

ULRR

Configuration management process improvement for the medical device industry

Item Type	Meetings and Proceedings
Authors	Mc Caffery, Fergal;Coleman, Gerry
Citation	6th International SPICE Conference on Process Assessment and Improvement;
Download date	2026-04-16 15:15:52
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/2248

Configuration Management Process Improvement for the Medical Device Industry

Fergal McCaffery
Irish Software Engineering Research Centre
(ISERC), University of Limerick.
Fergal.mccaffery@dkit.ie

Gerry Coleman
Dundalk Institute of Technology
gerry.coleman@dkit.ie

Abstract

This paper outlines how the goals, practices and capability levels for the configuration management (CM) process area within a software process improvement (SPI) framework have been developed. This framework addresses an opportunity to integrate the regulatory issues and SPI mechanisms so as to achieve improvements that are critical to the development of software for medical devices [1].

Software is becoming an increasingly important aspect of medical devices and medical device regulation. Medical devices can only be marketed if compliance and approval from the appropriate regulatory bodies of the Food and Drug Administration (FDA) [2] (US requirement), and the European Commission under its Medical Device Directives (MDD) [3] (CE marking requirement) is achieved. Integrated into the design process of medical devices, is the requirement of the production and maintenance of a device technical file, incorporating a design history file. Design history illustrates the well documented, defined and controlled processes and outputs, undertaken in the development of medical devices and for our particular consideration with this framework - the software components.

1. Introduction

Medical device companies base their software development processes on the need to comply with the FDA and European MDD regulations. To date, however, the primary focus has not been on the software processes but rather that ensuring the elements that satisfy the regulatory requirements are in place. As part of our research we sought the opinions of five medical device companies in N.Ireland in relation to the importance of having effective software processes in place. Each of the five companies expressed a desire to have access to a software process improvement framework that would

incorporate the goals of medical device software for meeting regulatory compliance.

2. SPI framework development

The Irish Software Engineering Research Centre (ISERC) is currently researching how software processes may be improved within the embedded software industry. As part of this research ISERC is aiming to develop a software development framework for the medical device sector that addresses existing regulatory requirements for the control of the design, development, maintenance and support of software. The approach for delivering the software development framework is to establish a model (implemented as illustrated in figure 1) that addresses the relevant regulations, and integrates those constraints within an SPI framework (i.e. Medical Device Software Process Improvement-MedeSPI).

The model will be flexible in that relevant elements of the SPI framework may be adopted as required to provide the most significant benefit to the business. The intention is to develop two frameworks, one that is based on ISO:15504 [4] and the other on the Capability Maturity Model Integration (¹CMMI[®]).

For the purpose of this paper, the SPI framework used will be that of the Capability Maturity Model Integration CMMI[®] [5] (as this work is at a more advanced stage) and the regulations used to extend the CMMI[®] framework will be those of the FDA and the ANSI/AAMI SW68:2001 standard (SW68) (Medical device software – Software life cycle processes) [6]

¹ CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University

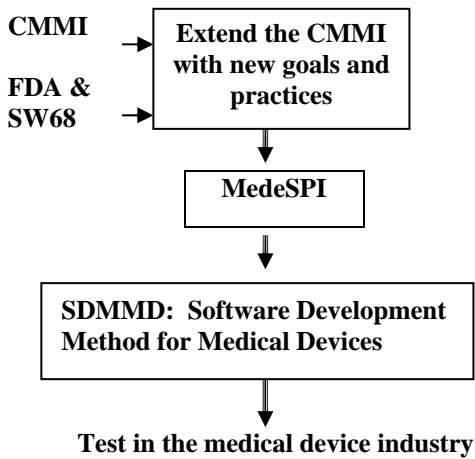


Figure 1: Software framework approach

The MedeSPI framework was initiated by work that one of the authors performed whilst performing research for the Centre for Software Process Technologies at the University of Ulster. This author is now progressing this work with the ISERC. The initial research work was assisted by the involvement of a steering group with a pilot of 5 medical device companies and a notified standards body (all based in N.Ireland). However, we now intend extending the steering group to include companies in the Republic of Ireland.

The Software Development Method for Medical Devices (SDMMD) will be a defined set of software process models (in effect a methodology) which when utilised will meet the goals of MedeSPI. SDMMD will cover the complete lifecycle. The project is divided into several stages.

1. Assess the need for and commitment to the creation of SDMMD and MedeSPI;
2. identify which parts of the CMMI[®] are required to comply with FDA regulation and extend the CMMI[®] with new goals and practices that are necessary to achieve FDA compliance (i.e. creation of MedeSPI);
3. Develop process models for meeting the goals of MedeSPI (i.e.create SDMMD);
4. Test SDMMD with Irish medical device companies.

We have completed stage 1 of this work and are currently performing stage 2 activities. In fact stage 2 has been performed for the risk management [7] and requirements management [8] process areas.

3. MedeSPI Development

SDMMD will provide a software development methodology, which addresses the regulatory guidance

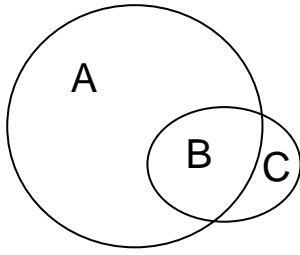
criteria, while introducing best practices that can be selected as required. MedeSPI will provide a means of assessing software engineering capability in twelve areas that have been defined by the FDA [9,10,11] as:

1. Level of Concern;
2. Software Description;
3. Device Hazard and Risk Analysis;
4. Software Requirements Specification;
5. Architecture Design;
6. Design Specifications;
7. Requirements Traceability Analysis;
8. Development;
9. Validation;
10. Verification and Testing;
11. Revision Level History;
12. Unresolved Anomalies & Release Version Number.

MedeSPI is being developed to promote SPI practices into the software development processes of medical device companies. This is an attempt to improve the effectiveness and efficiency of software processes used by medical device companies through investigating the mapping between twelve CMMI[®] process areas and the twelve FDA areas listed above. The twelve CMMI[®] process areas that we have deemed appropriate for the medical device industry are as follows:

1. Project Planning;
2. Project Monitoring & Control;
3. Supplier Agreement Management;
4. Risk Management;
5. Requirements Management;
6. Requirements Development;
7. Technical Solution;
8. Product Integration;
9. Verification;
10. Validation;
11. Configuration Management; I
12. Process and Product Quality Assurance.

The mappings between the FDA regulatory guidelines and the CMMI[®] process areas listed above then produce twelve MedeSPI process areas which retain the CMMI[®] process area names listed above. Each of the MedeSPI process areas will then be composed of a number of goals and practices. Goals and practices may be either generic (relating to the entire organisation) or specific (relating to the current process area). MedeSPI investigates what parts of the CMMI[®] process areas are required to satisfy FDA regulations, but also investigates the possibility of extending the CMMI[®] process areas with additional goals and practices that are outside the remit of CMMI[®], but are required in order to satisfy FDA regulations (see figure 2).



- A- CMMI Practices, not mandatory for FDA.
- B- CMMI Practices that are required for FDA.
- C- Non-CMMI Practices, mandatory for FDA.

Figure 2. Composition of the MedeSPI framework.

The model will help companies to measure their organisational capability and to track progression and achievements in each of the twelve process areas and against process capability levels. The MedeSPI framework has adopted the following capability levels:

- **Level 0** – In other SPI models level 0 indicates that a process is not performed. However, this is not the case within MedeSPI. Level 0 means that companies perform practices that enable them to adhere to medical device regulations. Companies must demonstrate that a process area satisfies the goals and performs the practices required to achieve FDA regulatory compliance. This will involve performing some practices which the CMMI® views as generic, although not to the extent of fulfilling any generic goals.
- **Level 1** - Companies must demonstrate that a process area satisfies level 0 and the CMMI® capability level 1 goal of performing the CMMI® base practices.
- **Level 2** – Companies must demonstrate that a process area satisfies level 1 and additionally performs CMMI® Advanced Practices, as well as the CMMI® capability level 2 generic goal of Institutionalising a Managed Process.
- **Level 3** - Companies must demonstrate that a process area satisfies level 2 and additionally the CMMI® Generic Goal to Institutionalise a Defined Process (CMMI® Generic Goal 3).
- **Level 4** – Companies must demonstrate that a process area satisfies level 3 and additionally the CMMI® Generic Goal to Institutionalise a Quantitatively Managed Process (CMMI® Generic Goal 4).
- **Level 5** - Companies must demonstrate that a process area satisfies level 4 and additionally the CMMI® Generic Goal to Institutionalise an Optimising Process (CMMI® Generic Goal 5).

Section 4 details a mapping of existing software development guidelines for the medical device industry

against the CMMI® for the CM process area. The FDA provides little insight into how CM should be performed other than to state that a CM plan should exist and that this should be adopted to manage configuration items for medical device software. Therefore in order to gain a greater understanding of the CM guidelines that medical device companies follow in order to achieve regulatory compliance we referred to the Medical device software-Software life cycle processes (SW68) standard. This standard was drafted for use in the medical device sector based on the lifecycle requirements of ISO/IEC 12207 [12].

4. Configuration management (CM)

CM can be defined as “the unique identification, controlled storage, change control, and status reporting of selected intermediate work products, product components and products during the life of a system” [13].

For many software companies, who report CM problems, it is the first major process weakness that they are required to address. For example, as the company expands, it must fulfil the task of acquiring new customers whilst satisfying the demands of existing customers. Often these demands include product customisations which many young companies, lacking reliable revenue streams, do not feel they can ignore. In many situations this results in companies having to support multiple code bases and product versions with very limited resources. Ultimately, a detailed CM process is the only way this problem can be solved.

A study of a small Danish software firm shows how it was forced to review the number of products it developed, and the amount of work it accepted, because of CM difficulties [14]. But CM is equally important in large software companies as a case study of Netscape and Microsoft’s development practices shows [15]. Therefore, in a software company or department without CM to control product development, there is no process to assess and no basis for measurement [[16].

To succeed in this area Humphrey [17] proposes that a CM plan be developed in conjunction with the establishment of a configuration control board to manage changes to all of the baseline configuration items and to ensure that configuration control procedures are followed.

There are four main activities to be considered as part of any CM strategy: *identification, storage, change control, & status reporting* [13].

Identification is concerned with uniquely categorising each *configuration item* and its relations to the outside world and other *configuration items*. A *configuration item* is an object, produced in several versions, which is placed under CM. *Storage* ensures that a *configuration item* will not disappear or be damaged, and can be readily retrieved as required. *Change Control* must properly manage any change requests and modifications made to products, and

provide traceability for any amendments to *configuration items*. *Status reporting* provides the information necessary to manage a product's development and maintenance. These issues are covered in detail under CMMI's CM process area.

4.1 CM mapping of FDA & SW68 against the CMMI

This section illustrates the MedeSPI capability structure for the CM process area. In order to achieve this, FDA regulations [2,9,10,11] & SW68 guidelines were mapped against the goals and practices of the CMMI[®] CM process area. We did not find any additional goals or practices (not included in the CMMI[®]) that would have to be added in order to satisfy the FDA and SW68 guidelines.

CM has three specific goals (SG). These are as follows: **SG1: Establish Baselines**, **SG2: Track and Control Changes** & **SG3: Establish Integrity**. To meet each of these goals it is necessary for a number of specific practices to be performed.

4.1.1 SG1: Establish Baselines. In order to fulfil **SG1: "Establish Baselines"** the following specific practices have to be performed: *1.1-1 Identify Configuration Items*, *1.2-1 Establish a CM System*; & *1.3-1 Create or Release Baselines*.

4.1.1.1 Identify Configuration Items (1.1-1). *Identifying configuration items* involves demonstrating that the sub-practices listed in Table 1 are being performed.

Table 1, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI[®] SP 1.1-1. Therefore, in order to adhere to SW68 and FDA guidelines only 3 out of the 5 activities required for CMMI[®] SP 1.1-1 (*Identifying Configuration items*) are necessary (that is those indicated as level 0 sub-practices)

Table 1: MedeSPI capability levels for identifying configuration items

<i>Sub-Practice</i>	<i>Level</i>
Select the configuration items and the work products that compose them based on documented criteria.	0
Assign unique identifiers to configuration items.	0
Specify the important characteristics of each configuration item.	1
Specify when each configuration item is placed under CM.	0
Identify the owner responsible for each configuration item.	1

4.1.1.2 Establish a CM System (1.2-1). *Establishing a CM system* involves demonstrating that the sub-practices listed in Table 2 are being performed.

Table 2, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI[®] SP 1.2-1. Therefore, in order to adhere to the FDA and SW68 guidelines, only 2 of the 8 activities required for CMMI[®] SP 1.2-1 (*Establishing a CM system*) are necessary. The main differences are that CMMI[®] requests the usage of multiple control levels of CM, as well as archiving and restoration procedures to be in place.

Table 2: MedeSPI capability levels for establishing a CM system

<i>Sub-Practice</i>	<i>Level</i>
Establish a mechanism to manage multiple control levels of CM.	1
Store and retrieve configuration items in the CM system.	0
Share and transfer configuration items between control levels within the CM system.	1
Store and recover archived versions of configuration items.	1
Store, update, and retrieve CM records.	0
Create CM reports from the CM system.	1
Preserve the contents of the CM system.	1
Revise the CM structure as necessary	1

4.1.1.3 Create or Release Baselines (1.3-1). *Creating or releasing baselines*, involves demonstrating that the sub-practices listed in Table 3 are being performed.

Table 3, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI[®] SP 1.3-1. Therefore, in order to adhere to the FDA and SW68 guidelines only one of the 4 activities required for CMMI[®] SP 1.3-1 (*Creating or releasing baselines*) are necessary.

Table 3: MedeSPI capability levels for creating or releasing baselines

<i>Sub-Practice</i>	<i>Level</i>
Obtain authorisation from the CCB before creating or releasing baselines of configuration items	1
Create or release baselines only from configuration items in the CM system	1
Document the set of configuration items that are contained in a baseline	0
Make the current set of baselines readily available	1

4.1.2 SG2 Track and Control Changes.

4.1.2.1 Track Change Requests (2.1-1). *Tracking change requests*, involves demonstrating that the sub-practices listed in Table 4 are being performed.

Table 4, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI® SP 2.1-1. Therefore, in order to adhere to the FDA and SW68 guidelines all of the activities required for CMMI® SP 2.1-1 (*Tracking change requests*) are necessary, but not always to the same level of detail.

Table 4: MedeSPI capability levels for tracking change requests

<i>Sub-Practice</i>	<i>Level</i>
Initiate and record change requests in the change request database	0
Analyse the impact of changes and fixes proposed in the change requests.	0
Review change requests that will be addressed in the next baseline with those who will be affected by the changes and get their agreement.	0
Track the status of change requests to closure.	0

4.1.2.2 Control Configuration Items (2.2-1). *Controlling configuration items*, involves demonstrating that the sub-practices listed in Table 5 are being performed.

Table 5, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI® SP 2.2-1. As the control of configuration items is very important in terms of ensuring the integrity of medical device software it is therefore no surprise that 4 of the 5 activities required for CMMI® SP 2.2-1 (*Controlling configuration items*) are necessary in order to adhere to the FDA and SW68 guidelines.

Table 5: MedeSPI capability levels for controlling configuration items

<i>Sub-Practice</i>	<i>Level</i>
Control changes to configuration items throughout the life of the product.	0
Obtain appropriate authorisation before changed configuration items are entered into the CM system.	0
Check in and check out configuration items from the CM system for incorporation of changes in a manner that maintains the correctness and integrity of the configuration items.	1
Perform reviews to ensure that changes have not caused unintended effects on the baselines	0
Record changes to configuration items and the reasons for the changes as appropriate	0

4.1.3 SG 3 Establish Integrity

4.1.3.1 Establish CM Records (3.1-1). *Establishing CM records*, involves demonstrating that the sub-practices listed in Table 6 are being performed.

Table 6, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within

CMMI® SP 3.1-1. The process of establishing CM records is very important in terms of providing the traceability evidence that is required to meet the regulatory requirements associated with medical device software. Therefore, it is no surprise that most of the activities (3 out of 5) required for CMMI® SP 3.1-1 (*Establishing CM records*) are necessary in order to adhere to the FDA and SW68 guidelines.

Table 6: MedeSPI capability levels for establishing CM records

<i>Sub-Practice</i>	<i>Level</i>
Record CM actions in sufficient detail so the content and status of each configuration item is known and previous versions can be recovered.	0
Ensure that relevant stakeholders have access to and knowledge of the configuration status of the configuration items.	1
Specify the latest version of the baselines.	1
Identify the version of the configuration items that constitute a particular baseline.	0
Describe the differences between successive baselines	1
Revise the status and history of each configuration item as necessary	0

4.1.3.2 Perform Configuration Audits (3.2-1).

Performing configuration audits, involves demonstrating that the sub-practices listed in Table 7 are being performed.

Table 7, shows the mapping between the SW68 and FDA guidelines for each of the sub-practices within CMMI® SP 3.2-1. In terms of the mapping of SW68 process management guidelines against the CMMI® guidelines for CM this specific practice in CMMI® has no equivalent practice within the SW68 documentation. The SW68 CM documentation does not provide any details on the need for auditing the CM processes and activities.

Table 7: MedeSPI capability levels for performing configuration audits

<i>Sub-Practice</i>	<i>Level</i>
Assess the integrity of the baselines	1
Confirm that the configuration records correctly identify the configuration of the configuration items.	1
Review the structure and integrity of the items in the CM system.	1
Confirm the completeness and correctness of the items in the CM system.	1
Confirm compliance with applicable CM standards and procedures.	1
Track action items from the audit to closure.	1

4.1.4 The Generic goals and practices of CM. The CMMI® identifies a number of generic goals and practices. At a fundamental maturity or capability level it is only necessary to perform the specific base practices. It is

interesting to note that the SW68 guidelines for CM often have a counterpart in the CMMI[®]. For configuration management the generic goals and practices for capability level 2 (**GG 2: Institutionalise a Managed Process - IMP**) are:

- **GP 2.1 Establish Policy;**
- **GP 2.2 Plan the process;**
- **GP 2.3 Provide Resources;**
- **GP 2.4 Assign Responsibility;**
- **GP 2.5 Train People;**
- **GP 2.6 Manage Configurations;**
- **GP 2.7 Identify stakeholders;**
- **GP 2.8 M&C Process;**
- **GP 2.9 Evaluate Adherence;**
- **GP 2.10 Review.**

Table 8, illustrates what CM goals, practices and sub-practices have to be performed for each of the MedeSPI capability levels.

Table 8: MedeSPI CM generic process area components and capability levels

Goal: GG2: Institutionalise a Managed Process	
Practice	Level
GP 2.1 Establish Policy	0
GP 2.2 Plan the process	0
GP 2.3 Provide Resources	0
GP 2.4 Assign Responsibility	0
GP 2.5 Train People	0
GP 2.6 Manage Configurations	2
GP 2.7 Identify stakeholders	2
GP 2.8 M&C Process	2
GP 2.9 Evaluate Adherence	2
GP 2.10 Review	2
Goal: GG3: Institutionalise a Defined Process	
Practice	Level
GP 3.1 Establish a defined Process	3
GP 3.2 Collect Improvement Information	3
Goal: GG4: Institutionalise a Quantitatively Managed Process	
Practice	Level
GP 4.1 Establish Quantitative Objectives for the Process	4
GP 4.2 Stabilise Sub-process Performance	4
Goal: GG5: Institutionalise an Optimising Process	
Practice	Level
GP 5.1 Ensure Continuous Process Improvement	5
GP 5.2 Correct Root Causes of Problems	5

The FDA regulations state that each manufacturer shall establish the appropriate responsibility, authority, and interrelation of all personnel who manage, perform, and assess work affecting quality (this would relate the

CMMI[®] generic goal associated with assigning responsibility - GP 2.4). It also undertakes to ensure that all work is adequately resourced and that staff are trained (this sentence relates the CMMI[®] generic goals associated with providing resources and training - GP 2.3 & GP 2.5). Additionally, the guidelines specify that a company should have a policy and a process in place for CM (GP 2.1 & GP 2.2). However, it should be acknowledged that these linkages are made at a very high level as the FDA specifies little detail other than a very generic sentence in relation to these areas, and no specific guidance as to how these activities may be achieved.

5. Conclusions.

With respect to the specific goals and practices of the CM process area, it is clear that following FDA and SW68 guidelines will only, at best, partially meet the goals of this CMMI[®] process area. As might reasonably be expected, there is little guidance provided within the FDA regulations for the advanced practices of CM. Since failure to perform any specific practice implies failure to meet the specific goal, with respect to CMMI[®], it is clear, that the goals of CM cannot be obtained by satisfying FDA and SW68 guidelines during software development. But is the opposite true, can meeting the CMMI[®] goals for CM successfully meet FDA and SW68 guidelines? Certainly for CM, meeting the goals of the process area by performing the specific practices would in general more than meets the FDA and SW68 guidelines in this area. For CM, the existing CMMI[®] specification of goals and practices can be carried over, with the extension mentioned above into the MedeSPI framework.

This research work will now be extended to develop capability models for the the process areas of risk management, validation and verification within the medical device domain.

6. Acknowledgements.

This research is supported by the Science Foundation Ireland funded project, Global Software Development in Small to Medium Sized Enterprises (GSD for SMEs) as part of the Irish Software Engineering Research Centre - ISERC (<http://www.iserc.ie>).

7. References

- [1]. F. Mc Caffery , P. Donnelly, A. Dorling & F.G. Wilkie. (Apr 2004) "A Software Process Development, Assessment and Improvement Framework for the Medical Device Industry", Proceedings of [Fourth International SPICE Conference on Process Assessment and Improvement](#), Lisbon, Portugal, 28-29 April 2004, SPICE User Group (Lisbon, Portugal), ISBN 972-9071-73-X, Pages 100-109.

- [2]. FDA Regulations. "Code of Federal Regulations 21 CFR Part 820." June 1997.
- [3]. European Council, "Council Directive 93/42/EEC Concerning Medical Devices", 14 June 1993.
- [4]. ISO/IEC 15504: Information Technology – Process Assessment – Part 5: An exemplar Process Assessment Model, ISO/IEC JTC1/SC7, October 2003.
- [5]. Capability Maturity Model[®] Integration^{CM} (CMMISM) for Software Engineering (CMMI-SW, V1.1, Version 1.1, August 2002)
- [6]. Medical device software-Software life cycle processes, ANSI(American National Standard)/AAMI (Association for the Advancement of Medical Instrumentation) SW68:2001.
- [7]. F. Mc Caffery, D. McFall, P. Donnelly , F.G. Wilkie "Risk Management Process Improvement for the medical device industry". Proceedings of the International Conference on Software Development (SWDC-REK-2005) , University of Iceland, , 27th May - 1st June, 2005, in "Software Development", Pages 92-103
- [8]. F. Mc Caffery, D. McFall, P. Donnelly , F.G. Wilkie & R. Sterrit. "A Software Process Improvement Lifecycle Framework for the Medical Device Industry" , in Proceedings of the 12th Annual IEEE International Conference on the Engineering of Computer Based Systems (ECBS 2005),Greenbelt Marriott Hotel, Greenbelt, Maryland, USA (Washington DC metro), 3-8 April, 2005, Pages 273-280. ISBN 0-7695-2308-0
- [9]. FDA/CDRH Guidance Document. "Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices." May 1998.
- [10].FDA/CDRH Guidance Document. "Guidance for Off-the-Shelf Software Use in Medical Devices." September 1999
- [11].FDA/CDRH Guidance Document. "General Principles of Software Validation." June 1997.
- [12].ISO/IEC 12207:1995, Information technology – Software lifecycle processes
- [13].A.M. J. Hass, 2003, *Configuration Management Principles and Practice*, Addison Wesley.
- [14].R. Baskerville and J. Pries-Heje, 1999, 'Knowledge Capability and Maturity in Software Management', in *The Data Base for Advances in Information Systems*, Spring, Vol. 30, No. 2, 26-43.
- [15].M. A. Cusumano, and D.B. Yoffie, 1999, 'Software Development on Internet Time', in *IEEE Computer*, Vol. 32, No. 10, Oct., 60-69.
- [16].M.E. Fayad and M. Laitinen, 1997, 'Process Assessment Considered Wasteful', in *Communications of the ACM*, Vol. 40, No. 11, 125-128.
- [17].W. S. Humphrey, 2000, *Introduction to the Team Software Process*, Addison Wesley

SM SCAMPI is a service mark of Carnegie Mellon University

