Competing Values in Software Process Improvement: An Assumption Analysis of CMM From an Organizational Culture Perspective

Ojelanki Ngwenyama and Peter Axel Nielsen

***Abstract*—The capability maturity model (CMM) approach to software process improvement is the most dominant paradigm of organizational change that software organizations implement. While some organizations have achieved various levels of success with the CMM, the vast majority have failed. In this paper, we investigate the assumptions about organizational culture embedded in the CMM models and we discuss their implications for software process improvement (SPI) initiatives. In this paper, we utilize the well-known competing values model to surface and analyze the assumptions underlying the CMM. Our analysis reveals contradictory sets of assumptions about organizational culture in the CMM approach. We believe that an understanding of these contradictions can help researchers address some of the difficulties that have been observed in implementing and institutionalizing SPI programs in organizations. Further, this research can help to open up a much-needed line of research that would examine the organization theory assumptions that underpin CMM. This type of research is important if CMM is to evolve as an effective organizational change paradigm for software organizations.**

***Index Terms*—Capability maturity model (CMM), competing values model, organizational culture, software engineering, software process improvement (SPI).**

I. Introduction

DURING THE LAST decade, software process improvement (SPI) has emerged as the dominant approach for improving quality and productivity in software development organizations. Inspired by the work of Humphrey [15]–[17], a large body of knowledge on SPI has become available including specific models such as SPICE [7], the European bootstrap model [22], the capability maturity model (CMM) [33]–[35], quality improvement process [27], and quality software management [49]. The CMM and its SPI implementation methodology, IDEAL, are the most widely known and are used by software companies all over the world. The theoretical foundations of SPI approaches (specifically CMM) are rooted in the technical perspectives of cybernetics and total quality management. Ever since its first presentation, CMM has been extremely influential on software engineering practices around the world. The model has served as a framework for software process and quality improvement efforts in thousands of software organizations and the resources expended on CMM-based SPI are in the billions of dollars [9]. Despite the large investments of resources, the failure rate for SPI programs is high—too high many would say. The most recent report from the Software Engineering Institute puts the rate of failure at around 70% [45]; a prior report [44] showed equally dim results.

There are several possible explanations for the high rate of failure. Several researchers have suggested that CMM does not effectively deal with the social aspects of organizations. Johansen and Mathiassen [18] argue that CMM needs a more managerial focus. Nielsen and Nørbjerg [31] argue that CMM needs to be supplemented with socially oriented theories in order to address organizational change issues and organizational politics. Aaen *et al*. [51] argue that the scale and complexity of the organizational change proposed by CMM necessitates a managerial rather than technical approach. We agree that the CMM-SPI paradigm lacks an awareness of the social nature of organizations, but we also believe that assumptions about organizational culture embedded in CMM constitute a fundamental issue. In general, the CMM–SPI paradigm holds a rational and mechanistic view of organizations. The risk is that this mechanistic view reduces software organizations to little more than input–output processes governed by technical rules. The primary objective of the CMM is to achieve "optimal repeatable processes for software development" [32], [33]. Although the SPI paradigm is an attempt to change how software professionals think and act in their everyday organizational activity [50], researchers and proponents of SPI have yet to incorporate the organizational culture perspective in their work. The fundamental conjecture of this paper is that SPI is an intervention in the organizational culture with the objective of changing it. In this regard, SPI theory and practice cannot ignore the body of knowledge about organizational culture. As Lundberg puts it: "Organizational culture determines much of what we can do as we attempt to manage change" [24].

In this paper, we investigate the core assumptions about organizational culture embedded in the CMM models. Our motivation for conducting this study stems from our experiences in longitudinal (1997–1999) studies of CMM–SPI implementation conducted in four companies. During these studies, it became clear to us that contradictory assumptions within the CMM models were presenting difficulties to the implementation teams. We believe that a clear understanding of the core assumptions of CMM–SPI

Manuscript received October 13, 2001; revised June 5, 2002. Review of this manuscript was arranged by Special Issue Editors M. G. Martinsons and R. Davison.

O. Ngwenyama is with the Department of Information Systems, Virginia Commonwealth University, Richmond, VA 23284-4000 USA (e-mail: ojelanki@comcast.net).

P. A. Nielsen is with the Department of Information Systems, Virginia Commonwealth University, Richmond, VA 23284-4000 USA and also with the Department of Computer Science, Aalborg University, DK-9220 Aalborg, Denmark.

Digital Object Identifier 10.1109/TEM.2002.808267

can help both researchers and implementers understand the limitations of the paradigm. The research presented here can help researchers to reconceptualize the CMM–SPI models in order to address not only the cultural contradictions, but also blind spots in organizational change that inhibit successful implementation. It can also help managers using current CMM models to anticipate implementation problems and to design solution strategies for overcoming them. In this paper, we utilize the well-known competing values model to surface and analyze the assumptions underlying the CMM. Our analysis uncovers two distinctly different and contradictory sets of assumptions about organizational culture in the CMM approach. We believe that these contradictions can lead to significant problems in implementing and institutionalizing SPI programs in organizations. The rest of the paper is organized as follows. In Section II, we review the basic concepts of organizational culture and outline the framework and method for our analysis. We use this in an analysis of the assumptions in the CMM about organizational culture. In Section III, we present the research findings. In Section IV, we discuss our findings and conclude with some ideas for future research into CMM-based SPI.

II. Organizational Culture

Our primary objective is to surface and analyze the assumptions about organizational culture that are embedded in the CMM. Some of these assumptions are explicitly addressed in the CMM, while others are not explicated and we need to go beyond the claims made by its developers and proponents. To assist in surfacing the assumptions, we look particularly for inherent contradictions and hidden meanings in the CMM texts. Specifically, we are interested in surfacing contradictory assumptions about organizational culture present in the text. We use the competing values in organizational culture framework as a lens for examining the CMM documents. Before we outline the competing values framework, we will briefly overview some basic concepts of organizational culture.

Although there is a growing body of literature on organizational culture, there is no formal definition of the term [14], [43]. Definitions of organizational culture range from mentalist (mental models) to social constructivist (social structures) views. Hofstede, for example views organizational culture as "the collective programming of the mind which distinguishes the members of one organization from another" [13]. Louis views organizational culture as "the tacit, shared, and coherent understandings among members about who and what matters; how, what, and why things get done as they do" [23]. Others view organizational culture as social structures (symbols, norms, shared meanings) that influence an actor's lived experience and sense-making about organizational realities [12], [28], [30], [37], [46]. Although there are different conceptualizations of organizational culture, most authors agree that it is the basis upon which organizational actions are constructed and enacted [1], [13], [36], [43].

Smircich [46], for example, suggests that organizational culture can be viewed as a cognitive-social structure that is partly embedded in the minds of organizational members and partly externalized in specific symbolic activities, shared values, norms, and understandings. Such structures are built up over time via socialization and lived experience in the organization and society [2]. From this perspective, organizational culture is viewed as cognitive and social structures that circumscribe and determine the potential and the options in organizational action. Other researchers view organizational culture as the ground of organizational action and its manifestations as various levels of organizational reality. Schein [42], [43] distinguishes three levels of organizational culture: artifacts, values, and underlying assumptions. In Schein's conceptualization [43, p. 252], *artifacts* are "visible organizational structures and processes;" *values* are "strategies, goals and philosophies;" and *underlying assumptions* are "unconscious, taken for granted beliefs and habits of perception, thought and feeling. " Lundberg [24] also distinguishes between three levels of organizational culture: manifest, strategic, and core. In Lundberg's conceptualization, the *manifest* level is composed of "symbolic artifacts, language, stories, ritual activity, and patterned behavior." The *strategic level* is composed of strategic beliefs. The *core level* is composed of ideologies, values, and assumptions. Schein's conceptualization has been criticized for its shallowness and augmented by other theorists such as Marcoulides and Heck [26] and Grundy and Rousseau [8]. It is generally agreed that what is most visible in organizational culture are "symbols of identification," such as logos, ritualistic activity, patterns of behavior, and communication (jargon, slogans, etc.). These symbols are simply a reflection of deeper levels of culture, such as core values, ideologies, and assumptions [40], [41]. Some theorists have suggested that the stronger the integration between layers of culture, the stronger the culture is [6], [36] and the more difficult to change [24].

A. Framework for Analysis

In this research, we take the view that organizational culture can be interrogated via *organizational artifacts*, such as: 1) visible organizational structures and processes; 2) values and underlying assumptions; and 3) symbols [24], [40], [41], [43]. We are not suggesting that organizational culture is static—we take the view that organizational culture is emergent. We view the cultural process as a continual enactment within a context of cognitive and social structures that circumscribe and determine the potential and options in organizational action. We share Smircich's [46] view that organizational culture is a cognitive-social structure. On the manifest level, organizational structures, processes, and symbols are carriers of organizational culture. On the core level, organizational members share beliefs, values, and understandings, which guide their actions.

We have selected the competing values framework of Quinn and McGrath [38] and its extensions as the lens through which we examine the artifacts of the CMM paradigm. The reasons for choosing this framework are as follows. First, the framework focuses on the problems of organizational change and this is certainly relevant to understanding SPI. Second, the framework focuses on how the values of the different schools of organization theory are embodied in management practice and through that we can analyze the organization theories that underpin the CMM. Third, the competing values framework provides the lens through which we can observe and analyze the contradictions of

TABLE I
COMPETING VALUES IN ORGANIZATIONAL CULTURE [38]

Aspect	Hierarchical	Rational	Consensual	Developmental
Organizational orientation	Stability and control	Productivity and efficiency	Cohesion and morale	Flexibility, adaptability and readiness
Organizational objectives	Execution of regulations	Pursuit of objectives	Group maintenance	Growth and development
Organizational structure	Routine tasks and technology; formal rules and policies	Complex tasks; Responsibilities based on expertise	Complex tasks; Collaborative work groups	Complex tasks; Collaborative work groups
Base of power	Knowledge of organizational rules & procedures	Competence	Ability to cultivate relationships	Values
Decision making	Top-down pronouncements	Goal-centered, systematic and analytical	Participatory, deliberative	Organic, intuitive
Leadership style	Dominance, conservative, cautious	Rational achiever, goal oriented	Team builder; concerned, supportive	Idealistic, risk oriented, empowering
Compliance	Monitoring and control	Contractual agreement	Commitment to process	Commitment to values
Evaluation of members	Adherence to rules	Level of productivity	Quality of relationships	Intensity of effort
Orientation to change	Resistant (orientated to maintaining the status quo)	Open to goal driven change	Open to change	Change is embraced as part of growth

organizational culture embedded in the CMM paradigm of organizational change.

We are aware that any type of framework of organizational culture types can be seen as an objectification that can limit understanding of a dynamic and emergent organizational practice. It can also provide a set of *concepts* to help us understand how certain organizational cultures enable or constrain organizational changes. Our interest is in the latter; we are interested in interrogating SPI in general and CMM in particular from an organizational culture perspective with a view to understanding the strengths and limitations of SPI theory and practice.

Quinn and McGrath [38] described and characterized four organizational culture forms: Hierarchical, Rational, Consensual, and Developmental (see Table I). These four forms are rooted in four schools of organization theory: Internal Process, Rational Goal, Human Relations, and Open Systems. For a detailed historical analysis of these four schools of organization theory and the basis for these competing values see [30]. The four organizational culture forms that derive from these schools can be characterized by their core beliefs, routines, and symbolic representations of key aspects of organizational life. The prototypical hierarchical culture is the military, but it can be found in many other organizations. The prototypical rational culture is the production plant oriented to economic measures, productivity, and efficiency. The consensual cultures are oriented toward cohesion, group maintenance and morale. Authority rests with the group, decision-making is participative, and power derives from ability to cultivate and maintain relationships. Consensus cultures are open to change, but require agreement of the group members. The leadership style focuses on team building and a high degree of commitment to group process is expected of members. The development culture is oriented to growth and its organizational purpose is human development. The leadership style is open and empowering, decision-making is people-oriented, and power is based on deeply held values. Change is embraced as the natural evolution of things. As stated earlier, these are ideal types. Organizations may exhibit these cultures to varying degrees or be a mix of cultures [47].

B. Method of Analysis

Our methodology for uncovering the assumptions embedded in the empirical materials was based on content analysis [20], [25]. For this analysis, we selected the authoritative documents of the CMM, SW-CMM 1.1, and P-CMM 2.0, the latest versions published by the Software Engineering Institute (SEI), Carnegie-Mellon University, Pittsburgh, PA. The background for the CMM can be found in the late 1980s U.S. Department of Defense project for evaluating which software developing companies could be expected to deliver high-quality software on time and within budget. Humphrey has explained the philosophical foundations of CMM in [16]. The first version of CMM was published and used by SEI in 1990. From SEI's interaction with software developing companies and from the documentation readily available on their website, the CMM became the most influential framework for improving software processes. A revised version 1.1 was authored by Paulk and others and published in 1993 in two significant technical reports: [33], which provides the rationale and the overview and [35], which gives detailed descriptions of the maturity levels and the key process areas. These two reports were later compiled into a book [34]. Curtis and others published the People CMM (P-CMM) addressing similar improvement of human resources in 1995 in

its first version [3], [4]. A version 2.0 of SW-CMM has been underway since 1995, but its release has been cancelled and the scope of CMM has later broadened considerably. However, the core documents of SW-CMM are still [33], [35] and Version 2.0 of P-CMM [5].

Our analysis of these documents followed a three-stage process:

1) identifying themes around the terms suggested in the competing values framework;
2) exhaustive searching for empirical observations on the themes;
3) analysis and interpretation of the findings.

We started this research with significant knowledge of the CMM gained from participation in a longitudinal (1997–1999) action research project on the implementation of CMM-based or CMM-inspired SPI in four software companies. In this regard, we were able to enter at Stage 2. Using keywords defined from Table I, we conducted an exhaustive iterative search of the documents. The procedure for Stage 2 is as follows.

0) The three documents are available in PDF format. We stripped them of all meta-data by sending them to pdf2txt@adobe.com. The resulting ASCII texts were loaded into ATLAS/TI without modification.
1) We searched the text matching key words derived from the four culture types. For example, from Table I in the category of Organizational Structure, we derive the key words: a) Responsibilities; b) Group; c) Role; d) Complex Task; e) Competence. We used the key words to search the documents and when matches were found we coded the term for all occurrences of it in the text. E.g., all occurrences of "goal" and "goals" where coded with "goal." We found 638 instances of the term *Responsibility*, 1980 instances of the term *Group*, 254 instances of the term *Role*, 5 instances of the term *Complex*, and 1736 instances of the term *Competence*. Table V illustrates part of the results from the search process.
2) The coded sentences were then read. By reading the sentences we were able to find new search terms and repeated Step 1.
3) One of the authors read all the overview chapters of the three documents to find additional search terms. These were used and Steps 1 and 2 were repeated.
4) For each of the three documents, we ran a count of frequencies of word occurrences. Frequently occurring words were selected based on whether they were judged to be significant either in the one of the four culture types or in one of the CMMs. Steps 1 and 2 were repeated for significant terms.
5) We then retrieved segments of text from the documents based on logical relationships between codes, e.g., the sentences coded with "effective" and "goal."

Through this iterative search, reading, and rereading the text excerpts in context, we were able to uncover empirical evidence of assumptions of organizational culture relating to the four types in the competing values framework outlined in Table I. The main themes of our findings are:

1) both CMMs, SW-CMM in particular, are predominantly rational and lead to organizational cultures of the rational form;
2) SW-CMM suffers from some internal contradictions as it shares many aspects with the hierarchical culture type; SW-CMM turns gradually hierarchical at higher levels;
3) P-CMM suffers from an additional, significant, main contradiction between the rational and the consensual culture forms; SW-CMM and P-CMM are mutually inconsistent;
4) both CMMs, P-CMM in particular, express allegiance with the developmental culture form as an end but not as a means.

In the next section, we discuss each of the findings and outline the present significant contradictions for organizational change within the CMM paradigm.

III. Research Findings

According to Paulk *et al.*, SW-CMM is a framework that describes the key elements of an effective software process [33, p. O-7]. In the CMM, "capability" refers to "the range of expected results that can be achieved" [35, p. O-10]. As part of CMM software development, organizations are characterized as *immature* and *mature* (see Table II). According to SW-CMM, software development in immature organizations is accomplished by improvization as opposed to adherence to rules. Software functionality and quality are often compromised to meet deadlines. Schedules and budgets are not based on realistic estimates and are routinely exceeded. However, in mature organizations, software development is carried out according to planned and well-defined processes in which roles and responsibilities are clear. Schedules and budgets are realistic and based on historical data; cost expectations are met and product quality is achieved. In the SW-CMM the *immature* organization is the suboptimal organization that does not follow the rules of "good" software development practice and, thus, falls into difficulty. As an antidote to the "problems of immaturity," "the CMM describes an evolutionary improvement path from an *ad hoc*, immature process to a mature, disciplined process" [35, p. O-7]. The evolutionary improvement path is a five-level model: Level 1 is the immature stage and Levels 2–5 describe what the maturing software development organization needs to focus on in order to achieve that level of maturity.

1) *Initial*. The software process is characterized as *ad hoc* and occasionally even chaotic. Few processes are defined and success depends on individual effort and heroics.
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*. Management and engineering activities are documented, standardized, and integrated into a family of standard software processes for the organization. Projects use a tailored version of the organization's standard software processes for developing and maintaining software.
4) *Managed*. Detailed measures of the software process and product quality are collected. Software processes and products are quantitatively understood and controlled.

TABLE II
CHARACTERIZATION OF IMMATURE AND MATURE ORGANIZATIONS [33]

The Ad Hoc (Immature) Organization	The Mature Organization
• Software processes are generally improvised during the course of the project.	• Software processes are defined and communicated to both existing and new employees.
• Even if software processes have been specified they are not rigorously followed.	• Work activities are carried out according to the planned process.
• The organization is reactionary and the managers are fire fighters.	• Roles and responsibilities are clear throughout the project and across the organization.
• Schedules and budgets are not based on realistic estimates and are routinely exceeded.	• Schedules and budgets are based on historical performance and are realistic.
• Product functionality and quality are often compromised to meet hard deadlines.	• Expected results for cost, schedule, functionality and quality of the product are usually achieved.

5) *Optimizing*. Continuous process improvement is facilitated by quantitative feedback from the process and from piloting innovative ideas and technologies.

The fundamental conjecture of the CMM is that organizational change, along the trajectory of maturity levels, means improving the efficiency and effectiveness of software production, with Level 5 being the pinnacle of software development capability—the mature organization.

A. CMM: The Rational Ideal

Our analysis of the CMM documents reveals that while the proponents espouse the idea that CMM would lead to a dynamic, flexible learning organization, the core assumptions of the CMM paradigm are based on rational rule-governed organization structures that are oriented toward stability, control, and productivity. SW-CMM defines an organization that develops software as a set of processes (i.e., software processes) that can be monitored and controlled to achieve optimal output. This process view of organizations is the fundamental premise of the rational bureaucratic organization (Table I, [38], [39]).

The underlying assumptions about organizational culture in SW-CMM are very much of the rational culture type. The *organizational orientation* in SW-CMM is that of increasing software developers' productivity as well as the organized efficiency and produced quality.

> "At Level 5, new and improved ways of building the software are continually tried, in a controlled manner, to improve productivity and quality. Disciplined change is a way of life as inefficient or defect-prone activities are identified and replaced or revised" [33, p. 38].

> "[A]s a software organization matures, costs decrease, development time becomes shorter, and productivity and quality increase"[33, p. 41].

The task of developing reliable software is a complex task which is reflected in the organizational structure. The primary organizational unit is the project and all effort directed at getting a software organization to Level 2 deals with managing the project.

> "An objective in achieving Level 2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices developed on earlier projects" [33, p. 27].

The *organizational structure* is also created around different roles and groups with separate responsibility and specialized expertise. Special groups are established for software quality assurance, software configuration management, software engineering process, and quantitative process management, among others.

> "There is a group that is responsible for the organization's software process activities, e.g., a software engineering process group, or SEPG" [35, p. O-15].

> "Roles and responsibilities within the defined process are clear throughout the project and across the organization"[33, p. 19].

The software engineering group, for example, has in SW-CMM definite and explicit responsibilities at Level 2, e.g.,

> "The software engineering group reviews the project's proposed commitments" [35, p. O-33].

The scope and responsibilities of the groups vary considerably depending on expertise:

> "Some groups, such as the software quality assurance group, are focused on project activities and others, such as the software engineering process group, are focused on organization-wide activities" [35, p. L3-4].

The SW-CMM is goal seeking in several ways. In the documentation, the term "goal" plays a significant role. The two main documents [33], [35] add up to more than 500 pages. The term "goal" occurs 344 times making it one of the 20 most used significant words. First and foremost, the whole idea of the CMM is a goal-seeking one where the ultimate goal is the Level 5 organization, but where subgoals are formulated in terms of Levels 2–4. The goals for the improvement are set by the CMM.

> "To achieve lasting results from process improvement efforts, it is necessary to design an evolutionary path that increases an organization's software process maturity in

stages. The software process maturity framework…orders these stages so that improvements at each stage provide the foundation on which to build improvements undertaken at the next stage" [33, p. 22].

It is also goal seeking within each of the processes. For each key process area there is a set of goals. A goal is defined as:

"The goals summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries and intent of each key process area" [35, p. O-11].

These goals are set by the SW-CMM, e.g., for the key process area Intergroup Coordination at Level 3 the goals are:

"Goal 1: The customer's requirements are agreed to by all affected groups.

Goal 2: The commitments between the engineering groups are agreed to by the affected groups.

Goal 3: The engineering groups identify, track and resolve intergroup issues"[35, p. L3-85].

Decision making in CMM is also based on the rational culture form. Decisions are driven by goals and in decision-making goals are pursued in a systematic and analytical way.

"In a mature organization, managers monitor the quality of the software products and customer satisfaction. There is an objective, quantitative basis for judging product quality and analyzing problems with the product and process" [33, p. 19].

Further, each key process is evaluated based on a set of predefined measurements and a set of verification activities. For example, the intergroup coordination process at Level 3 is evaluated by the following measurements and verification activity.

"Measurement 1: Measurements are made and used to determine the status of the inter-group coordination activities" [35, p. L3-93].

"Verification 3: The software quality assurance group reviews and/or audits the activities and work products for inter-group coordination and reports the results" [35, p. L3-94].

Compliance is, to a large extent, based on meeting commitments. Commitments are common in the formulation of key process areas.

"Goal 3: Affected groups and individuals agree to their commitments related to the software project" [35, p. L2-12].

Humphrey, in providing the philosophical underpinning of the CMM, explains commitment and commitment discipline in the following way:

"The foundation for software project management is the commitment discipline. … Commitments are not met by reviews, procedures, or tools, however; they are met by committed people" [16, p. 69].

"In simplest terms a commitment is an agreement by one person to do something for another" [16, p. 70].

"Commitment is a way of life. Committed organizations meet their large and their small commitments" [16, p. 71].

This attitude coincides with the idea of compliance as contractual agreement as defined in the rational culture type. In the SW-CMM, many commitments are made—they form a hierarchy, they are sometimes made informally (but not lightly), and they are often made formally and documented, hence, making it easier to observe compliance and noncompliance.

Based on a similar analysis of P-CMM, we found that it too leads to a dominant rational culture type.

B. SW-CMM: The Hierarchical Ideal

In SW-CMM, a software process is set of activities, methods, practices, and transformations that developers use to develop and maintain software and the associated products, e.g., project plans, design documents, code, test cases, and user manuals [35]. Further, SW-CMM defines "software process maturity [as] the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective" [33, p. 4]. Some of the key features of the CMM process view that clearly reflect the hierarchical culture are: 1) an orientation to stability and control; 2) precise job definitions; 3) clear lines of authority; and 4) strict policies and management controls. Table IV shows the organizational structures required by SW-CMM. As Paulk *et al.* explain, the SW-CMM "provides software organizations with guidance on how to gain control over their processes for developing and maintaining software and how to evolve to a culture of software engineering and excellence" [33, p. 5]. From this perspective, the SW-CMM espouses an organizational culture form in which people and processes are treated mechanistically like a machine, for which the operation and performance can be quantified, measured, and controlled. These assumptions are evident in the SW-CMM's overview of itself:

"The key process areas are categorized … into three broad categories: Management, Organizational and Engineering processes. The Management process category contains the project management activities as they evolve from planning and tracking at Level 2, to managing according to a defined software process at Level 3, to quantitative management at Level 4, to innovative management in a constantly changing environment at Level 5. The Organizational process category contains the cross-project responsibilities as the organization matures, beginning with a focus on process issues at Level 3, continuing to a quantitative understanding of the process at Level 4 and culminating with the management of change in an environment of continuous process improvement at Level 5. The Engineering process category contains the technical activities, such as requirements analysis, design, code and test, which are performed at all levels, but that evolve toward an engineering discipline at Level 3, statistical process control at Level 4 and continuous measured improvement at Level 5" [35, p. O-26].

In more detail, there are several elements of SW-CMM that share its assumptions about organizational culture with the hierarchical culture type. The orientation is toward stability and control.

"The first responsibility and the focus of Level 4, is process control. The software process is managed so that

it operates stably within a zone of quality control" [33, p. 17].

"At Level 2, the customer requirements and work products are controlled and basic project management practices have been established. These management controls allow visibility into the project on defined occasions" [33, p. 21].

The SW-CMM requires considerable knowledge of organizational rules and procedures in terms of standard processes, though these are more general than what is often seen in hierarchical culture types. Decision-making, thus, also has elements of top-management pronouncement in the way the standard software process is common to all projects. The project decides its own defined software process, but within the limits of the standard software process.

"Projects tailor the organization's standard software process to develop their own defined software process, which accounts for the unique characteristics of the project. This tailored process is referred to in the CMM as the project's defined software process. … Because the software process is well defined, management has good insight into technical progress on all projects" [33, p. 12].

There are also elements of measuring compliance by monitoring and control. All key process areas stipulate measurement and verification activities. For example, the measurement and verification activity for the key process area "Software Configuration Management" at Level 2 is:

"Measurement 1: Measurements are made and used to determine the status of the SCM activities" [35, p. L2-87].

"Verification 1: The SCM activities are reviewed with senior management on a periodic basis" [35, p. L3-87].

Another aspect of CMM that reflects the hierarchical organizational form is its fascination with rules. In the more than 500 pages of SW-CMM documentation and 735 pages of P-CMM documentation, what stands out is pervasiveness of rules. There are detailed descriptions of key process areas with: goals, activities, measurement, and verification. These detailed description are formal rules, procedures, and policies to be followed, they also stipulate a large number of organizational rules and policies that establish and enforce the organizational structures.

The elements of the hierarchical culture type are present but not dominant at the lower levels of SW-CMM. The higher the level in SW-CMM the more the hierarchical culture type is imposed on the software processes. At Levels 2 and 3, the rational culture type is imposed as part of the evolutionary, staged change activities though there are definitely elements of the hierarchical culture type. At Levels 4 and 5, there is a drift away from the rational culture type toward more and more of the hierarchical culture type.

C. P-CMM: The Consensual Ideal

The P-CMM is a framework for guiding organizations in attracting, developing, motivating, organizing, and retaining the talented people needed to continually improve their software development [5]. However, the organizational culture that P-CMM prescribes is based on the human relations approach to management that is contradictory to the SW-CMM rational machine view of the organization. The management and leadership styles of the P-CMM and the SW-CMM models are diametrically opposed. The management and leadership styles embedded in the SW-CMM descriptions reveal an orientation to monitoring, control, and rule compliance. The style suggested in P-CMM involves mentoring, coaching, and team building. Further, decision-making processes in the SW-CMM are closed and roles and responsibilities explicitly defined, while the processes suggested by P-CMM are participatory and open.

According to Curtis *et al.* [5], the strategic objectives of P-CMM are:

1) improve the capability of the software organization by increasing the capability of its workforce;
2) ensure that competences for developing software are organizational rather than individual;
3) align the motivations of individuals with those of the organization;
4) retain the human assets of the organization.

P-CMM outlines a similar five-level evolutionary model for assessing the organization and implementing improvements. According to P-CMM, the worst-case scenario of Level 1 is that managers do not accept responsibility for developing their employees. They put little effort into evaluating job candidates and the performance of employees; consequently, employees are disgruntled and the capability of the organization is undermined. The P-CMM remedy for this problem is the five-level model.

1) *Initial*. The software organization's capability is unknown since there is no effort to measure it. Individuals are motivated to pursue their own agendas since there are few incentives to pursue the organization's objectives.
2) *Repeatable*. Instill basic discipline into workforce activities. Eliminate problems that keep people from being able to perform their responsibilities effectively. Establish a foundation of workforce practices that can continually improve the workforce.
3) *Defined*. Identify primary competences and align with workforce activities. Adapt the workforce practices to develop specific skills and competences that the organization needs. Identify best practices and tailor them to the organization.
4) *Managed*. Quantitatively manage organizational growth in workforce capabilities and establish competence-based teams. Collect and analyze performance data to evaluate competence.
5) *Optimizing*. Continuously improve methods for developing personal and organizational competence.

In line with the structure of SW-CMM, P-CMM defines a set of key processes that must be implemented for each level of maturity before moving on to the next. The processes are categorized into four areas: 1) developing capabilities; 2) building teams and culture; 3) motivating and managing performance; and (4) shaping the workforce. P-CMM suggests that the path to developing capabilities starts with delivering training in oral and written communication, followed by systematic assessments of competence requirements and implementing organizational structures for competence development. The

TABLE III
KEY PROCESSES FOR DEVELOPING ORGANIZATIONAL CAPABILITIES [5]

Maturity levels	Developing individual capabilities	Building workgroups & culture	Motivating & managing performance	Shaping the workforce
5 Optimizing	Continuous Capability Improvement		Organizational Performance Alignment	Continuous Workforce Innovation
4 Managed	Competency Based Assets Mentoring	Competency Integration Empowering Workgroups	Quantitative Performance Management	Organizational Capability Management
3 Defined	Competence Development Competency Analysis	Workgroup Development Participatory Culture	Competency Based Practices Career development	Workforce Planning
2 Repeatable	Training and Development	Communication & Coordination	Compensation Performance management Work Environment	Staffing
1 Initial				

model also suggests mentoring and coaching as important activities for developing the capabilities of the employees. Prescriptions are given for developing a participatory culture and team building. Table III summarizes the processes that the organization must implement to achieve each level of maturity in the P-CMM scheme.

In many respects, P-CMM exhibits fundamental assumptions of the consensual organizational culture type (see Table I). The following aspects of the consensual culture type are all part of the core idea of P-CMM: group maintenance as an organizational objective, the organization structured around collaborative workgroups, decision making to be participatory, and the leadership style based on team-building. At Level 2, we find the key process area Communication and Coordination where the purpose is:

> "To establish timely communication across the organization and to ensure that the workforce has the skills to share information and coordinate their activities efficiently" [5, p. 141].

At Level 3, there is the key process area Participatory Culture. The purpose of Participatory Culture is to create the ability to participate in decision-making.

> "The open communication established with Communication and Coordination practices at the Managed Level creates a foundation for developing a participatory culture. A participatory culture provides an environment in which competent professionals are fully able to exercise their capabilities" [5, p. 379].

At Level 4, there is the key process area Empowering Workgroups with the purpose:

> "To invest workgroups with the responsibility and authority for determining how to conduct their business activities most effectively" [5, p. 141].

The development and management of individual and organizational competences also play an equally important role in P-CMM. The primary vehicle for developing individual and organizational competence is the workgroup.

> "The purpose of Workgroup Development is to organize work around competency-based process abilities" [5, p. 347].

However, is this really a departure from the rational model of the work group espoused in SW-CMM? Close observation reveals that, in principle, the same rational culture assumptions about organizational structure are operating here. Some might argue that the team organization and goal-oriented structure of the SW-CMM software organization suggest a consensual or developmental organizational form. However, the underlying context of P-CMM tells a different story. At first sight the concepts "team work" and "participatory decision making" belong to the consensual culture type where the idea of process is also strong. However, this is not the same in SW-CMM. The key process areas are first and foremost defined by their goals. Goal-driven behavior—in particular adherence to predefined goals—belongs to the rational culture type and can never be a core idea of the consensual culture type. Also, in the consensual culture type compliance is measured by commitment to process and that seems to be a core idea in SW-CMM as well. However, we have already seen that compliance in SW-CMM is measured by contractual agreement, i.e., that commitments are met. The consensual culture's idea of a process is that the process itself ensures the right outcome. This is similar to democracy, where there is a particular focus on the process through which agreement and consensus is reached and without an ideology about which is the right outcome. Such participatory ideas are essential in consensual cultures but are foreign to SW-CMM. Such ideas about consensual processes are not common in P-CMM. The workgroup structure defined in P-CMM is created based on explicit rules.

It makes sense that SW-CMM and P-CMM should be considered as one "voice." They were both developed at the SEI, both are official publications of the SEI, and one person was a coauthor of both SW-CMM and P-CMM. Also in the P-CMM, we find the following:

> "The People CMM employs the process maturity framework of the highly successful Capability Maturity Model for Software … as a foundation for a model of best practices for managing and developing an organization's workforce" [5, p. vi].

P-CMM contains a major inner contradiction in adhering both to the consensual culture type and to the rational culture type together with SW-CMM. Thus, SW-CMM and P-CMM are also mutually inconsistent.

D. Tensions of Developmental and Hierarchical Cultures

Although the process view is dominant in CMM, it also espouses a developmental culture for the software organization. Paulk states that the CMM-based SPI approach "should build an organization that can dynamically adapt to a rapidly changing, even chaotic, environment; an organization that knows what business it is in and pursues software projects aligned with its strategic business objectives; a learning organization that explicitly, rather than implicitly, captures knowledge; an organization

managed by facts rather than intuition, while still valuing creativity; an organization that empowers its most crucial asset: its people" [32]. Although the P-CMM prescribes team building and a participatory culture, the hierarchical structures of CMM work processes with their explicitly defined role responsibilities and strict management control are contradictory to building trust upon which a developmental culture thrives.

There are also a few elements of P-CMM that suggests that it contains elements of a developmental culture. These have to do with empowerment of individuals. However, when viewed in the context of the super structure of the CMM paradigm, they seem contradictory. For example, the SW-CMM documents suggest that:

> "A disciplined process, then, empowers the intellect, while regimentation supplants it" [16, p. 13].

However, in SW-CMM the term "empower" is mentioned twice. First, in an activity in the key process area Change Process Management at Level 5:

> "Activity 1: A software process improvement program is established which empowers the members of the organization to improve the processes of the organization" [35, p. L5-37].

Paulk *et al.* mention this again in their Appendix where some of the activities of Change Process Management are repeated [35]. Empowerment and the orientation toward flexibility, adaptability, and readiness could be inherent in SW-CMM without the term "empower" being used, but this is not the case. Humphrey, for one, is aware of the fine balance between discipline and regimentation, but even so, neither he nor SW-CMM advocate any processes through which the software developers and their projects are given power, resources, means, and responsibilities for making their own decisions about their own activities. On the contrary, at Level 3 and above, the standard software process is organization-wide. There is no indication that any key process area or any aspect of SW-CMM, even in a generic way, resembles the developmental culture type. Decision-making is not organic and intuitive; it is goal-centered, systematic, and analytical. Compliance is not measured by commitment to value. Change is not part of growth; it is driven by a desire to increase flexibility, adaptability, and readiness without jeopardizing productivity.

> "At Level 5, new and improved ways of building the software are continually tried, in a controlled manner, to improve productivity and quality. Disciplined change is a way of life as inefficient or defect-prone activities are identified and replaced or revised. Insight extends beyond existing processes" [33, p. 22].

While there is the claim that SW-CMM leads to more creativity and empowerment, the assumption shows a different picture.

The P-CMM, on the other hand, presents an elaborate vision of the empowered group culture, but in a way that enforces its adherence to the consensual culture type. At Level 3, the P-CMM has a key process area called Empowering Workgroups.

> "Practice 4: Empowered workgroups are delegated the responsibility and authority to determine the methods by which they will accomplish their committed work" [5, p. 451].

That means that within the limits of the standard process in general, and specifically the commitments made, a workgroup functions with some autonomy. In that sense, it belongs to the developmental culture type. No other aspect of the P-CMM belongs to the developmental culture type. The key process area Empowering Workgroups, thus, puts emphasis on the group and the group's processes in a way that strengthens its consensual culture. The workgroup become a dominant organizational structure, but decision-making while participatory is based on systematic and deliberative application of rules.

IV. Conclusion

The design ideal of CMM is the rational bureaucratic learning organization that is flexible. It is not surprising to find that the CMM models (SW-CMM and P-CMM) contain several major organizational cultural contradictions in their core assumptions. Both models, and SW-CMM in particular, are designed on the basis of rational ideal and lead to organizational cultures of the rational form. SW-CMM suffers from some internal contradictions as it shares many aspects with the hierarchical culture type; SW-CMM turns gradually more hierarchical at higher levels. P-CMM suffers from an additional contradiction as its core elements are designed with major elements of both the rational and the consensual culture forms. In their core assumptions, the SW-CMM and P-CMM are contradictory and antagonistic. Both express allegiance with the developmental culture form as an end but not as a means, as they adopt a rational-hierarchical process view. While we agree that "value conflicts" are inherent in organizational change initiatives [19], [21], we believe that these inconsistencies and contradictions are not simple and may not be easily overcome. Further, managers of software organizations seeking guidance in the CMM not only run the risk of being confused at a theoretical level but also have to face the inconsistencies and contradictions in practice.

If we could come to terms with the inconsistencies and contradictions per se, the pervasive, massive task of SPI practice adds profoundly to the size of the problems to face. The scale and complexity of organizational change that an immature organization must implement to become a mature organization is simply breathtaking [48], [51]. According to CMM, becoming a mature/optimal software development organization requires fundamental change across several dimensions of the organization:

1) core processes;
2) software development technologies;
3) management and control procedures;
4) planning;
5) work group organization;
6) roles and responsibilities;
7) power and authority structures;
8) skills and knowledge.

The pervasive change in terms of scale, complexity, and depth of the organizational change required by SW-CMM is too much to outline in this paper. Therefore, we have chosen to summarize its basic elements in Table IV and outline some of it in very

TABLE IV
DIMENSIONS OF ORGANIZATIONAL CHANGE REQUIRED BY CMM [51]

Change Aspects	Change Objective CMM Level 2	Change Objective CMM Level 3	Change Objective CMM Level 4	Change Objective CMM Level 5
Core Business Processes	Requirements Management (RM), Project Planning (PP), Project Tracking & Oversight (PTO), Subcontract management (SM), Quality Assurance (SQA), Configuration Management (SCM)	Organization Process Focus (OPF), Organization Process Definition (OPD), Training Program (TP), Integrated Software Management (ISM), Software Product Engineering (SPE), Intergroup Coordination (IC), Peer Reviews (PR)	Software Quality Management (SQM), Quantitative Process Management (QPM)	Process Change Management (PCM), Technology Change Management (TCM), Defect Prevention (DP)
Product/Process Technology	Tools and techniques to support the work in RM, SPP, SPTO, SSM, SQA, and SCM; A configuration management library.	Tools and methods defined and integrated to support work in OPF, OPD, TP, ISM, SPE, IC, PR, Software Process Database; Library of process-related documentation.	Tools to support quantitative process management; Support for collecting, recording, and analyzing data; Tools to support predicting, measuring, tracking, and analyzing the quality of products; Measurement program in place.	Tools to support defect prevention activities, technology change management and SPI; Support for collecting and analyzing data needed to evaluate technology changes; Appropriate data on the software processes and products are available; Records of SPI activities;
Management & Control procedures	Measurements performed and used in core processes; Activities and work products are reviewed – by senior management – by project manager – by SQA group	Manage project in accordance to defined process; Measurements performed and used in core processes; Activities and work products are reviewed – by senior management – by project manager – by SQA group	Define, monitor, and revise project's quantitative quality goals; Measure, analyze, and compare the quality of the project's software products to the products quantitativve quality goals; Measurements performed and used; Activities and work products are reviewed - by senior management - by project manager - by SQA group	Measurements are made and used; Activities and work products are reviewed by senior management project manager SQA group
Planning	Software project planning, Quality planning, Configuration management planning, Resource planning	SPI planning; Training planning; Peer reviews planning;	Plan for quantitative process management; Project's software quality plan	Plan for defect prevention activities; Document and track defect prevention data; Plan for technology change management;
Work Group Organization	Software engineering groups in place	Staff groups, Software engineering groups	Quantitative process management group	Members of the organization participate in SPI teams
Tasks	Tasks defined and linked into documented procedures for project management, e.g. – develop project plan – estimate the size, effort, cost – perform formal reviews – define and plan work to be sub-contracted – prepare SQA plan – prepare SCM plan – control changes to baselines	Tasks defined, linked, and integrated into a standard process, Tasks are performed in accordance to a defined process, e.g. – develop and maintain organization's standard software process – develop and maintain organizations training plan – a waiver procedure for required training – tailor standard software process to project's defined software process – develop and revise project's software plan – identify, negotiate, and track critical dependencies between engineering groups – perform peer reviews	Perform tasks in accordance to documented procedures, e.g. - develop software project's plan for quantitative process management - collect measurement data - analyze project's defined software process and bring it under quantitative control - establish and maintain the process capability baseline - develop and maintain project's software quality plan	Perform tasks in accordance to documented procedures, e.g. - conduct causal analysis meetings - incorporate revisions resulting from defect prevention activities to the organization's standard software process and the project's defined software process - select and acquire technologies for the organization and software projects - incorporate appropriate new technologies into organization's standard software process and the projects' defined processes - develop and maintain a plan for SPI - handle SPI proposals - implement the improvement

Change Aspects	Change Objective CMM Level 2	Change Objective CMM Level 3	Change Objective CMM Level 4	Change Objective CMM Level 5
Roles	Particular project manager roles defined, Special groups or functions established for SQA and SCM	Special staff groups established with organization-wide responsibilities e.g. a SEPG to define and improve organization process and a training group to educate and train organization members.	Quantitative process management group; Individuals implementing and supporting software quality management	An organization-level team to coordinate defect prevention activities; A project-level group to coordinate defect prevention activities; A technology change management group
Power & Authority	A senior manager with authority to take oversight actions in SQA; A board with authority for managing the project's software baselines.	Senior management sponsors and oversees SPI;		Management participation in defect prevention activities; Senior management sponsors and oversees the organization's activities for technology change management and SPI; Establish a SPI program which empowers the members of the organization to improve
Co-ordinating Mechanisms	Written organizational policies for implementing RM, SPP, SPTO, SSM, SQA, SCM; Documented procedures for project management, Standard reports are made available; Project plans.	Written organizational policies in place for 1) coordinating software process development and improvement activities 2) developing and maintaining a standard software process, 3) meeting its training needs, 4) planning and managing the software project using the organization's standard process, 5) performing the software engineering activities, 6) establishing interdisciplinary engineering teams, 7) performing peer reviews; Standard process defined, tailored and used by projects	Written organizational policies in place for 1) measuring and quantitatively controlling the performance, 2) analyzing the process capability, 3) managing software quality; Project's software quality plan	The organization follows a written policy for 1) defect prevention activities, 2) improving its technology capability, 3) implementing SPI; The project follows a written organizational policy for defect prevention activities; Periodic meetings to review and coordinate implementation of action proposals; TCM group works with software projects; The SPI group; The SPI plan
Skills & Knowledge	Groups and project management positions staffed with competent people, Orientation to members, e.g. - role, responsibilities, authority, and value of the SQA group – technical aspects	Staff groups and projects staffed with competent people, Orientation to members, e.g. - SPI activities - training program – team work	Orientation to member, e.g. goals and values of quantitative process management reports documenting the results of software project's quantitative process management activities	Feedback to members, e.g. status and results of the organization's and project's defect prevention activities new technologies status and results of the SPI activities

brief detail. Level 1 is the starting point of the CMM scale; this is the immature organization. In order to move to Level 2, the Level 1 organization must undertake profound organizational changes in each of the eight dimensions. For example, the organization must implement business processes, formal procedures, and supporting technologies for the six key processes: requirements management, project planning, project tracking and oversight, subcontract management, quality assurance, and configuration management. It must also implement organizational policies and management, controlling, and tracking procedures for each of the key processes. CMM is quite ambitious as a change approach; it attempts to change all aspects of the organization structure and culture (cf. Table IV). On the symbolic level, CMM-based SPI seeks to change the activities and patterned behaviors of developers. On the level of structure, it seeks to change norms, conventions, customs, rules, and procedures for doing and managing software development.

CMM's adherence to the rational culture type makes it less effective as an approach to deal with such scale and complexity of SPI. These limitations of CMM and their consequences for informing SPI practice may be lessened, but probably never alleviated completely. We believe that CMM can benefit from a more thorough understanding of the organization theory (design, culture, and change). One direction for extending the CMM would be to incorporate a framework of organizational change that can guide the change process more effectively. However, even this extension to CMM would still leave the question of reconciling the organizational design and cultural contradictions that are

TABLE V
ILLUSTRATION OF SEARCH TERMS AND SOME CITATIONS IN SUPPORT OF THE RATIONAL CULTURE TYPE

Search Terms and Frequency	Some Excerpts From The Documents
Orientation: Productivity and efficiency Productivity (60), efficiency (94), quality (502)	At Level 5, new and improved ways of building the software are continually tried, in a controlled manner, to improve productivity and quality. Disciplined change is a way of life as inefficient or defect-prone activities are identified and replaced or revised. [33, p. 38]
	as a software organization matures, costs decrease, development time becomes shorter, and productivity and quality increase. [33, p. 41]
Organizational objectives: Pursuit of objectives Objective (786)	An objective in achieving Level 2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices developed on earlier projects [33, p. 27]
Organizational structure: Complex tasks, responsibilities based on expertise Responsibility (638), group (1980), role (254), complex (5), competence (1736)	There is a group that is responsible for the organization's software process activities, e.g., a software engineering process group, or SEPG [35, p. O-15]
	Roles and responsibilities within the defined process are clear throughout the project and across the organization. [33, p. 19]
	The software engineering group reviews the project's proposed commitments. [35, p. O-33]
	Some groups, such as the software quality assurance group, are focused on project activities, and others, such as the software engineering process group, are focused on organization-wide activities. [35, p. L3-4]
	While questions can appropriately be raised about the size and complexity of current systems, these are human creations, and they will, alas, continue to be produced by human beings (with all their failings and creative talents. … [T]he complexity of our systems is increasing, which will make the systems progressively more difficult to test. [16, p. 13]
	The purpose of Workgroup Development is to organize work around competency-based process abilities. (Curtis et al. 2001, p. 347)
Decision-making: Goal-centered, systematic, and analytical Result (561), goal (683), quality (502), monitor (72), measure (563)	To achieve lasting results from process improvement efforts, it is necessary to design an evolutionary path that increases an organization's software process maturity in stages. The software process maturity framework orders these stages so that improvements at each stage provide the foundation on which to build improvements undertaken at the next stage. [33, p. 22]
	The goals summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area. [35, p. O-11]
	Goal 1: The customer's requirements are agreed to by all affected groups. Goal 2: The commitments between the engineering groups are agreed to by the affected groups. Goal 3: The engineering groups identify, track, and resolve intergroup issues. [35, p. L3-85]
	In a mature organization, managers monitor the quality of the software products and customer satisfaction. There is an objective, quantitative basis for judging product quality and analyzing problems with the product and process. [33, p. 19]
	Measurement 1: Measurements are made and used to determine the status of the intergroup coordination activities. [35, p. L3-93]
	Verification 3: The software quality assurance group reviews and/ or audits the activities and work products for intergroup coordination and reports the results. [35, p. L3-94]
Compliance: Contractual arrangement Commitment (384)	Goal 3: Affected groups and individuals agree to their commitments related to the software project. [35, p. L2-12]
	The foundation for software project management is the commitment discipline. … Commitments are not met by reviews, procedures, or tools, however; they are met by committed people. [16, p. 69]
	In simplest terms a commitment is an agreement by one person to do something for another. [16, p. 70]
	Commitment is a way of life. Committed organizations meet their large and their small commitments. [16, p. 71]
Orientation to change: Open to goal-driven change Change (535)	At Level 5, new and improved ways of building the software are continually tried, in a controlled manner, to improve productivity and quality. Disciplined change is a way of life as inefficient or defect-prone activities are identified and replaced or revised. Insight extends beyond existing processes. [33, p. 22]

embedded in CMM. The question of how to reconcile the core assumptions of the rational culture and those of the developmental cannot be resolved by simply improving the organizational change strategy. This would require adopting a different strategy. We outline two such strategies, but whether either of them is effective we shall leave for future research.

In the first strategy, we believe that CMM might benefit from being supplemented with a rational and dynamic approach to handle the main contradictions in SPI practice. The contradictions between distinctive paradigms in information systems development have received much attention, (e.g., [10], [11], [21], [48]). In that perspective, the dilemmas facing CMM are not new and are not reconcilable. Nevertheless, Klein and Hirschheim [21] suggest a rational decision-making approach to choose between competing values and design ideals. Their approach serves the primary purpose of extending the decision-making well beyond the traditional technical issue. If we transfer that idea to CMM and outline it in terms of culture types, we would need to supplement CMM with an approach where we could for a particular software organization choose a specific balance between the organizational culture types. In one organization, we could end up with a main focus on implementing the rational culture type improvement while in another organization we could well end up with a main focus on implementing the consensual and developmental culture types. That would not in itself remove the cultural inconsistencies

and contradictions, but it would create a process through which these could be explicitly addressed and handled.

In the second strategy, we believe that CMM might benefit from a different perspective for the organization and management of software development. It is accepted that software development is a nonroutine complex undertaking requiring high levels of competence and a flexible organizing structure. That is why CMM not only has hierarchical and rational culture elements, but also the consensual elements. The fundamental issue for software organizations is how to achieve a balance between control and goal-orientation on the one hand and change and flexibility on the other hand—between the rational culture and the developmental culture. The current CMM focus on controlling the processes of software development can easily lead to a level of bureaucratization of software organizations that is less flexible than desired. While no one would suggest that software processes should not be defined, there needs to be some flexibility in their implementation and execution. There is another perspective for organizing and managing software development that may offer some possibilities for moving CMM out of its current process model. This organizational form, the professional bureaucracy, has been thoroughly researched and discussed by Mintzberg [29]. It has been found to provide a high level of flexibility coupled with specialization and predictability of outputs. The professional bureaucracy focuses on standardization of skills and indoctrination of the professional. Professional bureaucracies are based on trust and competence and have been highly successful in specialized work such as surgery, engineering, scientific research, and so on. Professionalization of software engineering would offer a tradition, standards and a culture that is well entrenched and recognizable without regard to the particular organizational setting.

REFERENCES

[1] M. Alvesson, "Organizations, culture and ideology," *Int. J. Stud. Manage. Org.*, vol. 17, pp. 4–18, 1987.

[2] P. L. Berger and T. Luckmann, *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*. New York: Anchor, 1966.

[3] B. Curtis, W. E. Hefley, and S. Miller, "Overview of the people capability maturity model," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, PA CMU/SEI-95-MM-01, 1995.

[4] ——, "People capability maturity model," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, CMU/SEI-95-MM-002, 1995.

[5] ——, "People capability maturity model (P-CMM)," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, CMU/SEI-01-MM-001, 2001.

[6] T. E. Deal and A. A. Kennedy, *Corporate Cultures: The Rites and Rituals of Corporate Life*. Reading, MA: Addison-Wesley, 1982.

[7] K. E. Emam, J.-N. Drouin, and W. Melo, *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. Los Alamitos, CA: IEEE Computer Society, 1998.

[8] L. K. Grundy and D. M. Rousseau, "Critical incidents in communicating culture to newcomers: The meaning is in the message," *Hum. Relat.*, vol. 47, pp. 1063–1088, 1994.

[9] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk, "Software quality and the capability maturity model," *Commun. ACM*, vol. 40, pp. 30–40, 1997.

[10] R. Hirschheim, J. Iivari, and H. Klein, "A paradigmatic analysis contrasting information systems development approaches and methodologies," *Inform. Syst. Res.*, vol. 9, pp. 164–193, 1998.

[11] R. Hirschheim, H. Klein, and K. Lyytinen, "Exploring the intellectual structures of systems development: A social action theoretic analysis," *Inform. Org.*, vol. 6, pp. 1–64, 1996.

[12] G. Hofstede, "Culture and organizations," *Int. Stud. Manage. Org.*, vol. 10, pp. 15–41, 1981.

[13] ——, *Culture and Organizations: Software of the Mind*. New York: McGraw-Hill, 1991.

[14] G. Hofstede, B. Neuijen, D. D. Ohayv, and G. Sanders, "Measuring organizational cultures: A qualitative and quantitative study across twenty cases," *Administ. Sci. Quart.*, vol. 35, pp. 286–316, 1990.

[15] W. S. Humphrey, "Characterizing the software process," *IEEE Software*, vol. 5, pp. 73–79, 1988.

[16] ——, *Managing the Software Process*. Reading, MA: Addison-Wesley, 1989.

[17] ——, "Introduction to software process improvement," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, CMU/SEI-92-TR-007, 1992.

[18] J. Johansen and L. Mathiassen, "Lessons learned in a national SPI effort," in *Proc. EuroSPI'98*, Gothenburg, Sweden, 1998.

[19] H. Klein, "Design ideals and their critical reconstruction," in *Proc. TIMS*, Toronto, ON, Canada, 1981, pp. 12–26.

[20] H. Klein and D. Truex, "Discourse analysis an approach to investigating organizational emergence," in *Signs of Work*, B. Holmqvist, P. B. Andersen, H. Klein, and R. Posner, Eds. New York: Walter de Gruyter, 1996.

[21] H. K. Klein and R. Hirschheim, "Choosing between competing design ideals in information systems development," *Inform. Syst. Front.*, vol. 3, pp. 75–90, 2001.

[22] P. Kuvaja, J. Similä, L. Krzanik, A. Bicego, S. Saukkonen, and G. Koch, *Software Process Assessment & Improvement—The Bootstrap Approach*. Oxford, U.K.: Blackwell, 1994.

[23] M. R. Louis, "A cultural perspective on organizations: The need for and consequences of viewing organizations as culture-bearing milieux," *Hum. Syst. Manage.*, vol. 2, pp. 245–258, 1981.

[24] C. C. Lundberg, "Working with culture," *J. Org. Change Manage.*, vol. 1, pp. 38–47, 1989.

[25] P. Manning and B. Cullum-Swan, "Narrative, content and semiotic analysis," in *Handbook of Qualitative Research*, N. Denzin and Y. Lincoln, Eds. Newbury Park, CA: Sage, 1994.

[26] G. A. Marcoulides and R. H. Heck, "Organizational culture and performance: Proposing and testing a model," *Org. Sci.*, vol. 4, pp. 209–225, 1993.

[27] F. McGarry, R. Pajerski, G. Page, S. Waligora, V. Basili, and M. Zelkowitz, "Software process improvement in the NASA software engineering laboratory," Software Eng. Inst,, Carnegie-Mellon Univ,, Pittsburgh, PA, Tech. Rep.CMU/SEI-94-TR-22, 1994.

[28] J. C. H. Mills and A. J. Mills, "Rules, sensemaking, formative contexts and discourse in the gendering of organization culture," in *Handbook of Organizational Culture & Climate*, N. M. Ashkanasy, C. P. M. Wildrom, and M. F. Peterson, Eds. Newbury Park, CA: Sage, 2000, pp. 55–70.

[29] H. Mintzberg, *Structure in Fives: Designing Effective Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[30] G. Morgan, *Images of Organization*, 2 ed. Newbury Park, CA: Sage, 1997.

[31] P. A. Nielsen and J. Nørbjerg, "Software process maturity and organizational politics," in *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective, Proc. IFIP WG 8.2 Conf.*, B. Fitzgerald and N. Russo, Eds., Boise, ID, 2001.

[32] M. C. Paulk, "Effective CMM-based process improvement," in *Proc. 6th Int. Conf. Software Quality*, Ottawa, ON, Canada, 1996.

[33] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "Capability maturity model for software, Version 1.1," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU/SEI-93-TR-024, 1993.

[34] M. C. Paulk, C. Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley, 1995.

[35] M. C. Paulk, C. V. Weber, S. M. Garcia, M. B. Chrissis, and M. Bush, "Key practices of the capability maturity model, Version 1.1," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, CMU/SEI-93-TR-025, 1993.

[36] T. J. Peters and R. H. Waterman Jr, *In Search of Excellence: Lessons From America's Best-Run Companies*. New York: Harper & Row, 1982.

[37] A. M. Pettigrew, "On studying organizational cultures," *Administ. Sci. Quart.*, vol. 24, pp. 570–581, 1979.

[38] R. E. Quinn and M. R. McGrath, "The transformation of organizational cultures: A competing values perspective," in *Organizational Culture*, P. J. Frost, L. F. Moorre, M. R. Louise, C. C. Lundberg, and J. Martin, Eds. Newbury Park, CA: Sage, 1985, pp. 315–334.

[39] R. E. Quinn and J. Rohrbaugh, "A spatial model of effectiveness criteria: Toward a competing values approach to organizational analysis," *Manage. Sci.*, vol. 29, pp. 363–377, 1983.

[40] A. Rafaeli and M. Worline, "Symbols in organizational culture," in *Handbook of Organizational Culture & Climate*, N. M. Ashkanasy, C. P. M. Wilderom, and M. F. Peterson, Eds. Newbury Park, CA: Sage, 2000, pp. 71–84.
[41] V. Sathe, *Culture and Related Corporate Realities*. Homewood, IL: Irwin, 1985.
[42] E. H. Schein, *Organizational Culture and Leadership: A Dynamic View*. San Francisco, CA: Jossey-Bass, 1985.
[43] ——, "What is culture?," in *Reframing Organizational Culture*, P. J. Frost, L. F. Moore, M. R. Louise, C. C. Lundberg, and J. Martin, Eds. Newbury Park, CA: Sage, 1991, pp. 243–253.
[44] "Process maturity profile of the software community," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, 1997.
[45] "Process Maturity Profile of the Software Community," Software Eng. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, 2002.
[46] L. Smircich, "Concepts of culture and organizational analysis," *Administ. Sci. Quart.*, vol. 28, pp. 339–358, 1983.
[47] H. Trice and J. Beyer, "Studying organizational cultures through rites and ceremonials," *Acad. Manage. Rev.*, vol. 9, pp. 653–669, 1984.
[48] D. Truex, R. Baskerville, and H. Klein, "Growing systems in emergent organizations," *Commun. ACM*, vol. 42, pp. 117–123, 1999.
[49] G. M. Weinberg, *Quality Software Management*. New York: Dorset House , 1992–97, vol. 1 (1992), 2 (1993), 3 (1994), 4 (1997).
[50] K. E. Wiegers, *Creating a Software Engineering Culture*. New York: Dorset House, 1996.
[51] I. Aaen, J. Arendt, L. Mathiassen, and O. Ngwenyama, "A conceptual MAP of software process improvement," *Scandinavian J. Inform. Syst.*, vol. 13, 2001.

Ojelanki Ngwenyama received the Diploma from Humber College, Toronto, ON, Canada, the B.G.S and M.S. degrees from Roosevelt University, Chicago, IL, the M.B.A. degree from Syracuse University, Syracuse, NY, and the Ph.D. degree in computer science from The Thomas J. Watson School of Engineering, State University of New York, Binghamton.

He is currently a Professor of information systems and the Director of the International Scholars Program at Virginia Commonwealth University, Richmond, VA. He is also an Extraordinary Professor in the Department of Informatics and a Member of the Board of Directors of the School of Information Technology, University of Pretoria, Pretoria, South Africa; a Research Professor with University of Jyväskylä, Finland; and a Visiting Research Professor with Aalborg University, Aalborg, Denmark. His research focuses on organizational issues of information technology and software development and use. He is a coauthor (with L. Mathiassen and J. Pries-Heje) of the book *Learning To Improve: Software Process Improvement In Practice* (Reading, MA: Addison-Wesley, 2001). He is currently an Associate Editor for *MISQ* and a member of the Advisory Board of the*Scandinavian Journal of Information System*. His papers have appeared in a wide range of international journals.

Dr. Ngwenyama has been a member of IFIP Working Group 8.2 since 1986.

Peter Axel Nielsen received the M.Sc. degree in computer science from Århus University, Århus, Denmark, and the Ph.D degree in information systems from Lancaster University, Lancaster, U.K.

He is currently an Associate Professor of computer science at Aalborg University, Aalborg, Denmark. He is also a Visiting Research Professor with Agder University, Kristiansand, Norway, and a Visiting International Research Scholar with Virginia Commonwealth University, Richmond, VA. His research interests include action research on software development practice and development of object-oriented methodologies, software engineering, and social and organizational aspects of software development. Currently, he is engaged in research on software process improvement.

Dr. Nielsen is a Member of the IEEE Computer Society and a Member of the IFIP Working Group 8.2.