI model. This approach is in keeping with the CMMI definition of Discipline Amplifications, which states:

> *"As the models incorporate more disciplines, other types of discipline amplification will appear."*

For example, "Hazard Identification" was mapped to the CMMI process area "Risk Management", particularly in the Goals "Prepare for Risk Management" and "Identify and Analyse Risks". The safety elements were added as structured Discipline Amplifications, such that "Identify Hazards" becomes an Amplification against "Identify Risks". We have identified existing CMMI process areas that accommodate all the elements of the safety process model, except for "Preparation of Safety Case", which requires further consideration.

For the stand-alone approach, the aim is to make the safety extension stand on its own as far as possible. The Safety Process Model was converted into a set of stand-alone Process Areas specific to safety-related applications. For example, "Hazard Identification" becomes a Goal within a new CMMI Process Area "Safety Engineering" under the CMMI category "Engineering", "Identify all possible hazards" becomes a Practice, and lower level elements become Subpractices.

Both of these approaches initially assumed that the existing CMMI model would be consistent with best-practice safety engineering. However, in the course of the work some inconsistencies became apparent. For example in the CMMI Practice "Identify Risks" it states:

> *"To be effective, risk identification should not be an attempt to address every possible event regardless of how highly improbable it may be."*

In safety engineering however, the accepted approach is to identify all possible hazards/accidents and then at the next stage (hazard and risk analysis) assess their probability and severity. In this case, a Discipline Amplification is required to qualify the CMMI statement for safety-critical applications.

The distributed and stand-alone approaches were compared and evaluated against DMO requirements, using a process that included scoring the two approaches against weighted DMO requirements. As a result, the decision has been taken to adopt the 'stand-alone' approach, with additional Discipline Amplifications added to CMMI in instances where there is a possibility of conflict with best-practice safety engineering.

The choice of the stand-alone approach resulted partly from the DMO's desire to use the safety extension as soon as possible. The other key consideration was a concern with the distributed approach that safety might get lost in the larger CMMI assessment, since system and software engineering process assessors may not be familiar with safety engineering principles, and safety considerations may not receive the attention they deserve. Although the clear intention in CMMI is for additional disciplines to be accommodated through Discipline Amplifications, this does not seem scaleable. Adding the Safety Process Model onto the CMMI would add many detailed amplifications. Adding more disciplines would eventually result in a lack of structure and clarity. This issue and other related issues are discussed later in this paper.

THE SAFETY PROCESS MODEL IN CMMI FORM

The basic structure of the Safety Process Model in CMMI form is shown in Table 1. The elements of the safety model have been fitted into CMMI categories and two additional elements have been created: "Safety Management" and "Safety Engineering".

An excerpt from the safety process model in CMMI form is shown in Figure 3. This excerpt is a part of the goal "Hazard Identification / Preliminary Hazard Analysis" under the "Safety Engineering" process area. Further guidance will be added in the final version.

CMMI Categories	Safety Process Areas	Goals
Process Management	Organisational Safety Management	
Project Management	Safety Management	Safety Planning
		Hazard Tracking and Management
Engineering	Safety Engineering	Hazard Identification / Preliminary Hazard Analysis
		Hazard Analysis and Risk Assessment
		Develop Safety Requirements
		Develop Technical Solution
		Safety Verification
		Safety Validation
		Preparation of Safety Case
Support	Safety Assurance	

Table 1 Safety Process Model in CMMI Form

SP 1.1-4. Identify all possible hazards

It is important that the hazard identification is as complete as possible. The earlier hazards are identified the easier and more cost-effective it will be to deal with them.

Typical Work Products

Hazard and Operability Analysis (HAZOP) tables

Functional Failure Analysis (FFA) tables

Subpractices

1. Use a systematic approach that includes consideration of all the phases of the system lifecycle.

   Suitable systematic approaches include Hazard and Operability Analysis (HAZOP) and Functional Failure Analysis (FFA).

2. Involve appropriate people / organisations.
   - Involve people with relevant domain experience
   - Involve people with knowledge of all parts of the system lifecycle (e.g. commissioning, operation, maintenance)
   - Involve people with experience of hazard identification
   - Consider the effectiveness of teams involved in hazard identification.

4. Make use of available historic data.

   Historic data may provide information on past incidents / accidents. Checklists may be available for the specific domain.

Figure 3 - Excerpt from Safety Process Model in CMMI Form

THE INTEGRATED MODEL 'DISINTEGRATED'

The main input into the development of CMMI was the Software Capability Maturity Model (SW-CMM), released in 1991. This was highly influential on software development processes worldwide. However, as stated in (CMMI, 2000), a proliferation of maturity models followed, based on the same idea. Maturity models now exist for Systems Engineering, Software Acquisition, Human Resources Management, Software Testing and many more disciplines. These models overlap and are often inconsistent, which makes them very difficult to use together. The aim of CMMI is to integrate these models into a single unified framework which can be used at enterprise level, and which can "accommodate current and future models".

It was this stated aim of CMMI that led us to initially prefer the "distributed" approach (discussed above) to integrating our safety model with CMMI. However, integration of the models was difficult in practice.

To begin with, as one would expect, CMMI is distributed in a controlled form that does not allow changes. We could make a copy, subject to copyright agreements, and amend it to include further amplification for safety-related applications. But clearly, when the next version of CMMI is released we would have to redo our work. The preferred path would be to propose that our amplifications are added to the CMMI model in the next version. However, the DMO has an immediate need that we would like to satisfy.

Even if we could wait to have our amplifications added through the official routes, as stated earlier, the presentation of CMMI does not appear to allow for multiple, large discipline amplifications.

Finally, we observe that in most real projects safety engineering is performed as a specialist activity running alongside the rest of the design activities. Whilst it is vital that the safety activities are well synchronised with other parts of the project and that, for example, the models used in safety analysis are consistent with the current design baseline, in practice the level of integration is not high. For example, CMMI includes consideration of safety within Risk Management. This is one approach that could be taken and indeed the framework of Risk Management is extremely similar to the framework used for safety specific work. However, in practice business risk tends to be analysed separately from safety risk. So whilst integration of all engineering and management activities seems a laudable aim from the point of view of creating consistent maturity models, in practice we are not convinced that it is useful in terms of how organisations and projects should be structured.

We are however very conscious of the desire not to "disintegrate" the integrated model. We therefore intend to consider this issue again following trials of the stand-alone CMMI safety extension in early 2001.

FURTHER WORK

Currently we are adding guidance to the existing process model in preparation for the first DMO trial, scheduled for early 2001.

The next step for the technical study is to consider whether the techniques in SCAMPI are sufficient for use in safety-critical applications. There

is some concern that the SCAMPI techniques do not address the issue of thoroughness that is so important in safety-critical work. For example, specification is used in almost all software development, but in the highest integrity applications a more thorough technique – formal specification – is often mandated (MOD 00-55,) (AEA 5679, 1999). The study also intends to consider the related question of the relationship between 'Safety Integrity Levels' and CMMI Capability Maturity Levels.

A draft safety extension to CMMI will be produced in 2001, ready for trial by the DMO. This will be one of the first attempts to add a new discipline to CMMI. It will be interesting to see if the idea of an integrated CMM will be effective in practice, when extended beyond basic system and software engineering.

CONCLUSIONS

We have established in principle that an extension of CMMI for safety-related applications is feasible and have defined a large portion of the safety process model. The safety extension will be used in Australian trials in 2001 to specifically assess organisations' capability in safety-related applications. We have also identified a potential weakness in the ability of CMMI to be extended for additional disciplines.

REFERENCES

Australian Defence Standard Def(Aust) 5679, *The Procurement of Computer-based Safety Critical Systems,* 1998.

CMMI Product Development Team, "CMMI-SE/SW: Capability Maturity Model – Integrated for Systems Engineering/Software Engineering, Version 1.0 Continuous Representation", Software Engineering Institute technical report CMU/SEI-2000-TR-019, Carnegie Mellon University, USA, August 2000.

CMMI Product Development Team, "SCAMPI: Standard CMMI Assessment Method for Process Improvement: Method Description, Version 1.0", Software Engineering Institute technical report CMU/SEI-2000-TR-009, Carnegie Mellon University, USA, October 2000.

IEC International Standard 61508, *Functional safety of electrical / electronic / programmable electronic safety-related systems*, 1998.

Leveson, Nancy, *Safeware: System Safety and Computers*, Addison-Wesley, 1995.

O'Connell, E. and Saiedian, H. "Can You Trust Software Capability Evaluations?", *IEEE Computer*, vol. 33(2), pp 28-35, 2000.

NATO Standarization Agreement STANAG 4404, *Safety Design Requirements and Guidelines for Munition Related Safety Critical Computing Systems*, Edition 1.

RTCA DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, RTCA, 1992.

UK Ministry of Defence DefStan 00-55, *The Procurement of Safety Critical Software in Defence Equipment*, 1997.

UK Ministry of Defence 2nd Draft DefStan 00-56, *Safety Management Requirements for Defence Systems Containing Programmable Electronics*, 1996.

Wabenhorst, A. and Atchison, B, "A Survey of International Safety Standards", Software Verification Research Centre technical report 99-30, University of Queensland, Australia http://svrc.it.uq.edu.au/Bibliography/svrc-tr.html?99-30