The Capability Maturity Model® (CMM)

Introduction

The Capability Maturity Model® (CMM) was developed by Watts Humphrey of the Software Engineering Institute. In recent years it has become increasingly popular as a model for Software Process Improvement, providing a benchmark against which organisations can assess the capability level of their software process and identify the process improvement activities that will provide the most effective short term benefits.

The CMM is a 5 layer model that characterises the typical stages on the path to process improvement. The 5 maturity levels span from the ad hoc level, where few controls are in place and success is dependent on individual commitment and heroic efforts, to the optimising level, where the focus is on defect prevention and continuous improvement.

The model prioritises the improvements that are most relevant to an organisation of a particular maturity level by identifying a set of "Key Process Areas" that must be addressed before moving on to the next level. When collectively implemented, the Key Process Areas address the typical problems that an organisation at each maturity level would encounter.

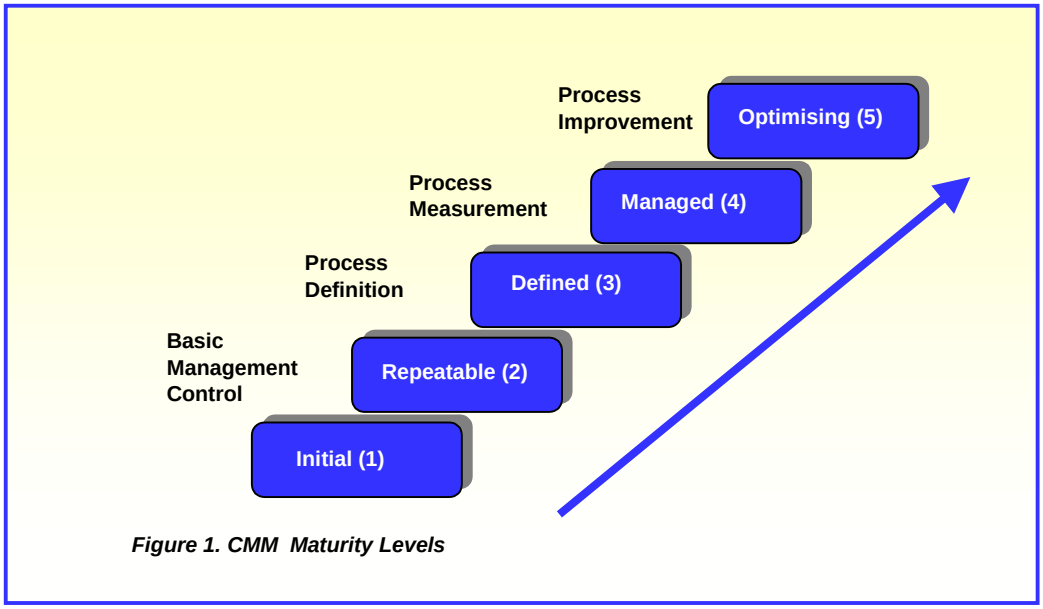

Figure 1. CMM Maturity Levels

Characteristics of each Maturity Level

Level 1 - Initial

Level 1 organisations are characterised as operating in an ad-hoc manner. Organisations are struggling with basic management issues. Problems are experienced in the areas of controlling changing requirements, cost estimation and planning, configuration management and the management and control of subcontractors. An organisation may have some procedures in place, but will have no formal mechanism to ensure that procedures are followed. At this level, an organisation may produce high quality products, but success is dependent on the skills and commitment of key individuals or "Heroes".

© Centre for Software Engineering, 1999
CMM® is registered in the U.S. Patent and Trademark Office

Level 2 - Repeatable

Level 2 is known as the repeatable level, where the success of a project is now dependent on an organisation's ability to learn from previous experiences of carrying out similar work, rather than on the skills and commitment of key individuals. Policies are established for managing the software project and procedures developed to implement these policies. Basic management controls are implemented on a project by project basis.

A level 2 organisation enjoys many benefits:

- As a result of effective project planning and tracking, a level 2 organisation is now in a stronger position to make and meet their estimates and plans.
- Size, cost and schedule estimates are based on historical data from previous projects.
- Software Requirements are baselined, ensuring improved control over changing requirements.
- Configuration Management is tightly controlled.
- Basic controls are in place for the management of sub-contractors.
- Software Quality Assurance provides management with the assurance that standards and procedures are being followed and are appropriate for use.

Simple, yet effective measurements are built up, which can be used as historical data for new projects e.g. size cost and schedule data (estimates v's actuals) number of change requests etc.

Level 3 - Defined

At Level 3, an organisation wide software process is now in place. A Software Process Engineering Group (SPEG) is established to implement the best practices from individual projects on an organisational level. Process information and best practices can now be shared among all members of an organisation, enabling people to learn from the experiences of other similar projects. The organisation wide standard software process is tailored for use by individual projects, based on the unique characteristics of their project. This tailored process is known as the defined software process.

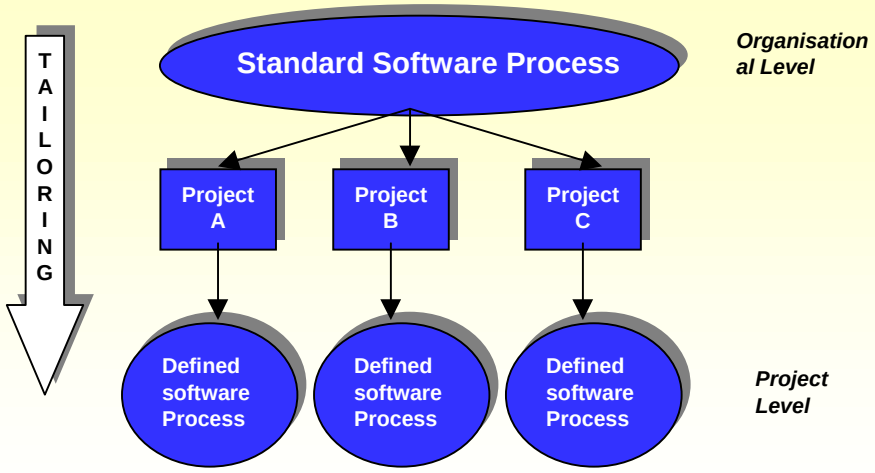

Figure 2: Implementing an organisation wide software process

Now that the basic management issues have been addressed at Level 2, an organisation is in a position to address the more technical software engineering issues. A suite of software engineering processes are introduced covering all phases in the software development life cycle, including requirements, design, coding and testing. A high emphasis is placed on the use of peer reviews as a disciplined approach to identifying and correcting errors early in the life cycle. An organisation wide training programme is introduced to ensure that all members of an organisation receive appropriate training to carry out their activities effectively.

A lot of valuable process data is gathered at level 3, and is now available on an organisational level e.g. Project Life Cycle Descriptions, Software Engineering procedures, results of Peer Reviews, Test Cases and Test Results, Problem Reports etc.

Level 4 - Managed

Significant gains in terms of quality and productivity are experienced at levels 2 and 3, through the introduction of best practice for management and engineering processes. The organisation wide software process introduced at level 3 forms the backbone for a comprehensive measurement programme which is introduced at level 4. Quantitative information gathered through the implementation of a project's defined software process is entered into an organisation wide software process database.

At level 4, process and product performance are measured and controlled within defined limits. When a limit is exceeded e.g. upper threshold of bugs found in a software product, appropriate action is taken.

Measurable goals are set for software product quality and actual progress towards achieving these goals is measured.

Level 5 – Optimising

At Level 5, the focus is on continuous improvement and process optimisation. The solid framework of effective processes and measurement data built up at the earlier stages provide a solid base for a highly proactive environment, where the focus moves towards defect prevention.

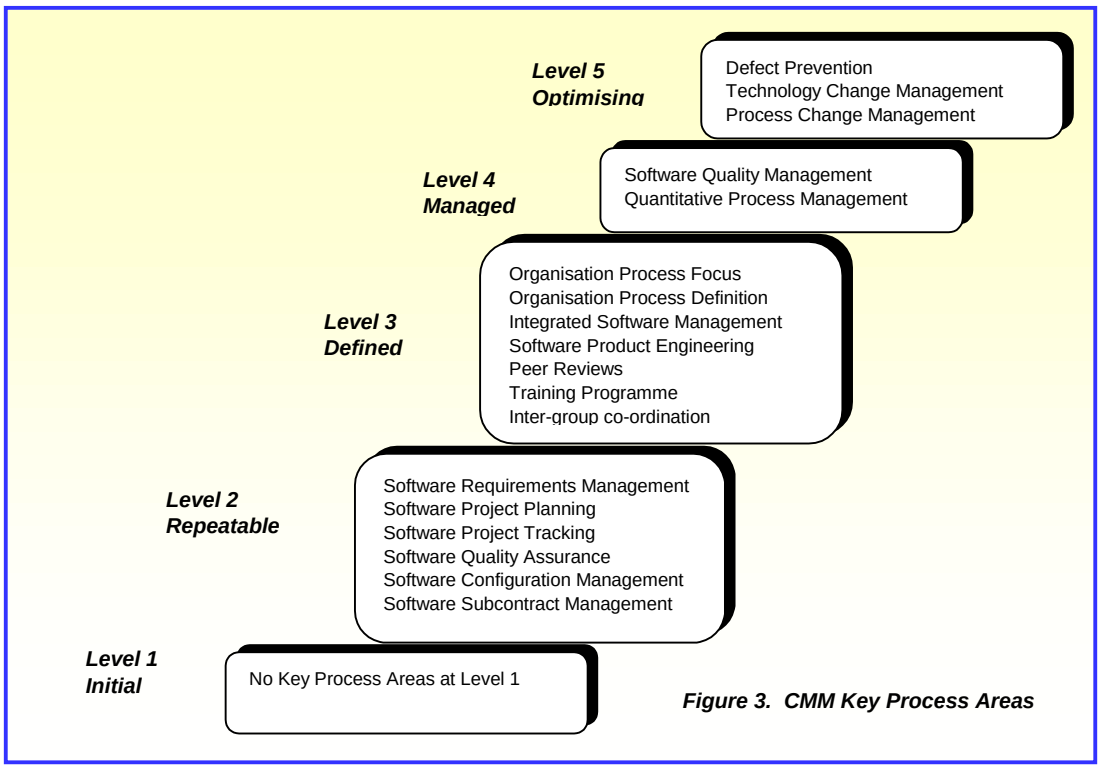

Figure 3. CMM Key Process Areas

Structure of the CMM

The CMM is composed of 5 layers, each layer representing the **maturity level** of an organisation's software process. To move from one maturity level to the next, a set of inter-related **key process areas** are identified, which represent the improvement steps that are most relevant to an organisation's process capability.

Each key process area (KPA) has a set of **goals**, containing short, simple statements which describe the essence of each KPA, indicating it's scope, boundaries and purpose. The achievement of all the goals ensures an effective implementation of a KPA.

The actual practices that describe the key process areas are organised into **common features**. These include the activities that must be carried out, and also the environment that must be in place to ensure that the activities are effectively implemented, or "institutionalised". Common features include:

Commitment to Perform:	Actions an organisation will take to ensure an effective and lasting process. These include the implementation of organisational policies and leadership.
Ability to Perform:	Preconditions that must exist to ensure that the process is implemented effectively and completely. These include adequate resourcing, organisational structures and training.
Activities Performed:	The actual activities that are carried out to implement a key process area.
Measurement and Analysis:	Basic measurements which should be collected as a process is implemented.
Verifying Implementation:	Actions that are taken to ensure that the process is being used effectively. These include reviews and audits by management and software quality assurance.

Each common feature contains a set of **key practices** which describe the activities and infrastructure required to implement the key process area. Each key practice is supported with sub-practices which provide further details on the implementation of a key practice.

Each category of the common features plays an important part in ensuring that the process is implemented effectively. A policy is issued by senior management indicating management commitment. An organisational structure is in place and adequate resources and funding are provided. In addition everyone receives adequate training to carry out their activities. As the process is implemented, basic measurements are collected and regular reviews are undertaken to ensure that the process is effective.

Using the CMM as a framework for Process Improvement

The CMM is becoming increasingly popular with many organisations as a model for process improvement. Based on the Watts Humphrey's theory:

"If you don't know where you are, a map won't help",

the first step is to identify the maturity level of an organisation's existing software process by carrying out a software process assessment. Based on the results of the assessment, a gap analysis is carried out, to identify the improvement actions that are most appropriate to the maturity level of an organisation, and a plan for process improvement is developed. Process improvement is not a once off effort. The CMM focuses on the concept of continuous improvement where an organisation continually strives to improve its software process. Based on information provided by the SEI, it can be seen from Figure 4 that most organisations are operating at the lower maturity levels, with the majority of organisations operating at levels 1 and 2. To date, few organisations have attained levels 4 or 5.

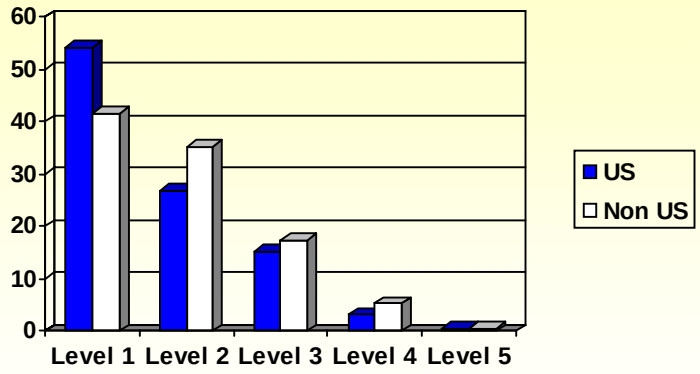

Figure 4. Percentage of organisations assessed at each level Source: SEI

New developments in the CMM

A new version of the CMM is scheduled for release by the end of 2001, which will provide an integration of the software CMM with other CMM Models e.g. Systems Engineering CMM. The new version will also address some new process areas, including validation, risk analysis and causal analysis.

Further Information:

The CSE provides a number training courses on the CMM Model. Our one-day introductory course, SPI approaches, provides an overview of and comparison between the 3 main approaches (ISO9001, CMM and SPICE/ISO 15504) assisting delegates in the selection of an approach which meets their business needs. We also provide a two-day course, Introduction to CMM, which provides an overview of the CMM Model, focussing on the Level 2 and Level 3 Key process Areas. In addition, we run a 9 month programme, SCATE (**S**mall **C**ompanies **A**ction **T**raining and **E**nabling), comprising a series of training workshops and individual mentoring support, to assist delegates in the implementation of CMM within their organisation.

In addition to our training programme, we carry out individual consultancy assignments, tailored to meet the individual needs of our clients. For further details of our services in this area may be obtained by contacting Jackie Berkery by e-mail at jackie@cse.dcu.ie, or by telephone at 01 7045612.

References

The Capability Maturity Model – Guidelines for improving the Software Process, Carnegie Mellon University, Software Engineering Institute, ISBN: 0-201-546647-7

Managing the Software Process - Watts S. Humphrey, ISBN: 0-201018095-2

A guide to the CMM - Understanding the Capability Maturity Model for software, Kenneth M. Dymond

SEI website: http://www.sei.cmu.edu

Technical Briefing Notes are issued on a range of software engineering topics as an aid to software developers, project leaders and managers. The intention is to provide a ‘status report’ on the state of the art (and/or the state of practice) in relation to particular aspects of software engineering. In addition they aim to highlight, where appropriate, a likely roadmap on a time horizon for future developments of the technology.

Centre for Software Engineering Ltd, Dublin City University Campus, Dublin 9, Ireland.
Telephone: +353 1 7045750 Fax: +353 1 7045605 Email: admin@cse.dcu.ie