

The Frameworks Quagmire, A Brief Look

Sarah A. Sheard

Software Productivity Consortium
2214 Rock Hill Rd, Herndon VA 22070
sheard@software.org (703) 742-7106

Abstract. Organizations wishing to remain competitive often want to comply with all possible contractor evaluation criteria, process models, and quality standards, but the field is evolving rapidly. This paper describes the categories of compliance frameworks and characteristics of seven important frameworks: the CMM, SE-CMM, IPD-CMM, ISO 9000, SDCE, MIL-STD-498, and Trillium. The paper also discusses trends and recommendations for how to deal with the Frameworks Quagmire.

INTRODUCTION

More and more, software and system

developers are discovering that their ability to win and perform on contracts is subject to investigations of their processes as well as the quality, cost, or effectiveness of their products. The frameworks against which their processes are evaluated are multiplying, as shown in Figure 1.

In the late 1980s the CMMsm for Software [CMM] was created to help software developers mature their software development processes, to better meet contractual requirements. Recently, the engineering community presented two capability models ([SE-CMM] and [SECAM]) for companies to use in improving their systems engineering processes.

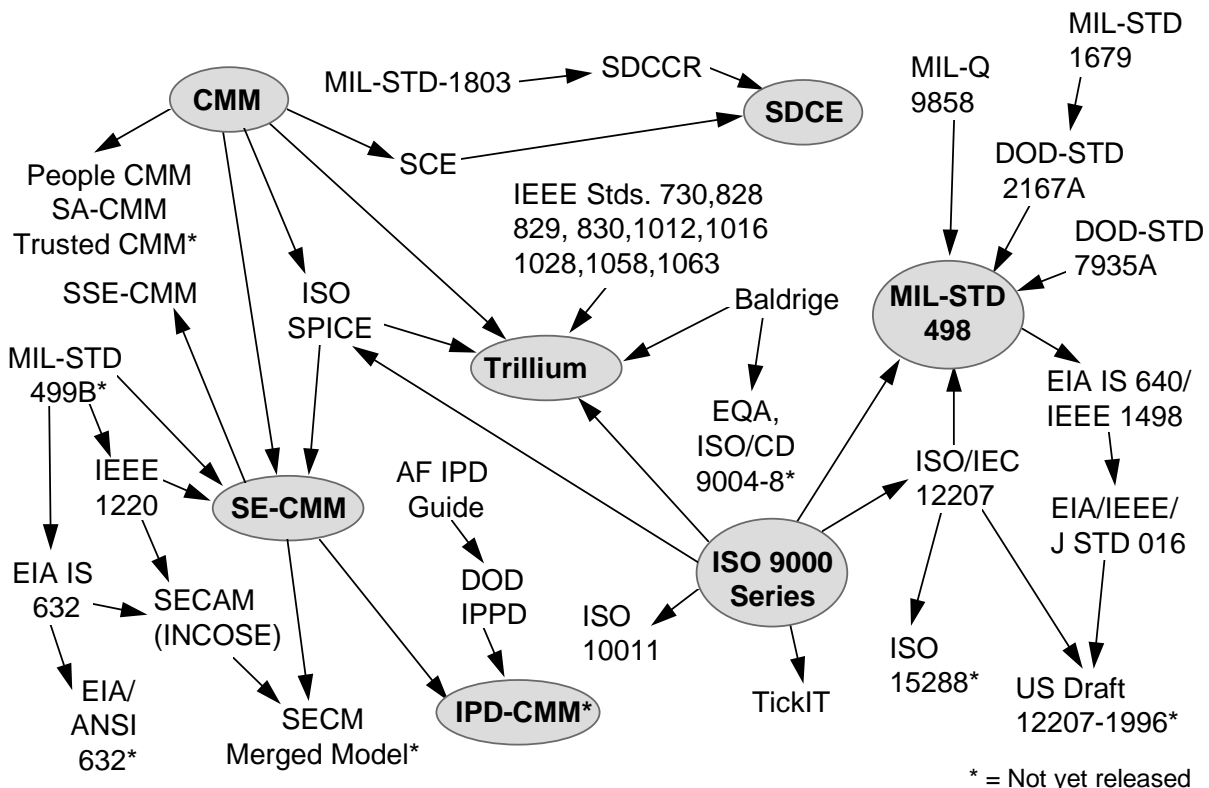


Figure 1. The Frameworks Quagmire

Meanwhile, the international community developed several different groups of process improvement and quality standards ([ISO 9000] and ISO's [SPICE]), and the military developed ways to evaluate bidders during source selection, such as the [SDCE].

Even now, new standards are appearing regularly. Asterisks in Figure 1 indicate frameworks that have not yet been publicly released. Omitted from this chart were efforts, such as the Testing Maturity Model, documented in [Burnstein 96], that were not driven by large, standards-setting or professional groups. The set of available frameworks is large and confusing, at best.

Clearly, organizations need help determining which standards and other frameworks are most beneficial. As a consortium of member companies, the Software Productivity Consortium has studied the compliance frameworks that are relevant to companies building software-intensive systems. This paper highlights some of the main points of a Consortium course on Compliance Frameworks, including framework types, characteristics, trends, and recommendations.

A MULTITUDE OF FRAMEWORKS

From Figure 1 it is evident that many, many frameworks exist which developers may need to consider. The field is truly a quagmire, in which process improvement efforts can bog down, if an organization is not careful.

The arrows in the Figure 1 show the usage of one framework in developing another. For example, the Systems Engineering Capability Maturity Model (SE-CMM) of EPIC¹ developed from the Capability Maturity Model (CMM)^{2,3} for Software, the International

Organization for Standardization (ISO) Software Process Improvement Capability dEtermination (SPICE), MIL-STD-499B (draft), and the Institute of Electrical and Electronics Engineers standard for systems engineering [IEEE 1220]. The SE-CMM was subsequently used in creating the Integrated Product Development CMM [IPD-CMM], the Security Systems Engineering CMM (SSE-CMM) [Hefner 96], and a merged systems engineering capability model (SECM) that is currently being developed with facilitation from the Electronics Industries Association (EIA).

TYPES OF COMPLIANCE FRAMEWORKS

The first step toward making sense of the Quagmire is to categorize the frameworks by purpose. One or more of the six categories in Table 1 apply to most of the frameworks.

- | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Standards and Guidelines 2. Process Improvement (PI) Models and Internal Appraisal Methods 3. Contractor Selection Vehicles 4. Quality Awards 5. Software Engineering Life-Cycle Models 6. Systems Engineering Models |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Table 1. Compliance Framework Categories

1. Standards and Guidelines

Standards and guidelines establish what must be done in a contractual situation. Most can be tailored as desired by both parties, and are used as recommendations of good practices in general. Guidelines may interpret associated standards or collect practices not intended to be specified in a contract.

Standards include:

¹ Enterprise Process Improvement Collaboration, a collaboration of Industry, Government, and Academic institutions.

² "CMM", and other names and acronyms including "Capability Maturity Model" and "CMM," are service marks of Carnegie-Mellon University.

³ The acronym "CMM," when used alone, refers to the CMM for Software.

- United States (US) Military Standards, such as MIL-STD-498 (Software Development and Documentation), and guidelines, such as the Air Force guidelines for Integrated Product Development
- Commercial Standards, such as [EIA IS 632], an interim standard on the Systems Engineering Process
- International Standards, such as the [ISO 9000] series for Quality Systems (which includes guidelines as well as standards)

2. Process Improvement Models and Internal Appraisal Methods

In general, these frameworks define the characteristics of good processes, and avoid prescribing how the processes must be enacted. The purpose of process improvement models is to establish a roadmap by which a route can be drawn from “where we are today” to “where we want to be.” In order to determine “where we are today,” an organization performs an appraisal, sometimes with the aid of an outsider with specific expertise in the model. These models include:

- CMM relatives, including the CMM for Software, the SE-CMM, IPD-CMM, People CMM [P-CMM], and others
- SECAM
- The [Trillium] model, for telecommunications

3. Contractor Selection Vehicles

These frameworks specify the examination of an organization’s processes by an outsider, either a second-party (the potential acquirer) or a third party, usually one hired by the potential acquirer. The purpose is a detailed comparison of competitors’ strengths and weaknesses, in order to minimize procurement risk to the acquiring agency. Two methods in use are:

- Software Capability Evaluation (SCE) (associated with the CMM)
- Software Development Capability Evaluation (SDCE), from the US Air Force

4. Quality Awards

In 1987, the US Government established an award program, the Malcolm Baldrige National Quality Award [Baldrige] to improve American businesses’ focus on quality and customer satisfaction. Subsequently, Europe established an award with very similar criteria and selection methodology, and recently ISO has begun a draft standard with identical categories to the Baldrige’s seven categories. Quality awards include:

- Malcolm Baldrige National Quality Award
- European Quality Award
- ISO/CD 9004-8, draft standard on Quality Management Principles

5. Software Engineering Life-Cycle Models

MIL-STD-498 developed from DOD-STD 2167A (for software development), DOD 7935A (for documentation), and MIL-Q-9858 (for quality). Approved after the official date of “no more military standards” in 1993, this standard was intended to be an interim standard until commercial standards replaced it in about two years. [ISO/IEC 12207] is an international standard in the same area, and MIL-STD-498 is being adapted (in several steps) to add ISO/IEC 12207-type requirements on acquirers, maintainers, and operators, parties not mentioned in MIL-STD-498. Life-cycle models include:

- MIL-STD-498
- EIA/IEEE J STD 016 [J STD 016]
- ISO/IEC 12207
- US Draft 12207-1996

6. Systems Engineering Models

The SE-CMM was developed in 1994 by half- or full-time authors working on the model for a year. Funding was provided by the authors’ companies, as a provision of their participation in EPIC, then called the Industrial Collaboration. The model was completed in a year, and revised, along with an

accompanying appraisal method, in the next eighteen months.

At the same time, an INCOSE working group developed the Systems Engineering Capability Assessment Model (SECAM) from several systems engineering assessment models used internally by companies in the aerospace and defense industry.

One primary difference is that the SE-CMM was first developed as a model, and the SECAM as an assessment method, intended to work with systems engineering standards (from the EIA and IEEE) serving as the model. Another difference is that the SE-CMM confines its scope to process characteristics, and the SECAM includes non-process characteristics, such as work quality and systems engineering team experience. Finally, the SE-CMM considers all practices as “base practices,” whose performance in an informal manner would earn the organization a rating of 1 in the process area, but the SECAM allowed some practices to be required only of higher-capability organizations, so that Level 1 organizations need not perform them at all.

In the SECM merger effort being facilitated by the EIA, these differences have been resolved and the models are being merged. Initial public release is scheduled for mid-1997.

[ISO 15288] is an effort to create an international system life-cycle standard, initiated by the group that created the ISO software life-cycle standard, ISO/IEC 12207, and augmented by people with systems engineering expertise. An INCOSE past president, who was also an author of MIL-STD-499B, EIA Interim Standard 632 [EIA IS 632], and IEEE 1220, represents INCOSE in the 15288 effort. Scheduled release is for 2001.

Systems engineering models include:

- MIL-STD-499B (Systems Engineering)
- SE-CMM
- SECAM

- IEEE 1220
- EIA IS 632 and EIA/ANSI 632
- ISO 15288

CHARACTERISTICS OF SEVEN FRAMEWORKS

Table 2 compares characteristics of the seven important frameworks which are circled on Figure 1. Brief notes about the frameworks follow.

CMM for Software

The CMM “nucleus” includes the CMM for Software, the P-CMM, the Software Acquisition CMM [SA-CMM], and the Trusted CMM (T-CMM).

The CMM, P-CMM, and T-CMM address the software development organization within an enterprise. The SA-CMM applies to an acquisition agency. In contrast, the SE-CMM addresses the organization building systems, which will be larger than the software development organization if the systems include hardware and software. The IPD-CMM addresses the product development enterprise, including such groups as marketing, manufacturing, and business management, as well as the development organizations.

SE-CMM

Frameworks centered around the SE-CMM nucleus all involve systems engineering. The systems engineering standards’ definition for systems engineering includes most of the twelve systems engineering roles described in [Sheard 96]. IEEE 1220 takes primarily a Technical Management view, while EIA IS 632 leans more toward requirements development and system design. The latter two are apparently being consolidated now, into the anticipated EIA/ANSI 632.

IPD-CMM

The IPD-CMM is being created by EPIC, which also wrote the SE-CMM. The authors used the Air Force IPD guide and the DOD Guide to Integrated Product and Process

Framework	Scope	Purpose	Length, pages	Major Focus	Notes
CMM for Software	Software developing organization	PI	500	SW Process	Staged architecture provides “Triptik” (a AAA map with exact roads, stops, and times highlighted) for improvement.
SE-CMM	Organization developing systems	PI	250	SE Process	Continuous architecture provides map of terrain.
IPD-CMM	Enterprise	PI	220	Process	Staged-continuous architecture provides map plus “Triptik.”
ISO 9000	Product producing organization	Trade	16	Quality process	Registration certifies a minimum quality system compliance.
SDCE	Bidding organization	Contractor selection	600	Process, Capacity, Technology	Evaluates risks to acquirer for each bid, and reduces risks with winning contractor.
Software Life Cycle Standards	Software developing organization	Contract compliance	60-200	Management process	Standards are evolving to include role of acquirer and others, as well as supplier.
Trillium	Enterprise	PI	130	Process	Combines requirements from CMM, ISO, Baldrige, and software quality standards.

Table 2. Characteristics of Seven Frameworks

Development [DOD IPPD] as input, and visited organizations practicing IPD well to verify principles and practices.

A goal of the IPD-CMM is to establish a framework into which other CMMs can fit. Its architecture is a hybrid of the staged and continuous models. Implementation of the integration depends, in part, on the existence of minimal “plug-and-play” models in specific domains. These models would be combined with the basic product development, integration, or supporting processes already covered by the IPD-CMM, to form a model that is appropriate for the specific organization using it.

ISO 9000

Despite the fact that ISO 9000 registration is for the purpose of trade, the quest for registration often has the same effect on the organization as the quest for a CMM level (generally Level 2 or Level 3): the organization’s processes are documented, and discipline is tightened to ensure documented processes are followed. Thus, although in theory, ISO addresses only “quality-impacting” processes,

this definition is broad enough that virtually all the CMM and SE-CMM processes will fall under the ISO 9000 umbrella.

TickIT, a program described by a roughly 75-page document, was developed by the British Standards Institute. TickIT provides detailed ISO 9000 guidance for software development. TickIT registration is generally optional, except in the United Kingdom.

The ISO 10011 series of standards specifies requirements and training for ISO 9000 auditors.

SDCE

The SDCE evolved from the Air Force’s Software Development Critical Capacity Review (SDCCR) and from the SCE, which is an assessment method associated with the CMM for Software. To CMM-type questions, the SDCE adds several areas of focus, including systems engineering and technology (such as artificial intelligence). The SDCE’s technology areas, in particular, may not apply on all procurements, and the SDCE is intended to be tailored by deleting up to half of the questions for any one procurement.

Software Life-Cycle Standards

The standards specify life cycles, including reviews that approve moving from one phase to the next. These standards provide processes, in contrast to CMMs, which provide requirements that good processes will meet.

Initial modification of MIL-STD-498 to become more commercial proceeded under the numbers EIA IS 640 and IEEE 1498. These numbers were retired before the standard was released, and the standard was given a J series number [J STD 016], meaning a Joint standard between the two organizations. The current number for the expected commercial version is US Draft 12207-1996.

Trillium

The Trillium model, created by Bell Canada, combines requirements from the ISO 9000 series, the CMM for Software, and the Malcolm Baldrige criteria, with software quality standards from the IEEE. The model cannot necessarily be adopted as-is because in some cases, goals of the frameworks are used rather than their detailed requirements, and because the model includes process information that is unique to the telecommunications field. However, the model serves as proof that the requirements of several of the popular frameworks can be combined, and it provides a template for additional efforts in this area.

FRAMEWORKS TRENDS

Four trends are evident to those following the field of compliance frameworks.

- Evolution
- Proliferation
- Integration and Coordination
- Consolidation

Evolution

Current models are being improved and adapted to better meet stakeholder needs. Professional standards and the CMMs all have comment-logging and solution-approval

processes, and are often updated on a predetermined schedule. This is good in that the models need to be improved, and in that improvements can include better integration with other models. However, any changes must be read, understood, and responded to by anyone who has used the old model. Because there are over 2000 organizations with investment in the Software CMM, and over 10,000 with investment in the ISO 9000 series, the inertia against change can be considerable.

Proliferation

More models are continually being developed. The T-CMM and SSE-CMM are examples of recent additions.

Proliferation is both good news and bad news. It is good in that new models capture wisdom and best practices, because developers have been seeking “best practices” for years.

But implementing best practices in a real organization is at least as hard as collecting lessons from the groups who have learned them the hard way. Someone depositing a new 300-page model on a process engineer’s desk is not likely to walk away feeling appropriately appreciated. Time must be devoted to reading new models and to understanding changes to existing models as they emerge. If a new model is not seamlessly and obviously integratable with current frameworks, half a staff-year can easily go into determining what, if anything, a new model suggests should be done differently.

Integration and Coordination

As mentioned above, the IPD-CMM provides a framework for future integration of CMMs. In addition, the SEI is sponsoring the CMM Integration effort, which may impose (on CMM revisions) requirements that will make model integration easier for users. The SEI is also coordinating with other model makers, notably ISO 9000 (at least keeping up

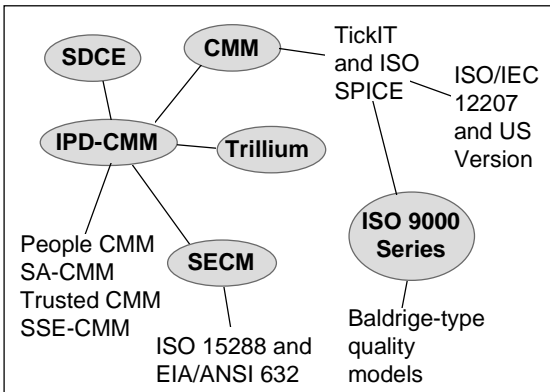


Figure 2. A Better Frameworks Quagmire

with them and publishing comparisons between the models) and ISO SPICE. Further, as various frameworks evolve, authors are reading other frameworks and incorporating the best features. It is reasonable to expect fewer differences in the next cycle of all the models.

Consolidation

Retirement of multiple models as they are consolidated into a single new model is a very positive trend. This appears to be happening with software life-cycle standards and, to a smaller extent, systems engineering standards and models.

Figure 2 shows an idealized picture of what the Quagmire should look like if similar frameworks are consolidated. Each of the lines would indicate a defined interface, for a defined purpose. Frameworks listed together would be consolidated. Even more consolidation may occur with the CMMs and the SECM.

RECOMMENDATIONS FOR CREATORS OF FRAMEWORKS

Clearly, those writing standards, process models, and contractor selection vehicles need to understand the predicament of developers. While there certainly is a need for well-crafted collections of best practices, creators of frameworks must take note of the frameworks that already exist, and must tailor their additions to fit into some of them. How should organizations integrate compliance

with a new model and compliance with other frameworks?

RECOMMENDATIONS FOR DEVELOPERS

Cost competitiveness and time-to-market dominate the factors that will keep companies alive, not to mention profitable, in the 1990s and beyond. But the act of defining and implementing process changes costs significant money – how can this be efficient?

The key lies in adopting only a few, high-leverage frameworks. In addition to the CMM for Software (which most have already adopted), the Consortium's member companies are generally considering the SE-CMM, ISO 9000, and the SDCE (when bidding on Air Force contracts). Many members already have contracts requiring compliance with MIL-STD-498 or its predecessors, so they are looking to the future, when MIL-STD-498 will be merged with ISO/IEC 12207. Members are also looking to the IPD-CMM to provide a needed integration framework.

It is recommended that developers delay implementation of most of the other frameworks. Some will disappear from lack of support. Others, which may prove long-lived, can be adopted after better integration methods have been made available.

CONCLUSIONS

Companies should focus on identifying a small set of high-value frameworks to adopt. Those already working with the CMM for Software may add the SE-CMM and the IPD-CMM to help broaden their process improvement effort, and may delay adopting other CMMs until methods for integrating CMMs are better defined.

Other frameworks worth investigating are the ISO 9000 series of standards, the SDCE (if bidding on Air Force contracts), Trillium (particularly for telecommunications companies), and the IPD-CMM.

Organizations with frameworks questions or problems in this area should consider joining with other industrial, academic, and government institutions, in order to leverage their efforts in navigating the Frameworks Quagmire.

REFERENCES

Square brackets in the text denote references, as shown below. Because most frameworks are written by institutions rather than individual authors, the framework documents are cited by acronym. Other articles are cited by first author and year. WWW references are included where practical.

- [Baldrige] National Institute of Standards and Technology, Malcolm Baldrige National Quality Award, see www.nist.gov/director/quality_program/.
- [Burnstein 96] Burnstein, Ilene, Taratip Suwannasart, and C. R. Carlson, "Developing a Testing Maturity Model," Parts I and II., *Crosstalk*, August and September, 1996.
- [CMM] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, *Capability Maturity Model for Software*, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24, February 1993.
- DOD IPPD] *DoD Guide to Integrated Product and Process Development*, February 5, 1996, OUSD(A&T)/DTSE&E; ATTN: Mr. Mark D. Schaeffer, 3110 Defense Pentagon, Washington DC 20301.
- [EIA IS 632] Electronics Industry Association, *EIA IS 632, Draft For Technical Committee Review, Systems Engineering*, 20 Sept. 1994.
- [Hefner 96] Hefner, Rick, David Hsiao, and Warren Monroe, "Experience with the Security Systems Engineering CMM," *Proceedings of INCOSE*, 1996.
- [IEEE 1220] IEEE Computer Society, *IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process*, IEEE Std 1220-1994, New York, 28 February 1995.
- [ISO 9000] International Organization for Standardization, *ISO 9000 International Standards for Quality Management*. Switzerland, 1991. See www.iso.ch or www.exit109.com/~leebee/.
- [ISO 15288] Lake, Dr. Jerome G., "ISO Standard 15288, System Life Cycle Processes," *INSIGHT*, Spring 1997.
- [ISO/IEC 12207] *Information technology -- Software life cycle processes*, International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 12207:1995 (E), 1 Aug 95.

- [IPD-CMM] Software Engineering Institute and EPIC, *An Integrated Product Development Capability Maturity Model*, Carnegie Mellon University, Version 0.9, 28 October 1996.
- [J STD 016] Electronics Industry Association, *EIA/IEEE/J STD 016, Standard for Information Technology - Software Life Cycle Processes - Software Development - Acquirer-Supplier Agreement*, 6 February 1996.
- [P-CMM] Curtis, Bill, William E. Hefley, and Sally Miller. *People Capability Maturity Model*. Software Engineering Institute, CMU/SEI-95-MM-02, September 1995.
- [SA-CMM] *Software Acquisition Capability Maturity Model*. See www.sei.cmu.edu/technology/risk/Risk_SW_Acq/SA-CMM.html.
- [SDCE] Software Development Capability Evaluation, see www.wpafb.af.mil/indguide/sdce.htm.
- [SECAM] INCOSE Capability Assessment Working Group, *Systems Engineering Capability Assessment Model*, Version 1.50, June 1996.
- [SE-CMM] EPIC, *A Systems Engineering Capability Maturity Model*, Version 1.1. Available from <http://www.software.org/secmminfo.html>.
- [Sheard 96] "Twelve Systems Engineering Roles," *Proceedings of INCOSE*, 1996.
- [SPICE] International Organization for Standardization, *SPICE: Software Process Improvement Capability dEtermination*; see www.sqi.cit.gu.edu.au/spice/.
- [Trillium] Bell Canada, Northern Telecom, and Bell-Northern Research. *The Trillium Model*. See <http://ricis.cl.uh.edu/trillium/trillium.html>.

AUTHOR BIOGRAPHY

Sarah A. Sheard has seventeen years' experience in systems engineering. Ms. Sheard worked as a satellite engineer at Hughes Aircraft Space and Communications Group, and in software systems at the Federal Systems group of IBM and Loral. Currently she coordinates systems engineering efforts at the Software Productivity Consortium in Herndon, Virginia, where she also develops technical products and consults and teaches in the areas of systems engineering, process improvement, and integrated product teams. Ms. Sheard received an MS in chemistry from the California Institute of Technology in 1979.