AN EVALUTION OF MODERN LEARNING METHODS

Capers Jones

Version 17.0 May 7, 2012

Abstract

Technical and economic factors are exerting a strong impact on the learning methods used by software engineers and other technical occupations. Huge volumes of information are now available on the web and in e-book formats. The lingering recession and steadily escalating travel costs make live education increasingly difficult to cost justify.

This paper is updated every two years and evaluates the status of software education and the channels for learning new skills. Currently there are 17 learning channels for software related topics.

As 2012 the software engineering world employs more than 100 occupation groups and uses more than 250 methods, tools, and practices. The diversity of occupations and educational topics presents a challenge for both universities and also for keeping technical workers current.

This version introduces the concept of a possible future 18th learning channel. This new channel would be a virtual software engineering university. The university would look like a real university and would be license the rendering engine from a commercial game to create a realistic campus environment.

HOW SOFTWARE PERSONNEL LEARN NEW SKILLS: AN EVALUATION OF MODERN LEARNING METHODS

INTRODUCTION

The viral expansion of information available on the web combined with the very rapid increase in e-book materials is changing the way humans and software personnel can learn new skills. Today in 2012 technical webinars are available essentially every day on every software topic.

The financial meltdown of 2008 and 2009 followed by the lingering global recession has made live instruction difficult to cost justify. Low-cost methods such as e-learning are expanding rapidly. High-cost methods such as live conferences and classroom training are declining rapidly. Fortunately the technologies associated with e-learning are at the point that their effectiveness is increasing.

The rate of change of software technology is extremely rapid. New programming languages appear almost monthly. New programming tools appear almost daily. New software development methodologies appear several times a year.

The fast rate of software technology change means that software professionals are faced with a continuing need to learn new skills. What channels are available to software professionals for learning new skills? How good are the available ways of learning, and what new ways are likely to occur?

Even as software learning methods improve, there are a number of critical topics where software education lags far behind what is truly needed to achieve professional status for software. The most glaring omissions include:

1. Software security practices for building low-risk applications
2. Software quality control practices for minimizing delivered defects
3. Software measures and metrics for effective economic analysis
4. Software architecture for optimizing use of reusable components
5. Software methods for effective renovation of legacy applications
6. Software requirements remain incomplete, ambiguous, and error prone

These gaps and omissions need to be quickly filled if software is to evolve from an art form into a true engineering discipline.

The global economic downturn that started in 2008 and 2009 reduced live software educational opportunities. The increase in fuel and travel costs still continue into 2012 and indeed may never come down. Attendance at live conferences has gone down, some in-

and webinars are likely to be the major education channel for professional education.

Even newer methods such as virtual environments, avatars, and data mining are likely to grow as the economy remains sluggish. Over and above their low costs, these newer methods may offer actual advantages as learning approaches.

The Evolution of Software Learning Channels

The world of software is evolving new technologies as rapidly as any industry in human history. This means that software professionals are faced with a need to acquire new knowledge and new skills at a very rapid clip.

As of 2012 the United States currently employs about 2,600,000 personnel in the technical areas of programming or software development and maintenance, about 280,000 software managers, and perhaps another 1,100,000 ancillary personnel in related specialties such as software sales, customer support, software technical writing, and many others. Table 1 shows the approximate U.S. software population circa 2012:

Table 1: Approximate U.S. Software Population Circa 2012

	Industry	
1	Government - federal civilian	300,000
2	Banks - commercial	275,000
3	Manufacturing - defense	235,000
4	Government - military	175,000
5	Government - state	155,000
6	Government - municipal	150,000
7	Software (commercial)	100,000
8	Insurance - medical	85,000
9	Insurance - property and casualty	65,000
10	Wholesale	65,000
11	Banks - investment	60,000
12	Manufacturing - general	60,000
13	Retail	60,000
14	Consulting	57,500
15	Insurance - Life	55,000
16	Manufacturing - pharmaceuticals	40,000

20	Government - intelligence	30,000
21	Software (outsourcing)	28,000
22	Stock/commodity brokerage	27,500
23	Agriculture	25,000
24	Social networks	23,000
25	Credit unions	22,000
26	Manufacturing - electronics	20,000
27	Manufacturing - automotive	20,000
28	Telecommunications operations	20,000
29	Public utilities - electricity	20,000
30	Construction	20,000
31	Public utilities - water	17,500
32	Oil extraction	17,500
33	Manufacturing - medical devices	15,000
34	Transportation - airlines	15,000
35	Entertainment - television	15,000
36	Government - police	15,000
37	Manufacturing - chemicals	15,000
38	Automotive sales	15,000
39	Government - county	15,000
40	Hospitals	12,500
41	Automotive repairs	12,500
42	Games - computer	12,000
43	Process control	12,000
44	Entertainment - films	12,000
45	Manufacturing - appliances	12,000
46	Professional support - medicine	10,000
47	Professional support - law	10,000
48	Manufacturing - apparel	10,000
49	Education - secondary	10,000
50	Food - restaurants	10,000
51	Education - University	9,500
52	Open source development	7,500
53	Publishing (books/journals)	7,500
54	Natural gas generation	7,500
55	Hotels	7,500
56	Mining - metals	7,500
57	Education - primary	6,000
58	Entertainment - music	5,000

63	Transportation - truck	3,500
64	Transportation - bus	3,000
65	Games - traditional	3,000
66	Real estate - residential	2,600
67	Transportation - trains	2,500
68	Waste management	1,500
	TOTAL/AVERAGES	**2,651,100**

The U.S. total is probably over 3,800,000 professionals if all software-related occupations are considered including marketing and sales personnel, customer support, and other forms of engineers who produce software such as telecommunications and automotive engineers.

The European total is slightly larger that the United States, and the world total is approaching 18,000,000. Exact counts are not available for India, China, and Russia but these three countries combined probably are equivalent to the U.S. total and software work is growing very rapidly in all three. Globally, all software personnel need constant refreshment to stay current. Table 2 shows the approximate global software populations circa 2012:

		National Population from CIA Fact Book in 2011	Estimated Software Population in 2012
1	United States	313,232,044	2,631,149
2	China	1,336,718,015	2,406,092
3	India	1,189,172,206	2,140,510
4	Japan	126,475,664	986,510
5	Brazil	203,429,773	976,463
6	Russia	138,739,892	943,431
7	Germany	81,471,824	635,480
8	France	65,312,249	509,436
9	United Kingdom	62,698,362	489,047
10	Indonesia	245,613,043	442,103
11	Mexico	113,724,226	432,152
12	Turkey	78,785,548	378,171
13	Italy	61,016,804	353,897
14	Pakistan	187,342,721	337,217
15	Iran	77,891,220	295,987
16	Philippines	101,833,938	285,135
17	South Korea	48,754,767	282,778
18	Thailand	66,720,153	253,537
19	South Africa	49,004,031	235,219
20	Canada	34,030,589	231,408
21	Egypt	82,079,636	229,823
22	Spain	46,754,784	224,423
23	Ukraine	45,134,707	216,647
24	Nigeria	155,215,573	186,259
25	Poland	38,441,558	184,519
26	Colombia	44,725,543	169,957
27	Viet Nam	90,549,390	162,989
28	Argentina	41,769,726	158,725
29	Bangladesh	158,570,535	158,571
30	Australia	21,766,771	148,014
31	Malaysia	28,728,607	137,897
32	Algeria	34,994,937	132,981
33	Netherlands	16,847,007	131,407
34	Iraq	30,399,572	115,518
35	Burma	53,999,904	97,200
36	North Korea	24,457,492	92,938
37	Taiwan	23,071,779	87,673
38	Belgium	10,431,477	81,366
39	Saudi Arabia	26,131,703	73,169
40	Austria	8,217,280	64,095

44	Portugal	10,760,305	51,649
45	Israel	7,473,052	50,817
46	Venezuela	27,635,743	49,744
47	Peru	24,248,933	43,648
48	Syria	22,517,750	40,532
49	Finland	5,259,250	35,763
50	Norway	4,491,849	35,036
51	Hong Kong	7,122,508	34,188
52	Denmark	5,529,888	32,073
53	Cuba	11,087,337	31,045
54	Hungary	10,011,000	28,031
55	New Zealand.	4,290,347	24,884
56	Jordan	6,508,271	24,731
57	Singapore	4,740,737	22,756
58	Ireland	4,670,996	22,421
59	Tunisia	10,629,147	19,132
60	Libya	6,597,060	11,875
61	Panama	3,460,462	9,689
62	Lebanon	4,143,101	7,458
63	Costa Rica	2,576,562	7,214
64	Kuwait	2,595,628	4,672
65	Bahrain	1,214,705	2,186
66	Iceland	311,058	1,493
	AVERAGE/TOTAL	**5,779,460,867 Total**	**18,873,814 Total**

Not only are there millions of software personnel, but the industry has followed the path of older occupations and splintered knowledge work into a very wide collection of occupational specialties. Table 2 shows the current occupation groups noted during a study of software occupations in major organizations commissioned by AT&T:

Table 3: Software Specialization Circa 2012

1. Accounting/Financial Specialists
2. Agile coaches
3. Architects (Software)
4. Architects (Systems)
5. Architects (Enterprise)
6. Assessment Specialists
7. Audit Specialists
8. Baldrige Award Specialists
9. Baselining Specialists
10. Benchmarking Specialists
11. Business analysts (BA)

16. CMMI Assessors
17. Complexity Specialists
18. Component Development Specialists
19. Configuration Control Specialists
20. Cost Estimating Specialists
21. Consulting Specialists
22. Curriculum Planning Specialists
23. Customer Liaison Specialists
24. Customer Support Specialists
25. Data Base Administration Specialists
26. Data Center Support Specialists
27. Data quality Specialists
28. Data Warehouse Specialists
29. Decision Support Specialists
30. Development specialists
31. Distributed Systems Specialists
32. Domain Specialists
33. Earned Value Specialists
34. Education Specialists
35. E-Learning Specialists
36. Embedded Systems Specialists
37. Enterprise Resource Planning (ERP) Specialists
38. Executive Assistants
39. Frame Specialists
40. Expert-System Specialists
41. Function Point Specialists (certified)
42. Generalists (who perform a variety of software-related tasks)
43. Globalization and Nationalization Specialists
44. Graphics Production Specialists
45. Graphical User Interface (GUI) Specialists
46. Human Factors Specialists
47. Information Engineering (IE) Specialists
48. Instructors (Management Topics)
49. Instructors (Software Topics)
50. Integration Specialists
51. Intellectual Property (IP) Specialists
52. Internet specialists
53. ISO Certification Specialists
54. Joint Application Design (JAD) Specialists
55. Kanban Specialists
56. Kaizen Specialist s
57. Knowledge specialists
58. Key Process Indicators (KPI) specialists

62. Marketing Specialists
63. Member of the Technical Staff (multiple specialties)
64. Measurement Specialists
65. Metric Specialists
66. Microcode Specialists
67. Model Specialists
68. Multi-Media Specialists
69. Network maintenance Specialists
70. Network Specialists (LAN)
71. Network Specialists (WAN)
72. Network Specialists (Wireless)
73. Neural Net Specialists
74. Object-Oriented Specialists
75. Outsource Evaluation Specialists
76. Package Evaluation Specialists
77. Pattern Specialists
78. Performance Specialists
79. Programming Language Specialists (Java, C#, Ruby, PHP, SQL, etc.)
80. Project Cost Analysis Specialists
81. Project managers
82. Project Office Specialists
83. Project Planning Specialists
84. Process Improvement Specialists
85. Productivity Specialists
86. Quality Assurance Specialists
87. Quality function deployment (QFD) Specialists
88. Quality Measurement Specialists
89. Rapid Application Development (RAD) Specialists
90. Research Fellow Specialists
91. Reliability Specialists
92. Repository Specialists
93. Reengineering Specialists
94. Requirements engineer
95. Reverse engineering Specialists
96. Reusability Specialists
97. Reverse Engineering Specialists
98. Risk Management Specialists
99. Sales Specialists
100. Sales Support Specialists
101. Scrum masters
102. Security Specialists
103. Standards Specialists
104. Systems Analysis Specialists

108. Test Case Design Specialists
109. Testing Specialists (Automated)
110. Testing Specialists (Manual)
111. Testing Specialists (Model Driven)
112. Total Quality Management (TQM) Specialists
113. Virtual Reality Specialists
114. Web Development Specialists
115. Web Page Design Specialists
116. Web Masters

The combination and permutations of educational topics and software occupation groups totals to about 58,000 potential courses if each major subject were to be customized for specific occupation groups.

The financial crisis and recession of 2008 and 2009 and the lingering impact in 2012 have disrupted the normal channels of software education. To conserve funds, many companies have cut back on training expenses.

A significant number of software personnel may have lost their jobs due to downsizing or to actual bankruptcy of their companies. As a result attendance at conferences diminished between 2008 and 2012, as did attendance at commercial education classes. The effect of a long recession on university and graduate school education is uncertain.

Webinars and on-line educational channels have to expanded due in large part to their low costs for both sponsors and students. Indeed webinars are exploding so rapidly that there is a need for a central catalog of all webinars, organized by topic. On any business day no fewer than 50 software webinars are being offered in the United States, and no doubt this number will soon rise into the hundreds.

It is conceivable that the recession may cause significant new research into hyper-modern training methods such as virtual reality “classrooms,” text mining to convert paper documents into web-accessible documents, and shifting information from paper form into e-book and web-accessible form.

When this report was first produced in 1995 there were only 10 major channels for software personnel to acquire new information. These channels varied in effectiveness and costs:

1) In-house education
2) Commercial education
3) Vendor education
4) University education
5) Self-study from work books
6) Self-study from CD-ROMs or DVDs
7) Live Conferences
8) On-line education via the Internet and World Wide Web
9) Books
10) Journals

Today in 2012 there are now 17 channels available for software personnel to acquire new information. New forms of electronic books, web browsing, webinars, simulation web sites such as "Second Life" have been added to the suite of learning methods.

The current report uses a new way of evaluating learning methods. Each method is ranked in terms of four factors using a scale from 1 (best) to 10 (worst):

1) Cost
2) Learning Efficiency
3) Learning Effectiveness
4) Currency of information.

The number "1" is the top rank or score for each category. All of the available methods are then listed in descending order for each topic.

The first factor or "*cost"* does not need much explanation. It simply refers to the expenses a student is likely to have to use the learning channel. Costs range from almost free for activities such as web browsing to extremely expensive, such as attending a major university or going to graduate school.

The second factor or "*learning efficiency*" refers to the amount of calendar time required to impart a given amount of new knowledge to students. A score of "1" indicates the most efficient form of learning. On-line education and web-browsing are the quickest methods of learning.

The third factor or "*learning effectiveness*" refers to the volume of information that the learning channel can transmit to a student. A score of "1" indicates the most effective form of learning. Live instructors in universities and in-house training transmit more information than any other channels.

The fourth factor or "*currency*" refers to the average age of the information that is being

Following are the 17 channels evaluated in the 2012 versions of this report:

Table 5: Ranking of Software Learning Channels as of Spring 2012

Average Score	Form of Education	Cost Ranking	Efficiency Ranking	Effectiveness Ranking	Currency Ranking
3.00	Web browsing	1	1	9	1
3.25	Webinars/e-learning	3	2	6	2
3.50	Electronic books	4	3	3	4
5.25	In-house training	9	4	1	7
6.00	Self-study from CD/DVD	4	3	7	10
7.25	Vendor training	13	6	5	5
7.25	Commercial training	14	5	4	6
7.50	Wiki sites	2	9	16	3
8.25	Live conferences	12	8	8	5
9.00	Simulation web sites	8	7	13	8
10.25	Self-study from books	5	13	12	11
10.25	Journals	7	11	14	9
10.75	On-the-job training	11	10	10	12
11.75	Mentoring	10	12	11	14
12.00	Books	6	14	15	13
12.25	Undergraduate training	15	15	3	16
12.25	Graduate training	16	16	2	15

Between 1995 and 2012 two computer-aided forms of learning, web browsing and webinars (e-learning), have not only been added to the list but now have achieved top ranking in the categories of cost, currency, and efficiency. They are still in the middle of the list in terms of effectiveness, however.

The following list is a small sample of current terms, acronyms, and abbreviations that have come into prominence within the software community over the past few years gives a flavor to the kinds of technologies that software personnel need to know about in 2012 that in some cases did not exist even as late as 2005:

Table 6: Technical Topics Requiring Instruction Circa 2012

1) Agile development
2) Android
3) ASP (application service provider – software available via the World Wide Web)
4) Automated static analysis
5) Automated testing
6) B2B (Acronym for "business to business" or business via the World Wide Web)
7) Botnet
8) BPR (business process reengineering)
9) Client-server computing
10) Cloud computing
11) Computer security
12) CMM® (capability maturity model)
13) CMMI® (capability maturity model integration)
14) COSMIC (a European function point variant)
15) Configuration control
16) CRM (Customer Resource Management)
17) Data quality
18) Data warehouses
19) Dot Com (a company doing business via the World Wide Web)
20) E-business (Short for "electronic business")
21) E-learning
22) EA (Enterprise Architecture)
23) ERP (Enterprise Resource Planning)
24) Extreme Programming (XP)
25) Formal design and code inspections
26) Function point metrics
27) GUI (Graphical user interface)
28) Hacking defenses
29) HTML (hypertext markup language)
30) I-CASE (integrated computer aided software engineering)
31) IE (information engineering)
32) ISO (International Standards Organization)
33) ISP (internet service provider)
34) ITIL (Information Technology Information Library)
35) JAD (joint application design)

39) OLE (object linking and embedding)
40) Orthogonal Defect Tracking
41) Pair programming
42) Process improvement
43) PSP (Personal Software Process)
44) RAD (rapid application development)
45) QFD (quality function deployment)
46) RUP (Rational Unified Process)
47) Renovation of legacy applications
48) Reusability
49) SCRUM
50) Security vulnerabilities
51) SOA (Service-Oriented Architecture)
52) Six-Sigma for Software
53) Static analysis
54) Story points
55) Supply-chain integration
56) TCO (total cost of ownership)
57) TQM (total quality management)
58) TSP (Team Software Process)
59) Trusted computing
60) UML (unified modeling language)
61) Use Cases
62) Use Case Points
63) Virtualization
64) Web-based applications
65) Web object points

Each of the topics in this list is relatively new. Each is relatively complicated. How can software personnel stay current with the latest technologies? Even more important, how can software personnel actually learn how to use these new concepts well enough to be effective in their jobs?

Since not every new topic is appropriate for every project, and since some topics are mutually exclusive, it is also important for software engineers and managers to know enough about each to select the appropriate methods for specific projects.

It is an interesting hypothesis that the volume of new topics in the software world is so large that it is one of the reasons why traditional topics tend to fall out of use. For example most personnel are limited in the amount of time that can be spent on training. If there is a choice between a new topic such as “agile development” and a traditional topic such as “design inspections” the new is likely to be chosen over the old.

terials, and the student's abilities to actually acquire new skills.

The rankings in this section are derived from interviews and client benchmark studies which the author and his company have performed. Overall about 600 companies have been visited, including roughly 150 large enough to be included in the Fortune 500 set. About 35 government sites have also been visited, at both state and national levels. More than a dozen of the enterprises visited employed at least 10,000 software personnel. Examples of major software employers include IBM, Microsoft, and Electronic Data Systems (EDS). More than 100 of the companies employed more than 1,000 software personnel.

Software Engineering Specialists who require special knowledge

Major software employers typically try to keep their employees up to speed by offering training and education on a continuing basis. The best in class software employers typically have a pattern that resembles the following:

1. From 4 to 10 weeks of intensive training for new technical employees.
2. Annual in-house training that runs from 5 to 10 days per year.
3. From 1 to 3 external commercial seminars per year.
4. Monthly or at least quarterly "webinars" on newer topics.
5. A growing library of DVD or streaming training courses for self study.
6. A significant library of technical books and journals including e-books.
7. Tuition refund programs for managers and technical staff.
8. Access to web-based technical sites.
9. Subscriptions to software and management journals such as CrossTalk
10. Subscriptions to executive journals such as CIO
11. Subscriptions to information providers such as Gartner Group
12. Increasing use of webinars, e-books, and web-based information

Unfortunately the recession of 2012 is likely to cause significant reductions in many of these methods, as companies scramble to stay solvent.

Because multiple channels of education are available, it is interesting to consider the topics for which each kind of channel is likely to be selected and their strengths and weaknesses. The following educational channels are ranked in terms of their overall scores. In the following ranking "1" is best.

Costs = 1; Efficiency = 1; Effectiveness = 9; Currency = 1; Overall Score = 3.00
Prognosis: Expanding rapidly

Web browsing using search engines such as Google or ASK is now the most rapid and cost-effective method for finding out about almost any technical topic in the world. By using a few choice search phrases such as "software quality" or "software cost estimating" within a few moments literally millions of pages of information will be at hand.

The down side of web browsing is that the information is likely to be scattered, chaotic, and a mixture of good and bad information. Even so, web browsing is now a powerful research tool not only for software engineers, but for all knowledge workers.

There are a number of web "portals" that provide unusually large numbers of links to other relevant sources of information. One of these is the portal of the Information Technology Metrics and Productivity Institute (ITMPI), which provides an extensive fan-out to topics in areas such as software engineering, quality assurance, project management, and maintenance of legacy applications (www.ITMPI.org).

Among academic links one of the most complete is that of Dave W. Farthing of the University of Glamorgan in the United Kingdom (www.comp.glam.ac.uk). This interesting portal has links to dozens of project management sites and the publishers of scores of project management books.

A third portal to useful software topics is the web site of the Software Engineering Institute (SEI), which is somewhat slanted towards the government and defense sectors. However it still covers a host of interesting topics and provides many useful links (www.SEI.cmu.edu).

However consolidation of information by topic, cross indexing, and knowledge extraction remain somewhat primitive.

Number 2: Webinars, "Podcasts," and e-learning
Costs = 3; Efficiency = 2; Effectiveness = 6; Currency = 2; Overall Score = 3.25
Prognosis: Expanding rapidly in usage; improving in effectiveness

Using computers for training has long been possible, but new technologies are making rapid improvements. Within a few years students may be able to take extensive training in virtual environments augmented by many new learning methods.

Webinars are a new kind of seminar that are exploding in popularity. With webinars, the speakers and the attendees are located in their own offices or homes and all use their own local computers. A webinar hosting company connects all of the participants, and also provides back-up support in case any participant gets disconnected. The hosting compa-

they are recorded, then the information is available on demand at any time. Podcasts can also include quizzes and tests, with automatic scoring and rerouting to different parts of the material based on student answers.

In webinars, the primary speaker and also an MC communicate with the attendees by phone, but PowerPoint slides or other computer-generated information appears on the attendee's screens, under the control of the primary speaker or the MC.

Because no travel is involved, the efficiency and cost scores of webinars are quite good. The currency score is also quite good. But as of 2012 the effectiveness is only mid-range but increasing rapidly. Webinars at the moment are primarily used for single-purpose presentations of up to about 90 minutes in duration.

Webinars and podcasts have expanded in numbers so rapidly that on any given day perhaps 50 such webinars are being offered concurrently by various organizations. It is almost impossible to keep track of the numbers of webinars. What would be of value to the industry is a non-profit catalog that would allow companies and universities to post the schedules and abstracts of all webinars for at least two months in advance.

In the future, it can be expected that because of the cost-effectiveness of the webinar and pod-cast approach, entire courses of perhaps 10 to 20 hours in duration will migrate to the webinar method.

At some point virtual environments using avatars will join the mix, at which point e-learning has a good chance of becoming more effective than any other form of training in human history.

Imagine if you will entering a virtual class room with avatars for the other students. Courses could include both static and dynamic materials, such as animated screens. Indeed 3-D images are almost ready to be used for training purposes.

As of 2012 the technology of webinars is still immature. Disconnects, low volumes, and intermittent telephone problems are not uncommon. The VOIP form of telephone calls, for example, often does not work at all. No doubt these issues will be resolved over the next year or two.

It is theoretically possible to envision entire courses taught on a global basis without the students and instructors ever meeting face to face using webinars and computer communications exclusively.

It is only a short step from today's webinars to using a virtual classroom where students and instructors can see each other, either as abstract avatars or using actual images of real people.

small step to extend this technology to hand-held devices or to record the lectures and discussions for later use.

In addition to merely having webinars and then saying goodbye, astute companies will note that the connections to several hundred possible customers might be a good marketing opportunity as well. Attendees can register with the instructor or the company sponsoring the event to receive additional information.

Indeed, it would also be possible to invite webinar attendees to participate in other forms of information transfer such as participating in wiki-sites.

Number 3: Electronic Books (e-books)
Costs = 4; Efficiency = 3; Effectiveness = 3; Currency = 4; Overall Score = 3.50
Prognosis: Expanding in usage; improving in effectiveness; declining in costs.

Electronic books or e-books have been on the periphery of education and entertainment for more than 20 years. In the past electronic books were characterized by dim screens, awkward interfaces, slow downloads, limited content, and general inconvenience compared to paper books or even compared to normal computers whose screens are easer to see.

There have long been organizations such as "Project Gutenberg" that makes books available in HTML, PDA, Word or other web-accessible forms. This trend is now exploding as Google, Microsoft, and other major players are attempting to use automated text-conversion tools to convert almost 100% of hard-copy books into web-accessible formats. Needless to say these attempts have raised serious issues on copyrights, intellectual property, and compensation for authors and copyright holders.

Also, given the huge volumes of on-line text now available, there are also technical issues that need to be improved such as cross-references, abstracts, indexes, inclusive catalogs, and a number of others. However the trend towards web-accessible text is expanding rapidly and will continue to do so.

Interestingly, the legal profession seems to be somewhat in advance of the software profession in terms of concentrating knowledge, cross-referencing it, and making information accessible to professional who need it. The Nexis-Lexis company, for example, has access to more than 5,000,000 documents from perhaps 30,000 sources. There is no equivalent organization that has such a well-organized collection of software information.

More recently Apple, Amazon, Barnes & Noble, and Sony began to change the equation for electronic books with new models that solved most of the problems of older e-books and simultaneously introduced powerful new features.

These new devices have at least an outside chance at transforming not only software learning, but also learning in all other scientific disciplines. The best features of these new devices include excellent screen clarity, very fast downloads (including wireless connections), long battery life, and much improved interfaces. In addition they have some features that are even superior to regular paper books. These include the ability to keep permanent annotations, to skip from book to book on the same topic, and to get frequent updates as new materials become available from publishers.

As software learning tools, it would be easy to select and download the top 10 books on software project management, the top 10 books on software quality, the top 10 books on software maintenance, and the top 10 books on software cost estimating, the top 10 books on software development, and have all of them present and available simultaneously.

For colleges and universities, it would easily be possible to download every book for every course each semester and have them all available concurrently.

As of 2012 the e-book formats are making a substantial dent in the market for conventional paper books. Some features that are superior to paper books include the ability to deal with graphics and animation downloads; e-book subscriptions from major technical journals; and inclusions of links to relevant web sites.

Costs = 9; Efficiency = 4; Effectiveness = 1; Currency = 7; Overall Score = 5.25
Prognosis: Declining due to recession.

Due to the economic crisis and recession starting in 2008 and 2009, in-house training has been cut back. Layoffs and cost-cutting on the part of major corporations, to say nothing of some major corporations filing for bankruptcy and going out of business, have made in-house education increasingly rare.

In-house education was number one in overall effectiveness from 1985 when our surveys started through the end of 2007 before the financial crisis and the recession. In-house education still ranks as number 1 in effectiveness, and is number 4 in efficiency. That means that a great deal of information can be transmitted to students with relative ease and speed.

However, this channel is only available for employees of fairly large corporations such as IBM, Microsoft, Google, and the like.. The author estimates that roughly half of the U.S. software personnel currently work in organizations large enough to have in-house training; i.e just over 1,600,000 U.S. software professionals have access to this channel.

The earlier studies in 2001, 2002, 2003, and 2004 indicated a decline in this channel. The economic downturn of 2008 and 2009 caused some in-house training to be cancelled, due in part to the fact that the instructors were laid off or took early retirement. Indeed many potential students also lost their jobs.

In addition, the reduction in entry-level hiring has reduced the need for education of recent college graduates who are joining large companies in junior positions. In-house education is likely to continue to be diminished through 2015. By then e-learning will probably be even more effective. As other channels such as virtual education and webinars continue to expand, the high-water mark for in-house education seems to have passed..

Some large software employers such as Accenture, AT&T, EDS, IBM, Microsoft, and many others have in-house curricula for software professionals and managers that are more diverse and current than most universities in the United States. A former chairman of ITT observed that the Fortune 500 companies in the United States have more software instructors than all universities put together. Employees within large companies have more student days of in-house training than all other channels combined.

The in-house courses within major corporations are usually very concentrated and very intensive. An “average” course would run from two to three business days, starting at about 8:30 in the morning and finishing at about 4:30 in the afternoon. However, to optimize student availability, some courses continue on into the evening.

current technologies.

Another advantage of in-house training is the ease of getting approval to take the courses. It requires far less paperwork to gain approval for a corporation's in-house training than it does to deal with a tuition refund program for university courses.

As this is written in 2012, it is uncertain if in-house education will regain the importance that it had during the 1980's and 1990's and indeed up to the recession of 2008. The economic future is too uncertain to make long-range predictions.

One interesting finding about in-house education is the impact on software development productivity. Companies that provide 10 days of training per year for software engineers have higher annual productivity rates than companies that provide 0 days of training, even though 10 working days are set aside for courses. The value of the training appears to pay off in better performance.

Although the information was not explored in depth and there may not be a provable relationship, it was interesting that companies with 10 days of training per year had lower rates of voluntary attrition than companies with 0 days of training. Apparently training has a beneficial impact on job satisfaction.

Number 5: Self Study Using CD-ROM's or DVD's

Costs = 4; Efficiency = 3; Effectiveness = 7; Currency = 10; Overall Score = 6.00
Prognosis: Improving slowly in usage; improving faster in effectiveness.

The technology of self-study courses is on the verge of being transformed by new CD-ROM and DVD approaches. It may also be transformed by e-books. The older form of self-study courses consisted of tutorial materials, exercises, and quizzes often assembled into loose-leaf notebooks. The CD-ROM or DVD varieties include all of the prior approaches, but can also feature hyper-text links and a huge variety of supplemental information.

The newest form of DVD training actually allows new content to be added to the course while it is in session. This method is so new that little empirical data is available. When it becomes widespread the "currency" score should climb rapidly.

The prospect of fully interactive learning via DVDs an exciting one, since graphics, animation, voices, and other topics can now be included. However, the costs and skill required to put together an effective DVD courses are significantly greater than those needed for traditional workbooks. The costs for the students are not particularly high, but the production costs are high.

gish access times and jerky animation.

However, by 2012 the author estimates that more than 95% of software personnel have DVD or CD ROM drives on their office computers. (Primarily ultra-light devices weighing less than 3 pounds often lack drives for DVDs and CDs. Also some organizations that are do highly classified work prohibit all forms of removable media on the premises.)

It is interesting that the tablet format does not support external drives and lack internal drives as well. However thumb drives and virtual storage on the web are becoming reasonably effective alternatives.

As of early 2012, the author estimates that more than 250,000 software personnel have taken self-study courses via CD ROM's or DVD. There are probably less than 125 such courses currently available due to the difficulty and costs of production.

The general market for DVDs including blue ray DVD's has declined in the face of streaming video and cloud access to materials. It will probably happen as the 21st century advances that cloud materials will supplant self-study courses using CD-ROM and DVD drives.

Even newer technologies will allow the equivalent to DVD's to be downloaded not only to computers, but also directly to television sets, I-Pods, smart phones, and other hand-held devices.

The impact of the new Blu-Ray format can potentially improve the interactive capabilities of DVD education, but so far this idea remains largely unexploited by educational companies due to the fairly high costs of Blu-Ray production.

Number 6: Commercial Education
Costs = 14; Efficiency = 5; Effectiveness = 4; Currency = 6; Overall Score = 7.25
Prognosis: Declining due to recession.

The economic crisis starting in 2008 and 2009 and still lingering in 2012 had a depressing impact on commercial education. Students are in decline and air-travel is becoming more and more expensive. No doubt lower cost methods such as e-learning will start to supplant normal commercial education.

For many years, commercial education ranked number two in overall effectiveness. There is a significant sub-industry of commercial education providers for the software domain. In 2012 it is now in 5th place, but still quite effective.

Companies within this sub-industry include Computer Aid (CAI), Construx, Coverity, Cutter, Data-Tech, Digital Consulting Inc. (DCI), Delphi, FasTrak, the Quality Assurance

Operating at a higher level are specialized information companies such as Gartner Group and Forrester Research. These companies provide both standard reports and also do customized research for clients. The main client bases of such high-end companies are executives and top management. In keeping with these target markets, the fees and costs of such research are high enough so that they appeal primarily to major corporations and some large government software executives, such as the CIO's of states and major cities.

There are also non-profit organizations that offer fee-based training. For example the non-profit International Function Point Users Group (IFPUG) offers training and workshops in function point topics. The Project Management Institute (PPI) also offers commercial education as does the Software Engineering Institute (SEI)..

There are also hundreds of local companies and thousands of individual consultants who teach courses on a contract basis within companies and sometimes as public courses as well. There are many courses offered by non-profit organizations such as the ACM, DPMA, IEEE, IFPUG, SEI, and scores of others.

The author estimates that about 500,000 U.S. software personnel take at least one commercial software seminar in the course of a normal business year. However from 2012 onward there was a decline due to the global recession.

Since the major commercial educators run their training as a business, they have to be pretty good to survive. A primary selling point of the larger commercial education companies is to use famous people as instructors on key topics. For example, such well-known industry figures as Bill Curtis, Chris Date, Tom DeMarco, Tim Lister, Howard Rubin, Ed Yourdon, the late Watts Humphrey, Dr. Gerry Weinberg, Dr. James Martin, and Dr. Carma McClure all offer seminars through commercial education companies.

A typical commercial seminar will run for two days, cost about $895 to enroll, and attract 50 students. A minor but popular aspect of commercial education is the selection of very good physical facilities. Many commercial software courses are taught at resort hotels in areas such as Aspen, Orlando, Phoenix, or San Francisco.

The main strengths of commercial education remain and are twofold:

1) Very current topics are the most salable.

2) Top-notch instructors are the most salable.

This means that commercial seminars are likely to cover material that is not available from a university or even from an in-house curriculum. It also means that students get a chance to interact with some of the leading thinkers of the software domain. For example

The commercial education market has been most widely utilized with companies in the top quartile of software productivity and quality levels as noted during the author's assessment and benchmark studies. In other words, companies that want top performance from their managers and technical workers realize that they need to bring in top-gun educators.

Number 7: Vendor Education

Costs = 13; Efficiency = 6; Effectiveness = 5; Currency = 5; Overall Score = 7.25
Prognosis: Declining due to recession.

Vendor education was formerly ranked number three in overall effectiveness and is now number 6. This is not because of slipping quality on the part of vendors, but because of the rapid emergence of webinars and new DVD technologies.

Vendor supplied education has been a staple of the software world for almost 50 years. Because vendor education in tools such as spreadsheets and word-processors are taken by non-software personnel, the total numbers of students is enormous. However, within the software domain the author estimates that about 500,000 software personnel will take at least one vendor course in a normal business year.

Vendor education used to be free in the days when hardware and software were still bundled together. Some vendor education is still free today, when used as part of marketing programs. Normally vendor education is sold to clients at the same time that they buy the tools or packages for which training is needed.

Almost every large commercial software application is complicated. Even basic applications such as word processors and spreadsheets now have so many features that they are not easily mastered. Thus large software companies such as Artemis, IBM, Oracle, and Computer Associates offer fee-based education as part of their service offerings.

The size, feature set, and complexity of software products mean that every major vendor now has some kind of education offering available. For really popular tools in the class of Microsoft Word, Word Perfect, Excel, Artemis Views, KnowledgePlan, etc. there may be local consultants and even high-schools and college courses that supplement vendor-supplied education.

For very large applications such as ERP packages from Oracle and SAP it is hardly possible to learn to use the software without extensive education either from the vendors themselves, or from specialized education companies that support these packages.

Vendor education is a mainstay for complicated topics such as learning how to deploy and utilize enterprise resource planning packages by vendors such as BAAN, Oracle,

whole does a creditable job of getting clients up and running on the applications in question.

Vendor education is usually a lot cheaper than commercial education, and the effective costs are sometimes less than $100 per student day. Vendor education is often offered on company's own premises, so it is generally very convenient. On the other hand, you don't expect big name instructors to constitute the faculty of vendor training either.

However newer methods such as e-learning are even cheaper, and have the advantage that courses can be taken 24 hours per day at the convenience of the students. Therefore vendor education will decline in future years and e-learning methods will become the major vehicle.

Further as the recession deepens and lengthens, instructors are usually among the first to lose their jobs as cost-cutting measures. Therefore vendor education with live instructors is entering a period of decline.

As software packages continue to evolve new features and more complexity, vendor-supplied education will remain a stable feature of the software world well into the next century, but moving from live instructors towards perhaps e-learning or even using virtual environments with avatars..

Vendor education was also been negatively affected by the economic downturn. Some vendors lost money due to reduced sales so course offerings were naturally reduced as well. Other vendors were acquired, and some went out of business. In spite of the reduction in courses and instructors, vendor-supplied education remains an important channel of instruction, even though the numbers of vendors, students, and courses are in decline.

Number 8: Live Conferences

Costs = 12; Efficiency = 8; Effectiveness = 8; Currency = 5; Overall Score = 8.25
Prognosis: Declining due to recession.

Unfortunately the financial crisis and recession of 2008 and 2009 caused a severe decline in live conferences, and this trend may continue indefinitely although 2011 and 2012 were fairly good years for conferences.

Software-related conferences rank number eight in effectiveness in teaching new skills. However, they would rank number 1 in highlighting interesting new technologies.

There are major software conferences every month of the year, and some months may have multiple events. Conferences are sponsored both by non-profit organizations, such as the Air Force STSC, IEEE, IFPUG, GUIDE, SHARE, or ASQC and also by commercial conference companies such as Computer Aid (CAI), Cutter, Digital Consulting Inc.

or the Sloan School.

In addition, there are vendor-hosted conferences for users by companies such as Apple, Microsoft, Computer Associates, Oracle, CADRE, COGNOS, SAS, SAP and the like. These are often annual events that sometimes draw several thousand participants.

The author estimates that about 250,000 U.S. software professionals attended at least one conference in calendar year 2012, and some professionals may attend multiple conferences.

Software conferences are where the cutting edge of software technologies are explained and sometimes revealed for the very first time. Many conferences also feature training seminars before or after the main event, and hence overlap commercial education channels and vendor education channels.

However, most of us go to conferences to find out what is new and exciting in our chosen domains. The mix of speakers and topics at conferences ranges from brilliant to boring. Conferences are arranged with many concurrent sessions so that it is easy to leave a boring session and find a better one.

Most conferences interleave keynote speeches to the whole audience with parallel sessions that cover more specialized topics. A typical U.S. software conference will run for three days, attract from 200 to more than 5000 attendees, and feature from 20 to more than 75 speakers.

In addition to primary speakers and seminar leaders, many conferences also have "vendor showcases" where companies in related businesses can display their goods and perhaps make sales or at least contacts. The fees vendors pay for participating in such conference defray the administrative costs, and sometimes even allow conferences to be run as profit-making opportunities.

On the whole conferences do a good job in their primary role of exposing leading-edge technologies to the audience. You seldom come away from a conference with an in-depth working knowledge of a new technology. But you often come away with a solid understanding of what technologies merit closer analysis.

Within recent years, several conferences have become so large that the proceedings are now starting to be released on CD ROM rather than traditional notebooks or bound hard copies.

In the future, it is possible that live conferences will merge with webinars and with DVD production. It is technically possible to have simultaneous live events and webinars so that the speakers are seen by a live audience and a remote audience at the same time. It

If the economy continues to decline and life conferences lose attendees, as happened before during the "dot.com" crash and earlier recessions, it may be that webinars will become effective substitutes for live instruction.

Number 9: Wiki Sites

Costs = 2; Efficiency = 9; Effectiveness = 16; Currency = 3; Overall Score = 7.50
Prognosis: Increasing rapidly in usage; increasing in effectiveness

The word "wiki" means "fast" in Hawaiian. The term has started to be applied to web sites which allow multiple people to participate towards common goals.

Wiki sites made their first appearance in this list of educational channels in 2009. This is a new and emerging technology that allows many participants to collaborate on a common theme or common undertaking. However other forms of collaborative work have joined wiki sites, such as Google circles and various commercial collaborative tools.

The most famous example of a wiki site is the "Wikipedia" which has become the largest encyclopedia in the world. Each entry or article in the Wikipedia is written by a volunteer. Readers or other volunteers can modify the entry and input their own thoughts and ideas.

At first thought, wiki's would seem to lead to chaotic and perhaps offensive final products. But in fact many wiki's, including the Wikipedia, are temperate in tone and rich in content. This is partly due to the fact that readers can immediately delete offensive comments.

Some wiki sites try to screen and monitor inputs and changes before posting them; others simply allow the material to be self-corrected by other readers. Both methods seem to be effective.

What wiki sites do best is allow people with common interests to quickly produce documents or web sites that contain their accumulated and shared wisdom. Thus wiki sites would be very appropriate for technical issues such as Agile development, software quality, software security, testing, maintenance, and other topics where there are many practitioners with a need to share new data and experiences.

Costs = 8; Efficiency = 7; Effectiveness = 13; Currency = 8; Overall Score = 9.00
Prognosis: Increasing rapidly in usage; Increasing rapidly in effectiveness

Simulators for teaching mechanical skills such as how to fly an airplane or how to assemble a machine gun have been used by commercial and military organizations for many years.

But in recent years new forms of broader simulation sites have emerged, such as "Second Life" which is a kind of virtual world where avatars (symbolic surrogates for computer users) wander through a virtual landscape and interact with other avatars and with resources such as text documents and graphics.

This new form of virtual reality has other purposes besides training and education, but it is starting to move in the direction of education. For example it would be possible to teach formal design and code inspections using avatars in a virtual room. Not only could inspections be taught via simulation, but indeed they could be performed as well.

Many other new technologies could also be taught in the same fashion. Currently the costs of producing tutorial materials are fairly high and the process is complex, but both of these issues should be eliminated fairly soon.

Virtual reality in simulated worlds is likely to become a fast-moving educational method within the next five years. It is theoretically possible to create a virtual "university" in which avatars for both students and professors will interact exclusively in on-line environments.

Although simulation and virtual worlds are not in the main stream of education as of 2012, the odds are good that they will rise to the top of the list within 10 years.

Number 11: Software Journals

Costs = 7; Efficiency = 11; Effectiveness = 14; Currency = 9; Overall Score = 10.25
Prognosis: Declining in numbers due to recession; Stable in effectiveness

Now that journals are available for the Apple I-Pad, Amazon Kindle Fire, Nook color and other e-book platforms, it can be anticipated that e-journals will begin to replace paper journals due to economic reasons. Electronic publishing is much quicker and cheaper than paper publishing, and far more friendly to the environment. Further electronic journals are distributed within minutes, so there is no delay due to surface mails.

As of 2012 dozens of journals are now offering electronic versions, and quite a few journals have abandoned paper publication entirely. With modern platforms such as I-Pads and Kindle Fires these electronic versions are as easy to read as paper versions.

gies rather than the depths of technologies.

There are scores of software-related journals. Some are commercial journals published for profit and depending upon advertising revenues. Many others are produced by non-profit professional associations. For example this journal, IEEE Computer, is produced by a non-profit association as are the other IEEE journals.

A literate and very broad-based software journal is published by the U.S. Air Force' Software Technology Support Center. This journal, Crosstalk, has become one of the few software journals to strive for depth in articles, rather than shallow coverage consisting primarily of short quotes from industry figures.

Another interesting journal is the Cutter Journal, originally founded by software guru Ed Yourdon under the name of "American Programmer." As with the Crosstalk journal, the Cutter journal publishes full-length technical articles.

There are so many journals that a number of them are quite specialized and occupy fairly narrow niches. Examples of some of these specialized niche journals include Metric Views (the journal of the International Function Point Users Group) and Cross Talk (the journal of the Air Force Software Technology Support Center).

Many software journals are available to software professionals for free, providing the potential subscribers bother to fill out a questionnaire (and have some responsibility for acquiring tools or software). On the other hand, some journals have annual subscriptions that can exceed $1000.

The software professional world ends up with subscriptions to quite a few journals, even though few may actually be read. SPR estimates that software technical personnel subscribe to or have access to about four journals per month on average.

Comparatively few software journals contain articles of lasting technical value. When journals do discuss technologies, it is seldom an in-depth treatment. This is understandable, given that neither journalists nor professional contributors can spare more than a minimum amount of time to assemble information before writing articles.

The best articles in terms of technology transfer are those on specific topics, often in special issues. For example, several journals have had special issues on topics such as quality assurance, measurement and metrics, software maintenance, object-oriented approaches, change management, and the like.

The least effective articles are the typical broad-brush surveys produced by journalists that consist largely of short quotes from 20 to 50 different people. This approach can seldom do more than bring a topic into public view.

Metrics and Productivity Institute" or ITMPI journal. The web site for this journal is www.ITMPI.org. This journal is published by Computer Aid and includes interviews with famous software persona such as Watts Humphrey and Ed Yourdon, technical articles, and citations to hundreds of relevant web sites.

As on-line communication becomes the pervasive medium for information exchange, both software journals and other print-based information media are moving to create parallel on-line versions.

Number 12: Self-study Using Books and Training Material

Costs = 5; Efficiency = 13; Effectiveness = 12; Currency = 11; Overall Score = 10.25
Prognosis: Increasing in numbers due to recession; Increasing in effectiveness

The self-study market using books ranks number eleven in overall effectiveness. The market for traditional self-study courses is not fast growing, but has been relatively solid and stable for 50 years. The author estimates that a total of about 50,000 software professionals will take some kind of self-study course over a normal business year.

The usual format of self-study material is a loose-leaf notebook. This form of self-study material can be effective for those who are self-motivated, but tends to be bland and frequently boring. Some self-study courses also include video or audio cassettes.

The most common topics for self-study are those that have relatively large market potentials. This means that basic subjects such as "Introduction to COBOL Programming" are most likely to be found in self-study form.

Really new technologies are seldom found as self-study courses, because of the time required to produce the materials and the uncertainty of the market. There are always exceptions to rules, and fairly new topics such as cloud computing or Android programming have already arrived in self-study form due to the obviously large markets.

In theory electronic books on hand-held devices would seem to be a useful media for studies of any kind. However, among the author's clients there were no formal course materials produced for such devices. We did not encounter enough students using these devices to form an opinion. Overall, usage of hand-held reading devices is still at a very early stage of deployment, although they show great potential for the future.

It is interesting that a number of school districts including all of California are thinking of moving from paper books to e-book formats. This topic is outside the scope of software, but shows that e-books are highly cost competitive vis a vis paper books and conventional text books.

Number 13: On-the-Job-Training and Coaching

Costs = 11; Efficiency = 10; Effectiveness = 10; Currency = 12; Overall Score = 10.75
Prognosis: Declining in numbers due to recession

On-the-job training has been most often utilized for either special techniques developed and used within companies, or for basic skills that are seldom taught at universities, such as formal design and code inspections.

With on-the-job training new hires are instructed in selected methods by experienced users of the method. The most effective kinds of on-the-job training are topics where performing a task is the best way to learn the tasks. For example formal inspections, Quality Function Deployment (QFD), learning to use proprietary specification methods, learning to use programming languages that are unique to specific companies are good choices for in-house training.

The down side of on-the-job training is that the experts who are doing the training need to take time away from their own work. This is why the costs are fairly high. Also, it sometimes happens that what the older employees teach the new employees may be somewhat out of date.

In recent years a variant of on-the-job training has become quite popular. These are positions called "coaches" who instruct teams in relatively complex topics where instruction is needed for success.

Probably the most widespread form of coaching in 2012 are "Agile coaches" where there are probably more than 5,000 in the United States alone. Because the concepts of Agile are fluid and open to many variations, they often lead to confusion and inefficiency. Agile coaches attempt to solve these problems by imparting best practices from the best Agile teams.

Number 14: Mentoring

Costs = 10; Efficiency = 12; Effectiveness = 11; Currency = 15; Overall Score = 11.75
Prognosis: Declining in numbers due to recession.

The concept of "mentoring" involves a one-to-one relationship between a new employee and an older and more experienced employee. Mentoring is most often used for proprietary methods, such as teaching a new employee about local corporate standards, or teaching a new employee about custom tools that are only available in one location.

Mentoring is often effective at a social level, but it is somewhat expensive if it requires that the mentor take too much time away from his or her normal work.

may have been customized.

Unfortunately there are no statistics on the number of mentors or people being mentored, so this channel of learning is ambiguous as to usage and overall effectiveness.

Number 15: Professional Books, Monographs, and Technical Reports

Costs = 6; Efficiency = 14; Effectiveness = 16; Currency = 13; Overall Score = 12.00

Software books tend to rank 14th in overall effectiveness in transferring software skills in the United States. The problem is not with the books themselves, but because the U.S. software industry does not appear to be especially literate, which is something of a surprise given the nature of the work.

Many of the books themselves are excellent. For example Dr. Barry Boehm's classic Software Engineering Economics (Prentice Hall), Watts Humphrey's book on Team Software Processes (TSP) (Addison Wesley), and Dr. Roger Pressman's classic Software Engineering – A Practioners Approach (McGraw Hill) have both seen numerous editions and sold close to 1,000,000 copies each.

Software books are a staple for reference purposes even if not for pedagogical purposes. A typical software engineer will have a library of between 20 and 100 volumes on topics of importance such as the programming languages, operating systems, applications, and hardware used in daily work.

As of 2012 there are more than 2,600,000 software personnel in the United States as noted in table 1, and the numbers are growing fairly rapidly. Yet software book publishers regard sales of more than 10,000 copies as a fairly good volume. Sales of 25,000 copies are remarkably good for software titles and only a few software titles have approached 1,000,000 copies.

The high-volume software books often aim at the end-user market and not the professional or education markets. For example books on Ruby with Rails, Visual Basic or primers on Windows 8 can exceed 1,000,000 copies in sales because there are about 10,000,000 users of these products who are not professional software personnel.

It is possible to learn a variety of software-related skills from books, but this approach is not as widely utilized as seminars or some kind of formal training. One possible exception is that of learning personal topics, such as Watts Humphrey's "personal software process" or PSP method. As of 2012 Watts' books are the primary channel for this topic.

Another situation where books are actually used for self-learning of new skills is the area of new programming languages. New programming languages have been coming out at a rate of more than one per month for the past 30 years, and more than 2,500 programming

There are many excellent software books on the market by publishers such as Addison Wesley Longman, Auerbach, Dorset House, IEEE Computer Society Press, McGraw Hill, Prentice Hall, Que, Microsoft Press, and the like.

Also included under the heading of books would be monographs published by various companies and associations. For example, software-related monographs of 50 to more than 100 pages in size are published by Accenture, IBM, IFPUG, varous IEEE associations, McKinsey, Gartner Group, Namcook Analytics, the Software Engineering Institute (SEI), and Software Productivity Research (SPR).

These privately published monographs are distributed primarily to the clients of the publishing organization. The costs range from zero up to more than $25,000 based on whether the monograph is intended as a marketing tool, or contains proprietary or special data of interest to the clients.

As of 2012 publication the web is far more common than paper publication, and of course much easier to access and much easier to update.

There are many software bookstores and large software sections within general bookstores such as Borders. And of course software books are featured in all of the major Web-based book stores such as Amazon and Barnes and Noble. The total volume of good books on software topics probably exceeds 2,500 titles for technical books and 250 titles for software management books.

Yet in spite of the plentiful availability of titles, many software managers and quite a few technical software personnel don't read more than one or two technical books a year, based on responses to assessment interviews.

The author estimates that software professionals purchase about four books per year on average (more than seem to be read). In any case, it would be hard to keep current just from books since software technology changes are so volatile.

There is a curious omission in the book domain for software and project management topics. Among the more mature professions such as medicine and law, a significant number of books are available in audio form such as cassettes or CD's so they can be listened to at home or in automobiles. We have not yet encountered any audio titles in the software engineering or project management fields although some may be available.

An increasing number of technical books are becoming available on-line and can be viewed on personal computer screens or downloaded. This method of gaining access to books is expanding, but has not yet reached a critical mass. The author estimates that less than perhaps 100 software-related titles are available in on-line form, and that perhaps

New electronic devices such as the Amazon and Sony hand-held book readers, the I-Phone, and various PDA devices can be used to store books, although this is not yet a major market for book publishers.

Of course personal ownership of books may not be necessary of a company or government agency maintains a good technical library. One interesting observation from author's assessments over the years is that companies with large technical libraries have higher annual productivity rates than companies of the same size without such libraries.

Having a library is probably not a causative factor for high productivity though. The most likely reason for the correlation is that companies with good technical libraries also tend to have better than average salary and compensation levels, so they select top performers for software occupations.

Smaller than books are vast numbers of monographs and technical reports, which usually range in size between 20 and 75 pages. Usually such documents are devoted to a single topic, such as service-oriented architecture (SOA) or perhaps marketing issues.

Some companies such as Gartner Group in the U.S. and Research and Markets do a surprisingly large business from marketing both paper and on-line versions of monographs and technical reports.

The primary market for these shorter documents are either corporate libraries or executives interested in business trends and high enough in company hierarchies to be authorized to spend money on subscriptions or individual studies, many of which are far more expensive than ordinary books.

As tools for informing executives about emerging business trends, the technical report business is fairly effective. As tools for learning the specifics of technical topics such as testing or inspections, these shorter reports are not as effective as books.

One technical problem with books published on paper is the fact that the software industry changes faster than the books can be revised and new editions brought out.

A possible new business model for book publishers as they migrate to e-books would be to sell subscriptions to updates at the same time as the book is originally downloaded. For example a downloaded software textbook in e-book format might sell for $25, and subscriptions to updates might sell for $10 per year. Not only would e-books be cheaper than paper books, but by offering subscriptions they would lead to recurring revenue streams.

Costs = 15; Efficiency = 15; Effectiveness = 3; Currency = 16; Overall Score = 12.25
Prognosis: Declining in numbers due to recession; stable in effectiveness

From the author's studies, undergraduate university education was only number fifteen in overall effectiveness for software professionals. Universities are often not very current in the topics that they teach. In general, university curricula lag the actual state of the art of software by between five years and 10 years. This lag is because of the way universities develop their teaching materials and their curricula.

There are a number of critical topics where software education at the university level lags far behind what is truly needed to achieve professional status for software. The most glaring omissions include:

1. Software security practices for safeguarding high-risk applications
2. Software quality control practices for minimizing delivered defects
3. Software measures and metrics for effective economic analysis
4. Software architecture for optimizing use of reusable components
5. Software methods for effective renovation of legacy applications
6. Software technology evaluation and technology transfer

These gaps and omissions need to be quickly filled if software is to evolve from an art form into a true engineering discipline.

There are some exceptions to this rule that universities always lag. Many universities have established fairly close ties with the local business community, and attempt to offer courses that match the needs of the area's software employers. For example, Stevens Institute of Technology in New Jersey has established close ties with AT&T and is offering a master's program that includes topics suggested by AT&T. Bentley College in the Boston area, Washington University in St. Louis, Georgia State University in Atlanta, St. Thomas University in the St. Paul area and many other schools adjacent to large software producers have adopted similar policies of curricula based on the needs of major software employers.

A strength of undergraduate education is that what gets taught tends to be used throughout the rest of the professional lives of the students. University education ranks #3 in effectiveness or the volume of information transmitted.

The author estimates that perhaps 95,000 U.S. software professionals will take university or college courses in the course of a normal business year.

Having performed a consulting study on continuing software education, a few practical issues were noted. The way companies fund tuition refund programs is often remarkably cumbersome. Sometimes several layers of management approval is required. The cours-

Also, the tuition refund policies are based on achieving passing grades. This is not an unreasonable policy, but it does raise the mathematical probability that the student will end up with out of pocket expenses which can be significant.

On the whole, university training appears to be more expensive and less effective than in-house training, commercial education, or vendor education for practicing software professionals.

A former chairman of the ITT Corporation once noted in a speech that it took an average of about three years of in-house training and on-the-job experience before a newly graduated software engineer could be entrusted with serious project responsibilities. This was about two years longer than the training period needed by electrical or mechanical engineers. The conclusion was that software engineering and computer science curricula lagged traditional engineering curricula in teaching subjects of practical value.

Indeed, a quick review of several university software engineering and computer science curricula found some serious gaps in academic training. Among the topics that seemingly went untaught were: software cost estimating, design and code inspections, statistical quality control, maintenance of legacy application, metrics and measurements, six-sigma methods, risk and value analysis, function points, and joint application design (JAD) for requirements analysis. While basic technical topics are fairly good at the university level, topics associated with project management are far from state of the art levels.

Universities are also being impacted by the recession, and the future of university training in terms of numbers of software engineering and computer science students and graduates is uncertain.

Number 17: Graduate University Education

Costs = 16; Efficiency = 16; Effectiveness = 2; Currency = 15; Overall Score = 12.25
Prognosis: Declining in numbers due to recession; stable in effectiveness

Graduate education in software engineering or computer science unfortunately tends to bring up the rear and is ranked number 17. Graduate school does rank #2 in effectiveness, and it does transmit a great deal of information to graduate students.

The down side is that a lot of the information that is transmitted may be obsolete, since university curricula often lag the business and technical worlds by five to 10 years.

Graduate education could be improved by greater concentration on special topics such as the economics of software. Software costs are so heavily dominated by defect removal expenses, catastrophic failures, and huge cost and schedule overruns that there is a need

Also, software security problems are not only rampant in 2012 but are becoming more numerous and more serious at a rapid pace. It is obvious that universities have lagged severely in this area, and also in the area of software quality control. For that matter, other key topics such as construction from reusable materials also lag at both the undergraduate and graduate levels.

Software Areas where Additional Education is needed

From working as an expert witness in a number of software lawsuits for breach of contract or for litigation involving claims of poor quality, it is apparent that some important topics need additional education or reeducation. Table 7 shows 25 major technology areas where performance in the software world seems to be deficient, in approximate order of importance to the software profession:

Table 7 Gaps in Software Training Circa 2012

1	Security vulnerability prevention
2	Security recovery after attacks
3	Quality control (defect prevention)
4	Quality control (defect removal)
5	Quality estimating
6	Quality measurement
7	Measurement and metrics of software
8	Change management
9	Tracking of software projects
11	Cost and schedule estimating
10	Intellectual property protection
12	Reuse of software artifacts
13	Risk analysis and abatement
14	Value analysis of software applications
15	Technology analysis and technology transfer
16	Renovation of legacy applications
17	Requirements collection and analysis
18	Software project management
19	Formal inspections
22	Test case design
20	Performance analysis
21	Customer support of software applications
23	Contract and outsource agreements

Failures and problems associated with these topics appear to be endemic in the software world, and especially so for large software applications. From the frequency with which large software projects fail, and the even larger frequency that have cost and schedule overruns, it can be concluded that training in software topics is urgently in need of major improvement.

New Directions in Software Learning

The global recession is causing thousands of organizations to reduce costs in order to stay in business. Almost all labor-intensive activities such as training are going to be scrutinized very carefully.

At the same time the technologies of virtual reality, e-books, and distributing information over hand-held devices are increasing in sophistication and numbers of users.

Within a period of perhaps 10 years, the combination of recessionary pressures and technology changes will probably make major differences in software learning methods. On-line web-based information, e-books, and hand-held devices will no doubt replace substantial volumes of paper-based materials.

In addition virtual reality may introduce artificial classrooms and simulated universities where students and teachers interact through avatars rather than face to face in real buildings.

The increasing sophistication of intelligent agents and expert systems will probably improve the ability to scan vast quantities of on-line information. The fact that companies such as Google and Microsoft are rapidly converting paper books and documents into on-line text will also change the access to information.

However software has a long way to go before it achieves the ease of use and sophistication of the legal and medical professions in terms of the organization and access to vital information. For example, there is no software equivalent to the Lexis-Nexis legal reference company as of 2012.

Over the next few years changes in learning methods may undergo changes as profound as those introduced by the printing press, by and television. However, the quality of software information is still poor compared to the quality of information in more mature fields such as medicine and law. The severe shortage of quantitative data on productivity, schedules, quality, and costs makes software appear to be more of a craft than a true profession.

As of 2012 the technologies exist to create a virtual-reality software university that would resemble a real university, only with more sophisticated access to learning materials. The essential idea is to use concepts from virtual reality sites such as Second Life but apply them to practical software education topics.

In order to do this the process would start with licensing a virtual reality rendering engine from one of the sophisticated computer game companies. But instead of using the engine to create virtual battlefields or forests, the engine would create what appears to be a university campus complete with buildings and students. To be convincing a virtual campus would probably need to be aesthetically pleasing and feature landscaping as well as campus buildings.

Potential students would be able to move their avatars through the campus and enter the buildings. For example there would be buildings labeled " Project Planning and Estimating Department," Project Governance Department," Project Requirements Department," and so forth.

Upon entering one of these virtual buildings there would be a series of virtual class rooms and virtual offices for the instructors and professors. This model assumes that live experts will be participate in the virtual university so the offices would have the names of actual experts such as Dr. Barry Boehm, Dr. Victor Basili, and others who entered into agreements to offer courses through the Virtual Software University.

Of course the instructional staff would not be present at all times, so office hours would be posted on the virtual offices. In addition students would be able to leave messages and requests for the various professors and instructors.

The class rooms would appear to be actual class rooms similar to those at MIT, Harvard, Princeton, and other major universities. Several kinds of courses would be offered. One form of course would be presented in real time by the avatars of live instructors. (It is assumed that the Avatars for the Virtual Software University would be images of the actual instructors.) These live courses would be announced and could be scheduled. Some of these would be free but others might be fee-based.

There would also be recorded course materials that students could download and use at home or at their convenience.

The virtual class rooms would be more sophisticated than most real class rooms, in that all of them would be able to have multiple screens, feature animation and dynamic materials, and possibly even use 3-D instructional materials.

Interaction between virtual students and virtual professors would be similar to real life, in

of course working compliers and interpreters for teaching various programming languages.

Of course every university needs a good library, and the library for the Virtual Software University would be world-class. It would have features not offered in normal libraries. For example suppose a student is interested in the topic of "software testing." Not only would the library have abstracts of every published book and article on testing, but it would constantly be refreshed by means of intelligent agents that would scan the web for new materials.

Of course for many topics the number of books and reference items might be in the millions, so the library would also include tools for narrowing searches and for assigning relevance scores.

Since the Virtual Software University might be accessed by students from several hundred countries, there would also be real-time translation services between all major natural languages. Thus courses might be simultaneously available in English, Russian, French, Italian, German, Portuguese, Arabic, Spanish, Japanese, and essentially every human language.

Ideally the translation services would encompass both text materials and perhaps even spoken discussions among students and faculty. A sophisticated university such as the Virtual Software University would no doubt license language translation tools plus perhaps voice to text tools such as Dragon or one of the others.

It is obvious that the virtual software university would want to offer world-class facilities for those who might have physical limits. For example to aid the deaf and hard of hearing all spoken material would be simultaneously translated into printed text. All video and instructional films would automatically include close captions or subtitles. This technology is available in 2012. It would also be possible to offer simultaneous translation of spoken courses into sign language. However translation of printed materials into sign language may not be fully available circa 2012.

For the blind all printed materials would be translated into speech. This technology also exists in 2012. It might even be possible to support simultaneous translation into Braille, although that is perhaps outside the state of the art as of 2012.

For those in wheel chairs who prefer that their avatars also have wheel chairs, the classrooms and buildings of the virtual software university would all be accessible to wheel chairs, and also clearly identified verbally for the blind.

As with real universities, students would be able to interact with one another and would also be able to participate in special interest groups or Wiki sites on topics such as static

ware University would have licenses from all major benchmark groups and would have working versions of a variety of planning and estimating tools, test tools, and many others.

Unlike real universities the Virtual Software University campus would be operational 24 hours a day 365 days per year. Of course live instructors would take normal holidays and vacations, but the library and the recorded course materials would always be available.

Assuming global students and global faculty, it makes good sense for the Virtual Software University to operate around the clock. After all it is always daylight somewhere.

Because topics of interest change fairly often, the Virtual Software University would include a "Student Center" where students from many countries and many fields could interact with one another in order to exchange information and find out what techniques are being used successfully and which ones are difficult to master.

As with real universities there would be many special interest groups or people who are all interested in the same topics. One service that the Virtual Software University could provide would be access to local and national information from many countries such as the U.S., China, Brazil, Japan, India, and many others. For example each country might have its own bulletin board that could be used to announce courses and webinars that are located in the various cities of the home countries of the students.

Another service that the Virtual Software University might provide is a daily summary of webinars on selected topics such as testing, requirements engineering, and new tools and methods. Currently there are so many webinars offered that it is not easy even to keep track of them.

In the student center there would be a virtual bulletin board. Here vendors of tools or services might place ads, and students with interests in special topics might start looking for "birds of a feather" groups.

The Software Virtual University might also use Linkedin or Plaxo or another network service to send messages to students with special interests or with common interests who might want to communicate with each other.

Since students would not be on campus more than perhaps an hour or two per day, the Virtual Software University would also include links to various e-book sources such as Amazon, Barnes and Noble, Google, etc. Indeed course curricula and selected texts would be capable of being downloaded and ordered as e-book packages for various courses such as testing, estimating, project management, and the like.

The fundamental idea for the Virtual Software University is to consolidate the huge but

Two other aspects of the Virtual Software University would be different from regular campuses. First, each of the major professional associations such as the American Society of Quality (ASQ) or the Project Management Institute (PMI) could have their own virtual buildings and offer both training and membership services.

The same concept would be available for major corporations such as IBM, Google, and Microsoft. They could design and commission corporate buildings on the virtual campus where training in their products could take place. In fact some of the funding for the Virtual Software University would no doubt be the fees paid by corporations for these structures and for participation in the Virtual Software University.

Another unique aspect of the Virtual Software University would be links to major conferences such as the Japanese Symposium on Software Testing (JaSST) or the IBM Innovate Conferences. The Virtual Software University would have several large conference halls where those who could not attend actual events in person would be able to participate in the major sessions and tutorials. Attendance policies for these virtual conferences would be set by the conference committees, and would probably offer reductions on the fees for attending in person.

The early versions of the Virtual Software University would probably offer short courses or webinars that lasted only an hour or less. However it is technically possible to envision the Virtual Software University linking to real universities and offering standard curricula in virtual environments.

If the idea catches on then eventually real universities such as Harvard and MIT or the University of Nalanda in India might participate and offer virtual courses either on their home campuses or through the facilities of the Virtual Software University.

At some point the facilities of the Virtual Software University would be sufficient to administer examinations and offer professional certification in topics such as requirements engineering, function point analysis, testing, project managements, and perhaps dozens of other technical disciplines where certification is available.

It is not impossible for the Virtual Software University to eventually award actual degrees up to the PhD level. However that could only occur if the curricula and faculty were accredited. Actual degrees from the Virtual Software University might not be feasible until 2030 or thereabouts due to the novelty of the concepts and the logistics of accreditation. The initial versions of the Virtual Software University would be aimed at professional training rather than undergraduate or academic training.

Security would have to be included as part of the design of the university virtual campus. This is to keep hackers and viruses from damaging the course materials or disrupting the

curs, it is interesting that the essential technologies to build the Virtual Software University all exist in 2012.

Not only do the technologies exist in 2012 but the costs for constructing a virtual campus would probably be only in the range of $150,000 which is much less expensive than building real class rooms. Assuming that companies such as IBM, Microsoft, and Google who already have course materials and instructors wanted to do this, a Virtual Software University could probably be up and running within 90 days of starting out.

It is not impossible that the Virtual Software University could do for education what Face Book and Twitter have done for social networks; i.e. make learning so easy and enjoyable that attendance would reach into the millions.

Needless to say the concepts of the Virtual Software University could also be used for other forms of education such as medicine and law. It is even possible to apply the same ideas to primary and secondary education. Even in 2012 it would be much cheaper to build a virtual school for the deaf than it is to build such schools in real life.

In any technical field it is hard to keep up with the latest developments. Software technologies are changing very rapidly as the 21st century advances, and this makes it difficult to stay current.

Seventeen different channels are available to the software community for acquiring information on new software technologies. The most effective historical channel is in-house training within a major corporation, but that channel is only available to the corporation's employees. Webinars and web-based research are rapidly advancing in sophistication, and are already very inexpensive.

The use of virtual reality or simulation web sites is another exciting prospect. It is technically possible for avatars of virtual students to participate in simulated virtual classrooms.

As the Internet and on-line services grow in usage, entirely new methods of education may be created as a byproduct of large-scale international communication channels. It is also possible that in the future some form of education may become available via satellite radio, although none are currently broadcast on that channel.

Unfortunately, given the large numbers of project failures in the software domain, all 17 channels put together are probably not yet enough to raise the level of software engineering and management competence to fully professional status.

Shown below is the author's proposed full curricula of managerial and technical topics related to the software industry. The courses range from top-level executive seminars through detailed technical courses. The set of curricula is aimed at corporations with large software staffs. The instructors are assumed to be top experts in the field.

Software Curricula for Executives, Management, and Technical Personnel

Capers Jones

	Executive Courses	Days	Value
1	Software's Role in Business	0.50	10.00
2	Software's Role in Government	0.50	10.00
3	Software Security Issues in 2012	1.00	10.00
4	Software Risk Avoidance	1.00	10.00
5	Software Portfolio Values	1.00	10.00
6	Software Development Economics	1.00	10.00
7	Software Maintenance Economics	1.00	10.00
8	Sarbanes-Oxley Compliance	1.00	10.00
9	Software Governance	1.00	10.00
10	Software revenue and FASB rules	1.00	10.00
11	The Impact of Cloud Computing	1.00	10.00
12	The Impact of Big Data	1.00	10.00
13	Software Business Intelligence	0.50	9.75
14	Case Studies in Software Best Practices	0.50	9.75
15	Case Studies in Software Failures	0.50	9.75
16	Case Studies in Process Improvement	0.50	9.75
17	Software and Knowledge Management	1.00	9.40
18	Software Clouds, SOA, and SaaS	1.00	9.25
19	Pros and Cons of Offshore Outsourcing	1.00	9.10
20	Global finance and economics	1.00	9.00
21	Economics of Outsourcing	1.00	9.00
22	Software Intellectual Property Protection	1.00	9.00
23	Intellectual Property and Outsourcing	1.00	9.00
24	Software Value: tangible and intangible	0.50	9.00
25	Corporate Software Usage Analysis	1.00	9.00
26	Software Litigation Avoidance	0.50	8.75
27	Overview of Viruses and Spyware	0.50	8.50
28	Software Architecture Trends	1.00	8.50
29	Software Research: SEI, ITMPI, CISQ, etc.	0.50	8.00

33	Overview of CISQ quality standards	1.00	6.00
	TOTAL	26.50	9.08

	Project Management Courses	**Days**	**Value**
1	Software Milestone Tracking	1.00	10.00
2	Early Sizing Before Requirements	1.00	10.00
3	Software Project Planning	2.00	10.00
4	Optimizing Multi-Country Teams	1.00	10.00
5	Best Practices in Project Management	1.00	10.00
6	Software Risk Management	1.00	10.00
7	Software Cost Estimating: Automated	2.00	10.00
8	Software Security Planning	1.00	10.00
9	Benchmark sources: ISBSG, Namcook.	1.00	10.00
10	Measurement and Metrics of Software	2.00	10.00
11	Software Quality and Defect Estimating	1.00	10.00
12	Software Defect Tracking	1.00	9.75
13	Software Benchmark Overview	1.00	9.75
14	Function Point Analysis: High Speed	1.00	9.75
15	Human Resource Policies	1.00	9.60
16	Software Change Control	1.00	9.50
17	Principles of Software Reuse	1.00	9.40
18	Appraisals and Employee Relations	1.00	9.00
19	Software Cost Tracking	1.00	9.00
20	Software Maintenance & Enhancement	1.00	9.00
21	Methodologies: Agile, RUP, TSP, others	1.00	9.00
22	The Capability Maturity Model (CMMI)	2.00	9.00
23	Tools: Function Points	1.00	9.00
24	Testing for Project Managers	2.00	8.75
25	Static Analysis for Project Managers	0.50	8.75
26	Inspections for Project Managers	0.50	8.75
27	Project Management Body of Knowledge	2.00	8.70
28	Software Metrics for Project Managers	1.00	8.50
29	Software Cost Estimating: Manual	1.00	8.00
30	Tools: Cost Accounting	1.00	8.00
31	Tools: Project management	1.00	8.00
32	Tools: Human Resources	1.00	8.00
33	Tools: Cost and Quality Estimation	1.00	8.00
34	Function Points for Project Managers	0.50	8.00
35	ISO Standards for Functional Measures	1.00	8.00

40	Principles of Balanced Scorecards	1.00	6.50
41	Six-Sigma for Project Managers	2.00	6.00
42	Six-Sigma: Green belt	3.00	6.00
43	Six-Sigma: Black belt	3.00	6.00
	TOTAL	36.50	8.74

	Software Development Courses	**Days**	**Value**
1	Software Design Principles	1.00	10.00
2	Software Development Principles	2.00	10.00
3	Software Requirements Analysis	1.00	10.00
4	Security Issues in 2012	1.00	10.00
5	Hacking and virus Protection	2.00	10.00
6	Defect Reporting and Tracking	1.00	10.00
7	Software Change Control	1.00	9.75
8	Error-Prone Module Avoidance	1.00	9.50
9	Test Case Design and Construction	2.00	9.50
10	Requirements Modeling	2.00	9.50
11	Mashups and High-Speed Builds	1.00	9.50
12	Tools: Requirements	1.00	9.50
13	Tools: Design	1.00	9.50
14	Tools: Development	1.00	9.50
15	Tools: Languages/Compilers	1.00	9.50
16	Tools: Debugging	1.00	9.50
17	Tools: Test Case Design	1.00	9.50
18	Tools: Static Analysis	1.00	9.50
19	Enterprise Architecture Principles	1.00	9.00
20	Software Architecture Principles	1.00	9.00
21	Programming Language Evaluation	2.00	9.00
22	Static analysis of code	1.00	9.00
23	Formal Design Inspections	2.00	9.00
24	Formal Code Inspections	2.00	9.00
25	Web Application Design and Development	2.00	9.00
26	Principles of Software Reuse	1.00	9.00
27	Design of reusable code	2.00	9.00
28	Development of reusable code	2.00	9.00
29	Certification of reusable code	1.00	9.00
30	Development of reusable tests	1.00	9.00
31	Development of reusable designs	1.00	9.00
32	Joint Application Design (JAD)	1.00	8.75

37	Platforms and Operating Systems	1.00	8.50
38	Quality Function Deployment (QFD)	1.00	8.40
39	Development with Rommana	1.00	8.40
40	Iterative and Spiral Development	1.00	8.00
41	Agile Development Methods	2.00	8.00
42	Using SCRUM	1.00	8.00
43	Object-Oriented Design	2.00	8.00
44	Object-Oriented Development	2.00	8.00
45	Extreme Programming (XP) Methods	2.00	8.00
46	Principles of Data Base Development	2.00	8.00
47	Certification: Architecture	2.00	8.00
48	Certification: Quality Assurance	2.00	8.00
49	Certification: Testing	2.00	8.00
50	Certification: Project Management	2.00	8.00
51	Certification: Functional Metrics	1.00	8.00
52	Function Points for Developers	1.00	7.00
	TOTAL	58.00	8.89

	Software Maintenance Courses	**Days**	**Value**
1	Principles of Legacy Renovation	1.00	10.00
2	Identifying and Removing Security Flaws	2.00	10.00
3	Reducing Bad Fix Injections	1.00	10.00
4	Defect Reporting and Analysis	0.50	10.00
5	Change Control	1.00	10.00
6	Regression Testing	2.00	10.00
7	Test Library Control	2.00	10.00
8	Test Case Conflicts and Errors	2.00	9.75
9	Error-Prone Module Removal	2.00	9.50
10	Complexity Analysis and Reduction	1.00	9.50
11	Configuration Control	1.00	9.50
12	Software Maintenance Workflows	1.00	9.50
13	Maintenance of COTS Packages	1.00	9.25
14	Maintenance of ERP Applications	1.00	9.25
15	Data mining for business rules	1.00	9.20
16	Data mining for lost algorithms	1.00	9.20
17	Mass Updates to Multiple Applications	1.00	9.00
18	Static analysis of code	1.00	9.00
19	Customer Support and ITIL	1.00	9.00
20	Dead Code Isolation	1.00	9.00

24	Tools: Function Point Analysis	0.50	9.00
25	Function Points for Maintenance	1.00	8.00
26	Reverse Engineering	1.00	8.00
27	Reengineering	1.00	8.00
28	Refactoring	0.50	8.00
29	Maintenance of Reusable Code	1.00	8.00
30	Object-Oriented Maintenance	1.00	8.00
31	Maintenance of Agile and Extreme Code	1.00	8.00
32	Maintenance of Open-Source Applications	1.00	8.00
	TOTAL	29.00	

	Software Quality Assurance Courses	**Days**	**Value**
1	Software Defect Estimating	1.00	10.00
2	Software Defect Removal Efficiency (DRE)	1.00	10.00
3	Software Defect Tracking	1.00	10.00
4	Software Security and Quality in 2012	2.00	10.00
5	Quality Benchmarks: ISBSG, SPR, etc.	1.00	10.00
6	Error Prone Module Analysis	2.00	9.95
7	Software Test Case Design	2.00	9.75
8	Software Test Library Management	1.00	9.75
9	Reducing Bad-Fix Injections	1.00	9.75
10	Test Case Conflicts and Errors	1.00	9.75
11	Software Design Inspections	2.00	9.50
12	Software Code Inspections	2.00	9.50
13	Software Test Inspections	2.00	9.50
14	Static analysis of code	2.00	9.50
15	Defect removal using TSP/PSP	2.00	9.00
16	Automated software testing	2.00	9.00
17	Quality Assurance of Software Reuse	1.00	9.00
18	Quality Assurance of COTS and ERP	1.00	9.00
19	Quality Assurance of Open Source	1.00	9.00
20	Tools: Quality Assurance	1.00	9.00
21	Tools: Defect Prediction	1.00	9.00
22	Function Points for Quality Measurement	1.00	8.00
23	Overview of the CMMI	1.00	8.00
24	ISO and IEEE Quality Standards	1.00	7.00
25	Six Sigma: Green Belt	3.00	7.00
26	Six Sigma: Black Belt	3.00	7.00
	TOTAL	39.00	

	Software Testing Courses	**Days**	**Value**

4	Test Schedule Estimating	1.00	10.00
5	Software Defect Estimating	1.00	10.00
6	Defect Removal Efficiency Measurement	1.00	10.00
7	Software Build Planning and Control	1.00	10.00
8	Test Coverage Analysis	1.00	9.50
9	Identifying Error-Prone Modules	2.00	9.50
10	Data Base Test Design	1.00	9.50
11	Removal of Incorrect Test Cases	1.00	9.50
12	Test Case Conflicts and Errors	1.00	9.25
13	Static analysis and testing	1.00	9.00
14	Reducing Bad-Fix Injections	1.00	9.00
15	Basic black box testing	1.00	9.00
16	Basic white box testing	1.00	9.00
17	Basic gray box testing	1.00	9.00
18	Fundamentals or Risk-Based Testing	1.00	9.00
19	Fundamentals of Unit Testing	1.00	9.00
20	Fundamentals of Regression Testing	1.00	9.00
21	Fundamentals of Component Testing	1.00	9.00
22	Fundamentals of Stress Testing	1.00	9.00
23	Fundamentals of Virus Testing	2.00	9.00
24	Fundamentals of Lab Testing	1.00	9.00
25	Fundamentals of System Testing	2.00	9.00
26	Fundamentals of External Beta Testing	1.00	9.00
27	Testing Web Applications	1.00	9.00
28	Tools: Automated Testing	2.00	9.00
29	Tools: Test Case Design	1.00	9.00
30	Tools: Test Library Control	1.00	9.00
31	Tools: Defect Tracking	1.00	9.00
32	Tools: Complexity Analysis	0.50	9.00
33	Tools: Test Coverage Analysis	0.50	9.00
34	Fundamentals of reusable test materials	1.00	9.00
35	Testing Cloud, SOA, and SaaS	2.00	8.80
36	Testing COTS Application Packages	1.00	8.75
37	Testing ERP Applications	1.00	8.75
38	Testing Reusable Functions	1.00	8.75
36	Supply Chain Testing	1.00	8.50
37	Function Points for Test Measures	1.00	7.00
	TOTAL	47.00	

4	Software Defect Estimating	2.00	10.00
5	Software Project Sizing	1.00	10.00
6	Software Security Issues	1.00	10.00
7	Software Milestone Tracking	2.00	10.00
8	Software Cost Tracking	2.00	10.00
9	Software Defect Tracking	2.00	10.00
10	Benchmark sources: ISBSG, Namcook.	1.00	10.00
11	Software Change Management	2.00	9.50
12	Software Configuration Control	2.00	9.50
13	Case Studies of Software Failures	1.00	9.50
14	Case Studies of Best Practices	1.00	9.50
15	Software Value Analysis	1.00	9.50
16	Outsource Contract Development	1.00	9.00
17	Software Risk Management	2.00	9.00
18	Tools: Earned Value	1.00	9.00
19	Tools: Budgeting	1.00	9.00
20	Tools: Cost Estimating	1.00	9.00
21	Tools: Quality Estimating	1.00	9.00
22	Tools: Cost Tracking	1.00	9.00
23	Tools: Milestone Tracking	1.00	9.00
24	Tools: Project Management	1.00	9.00
25	Tools: Function Points (normal)	1.00	9.00
26	Tools: Function Points (High Speed)	0.50	9.00
27	Test Benchmarks	1.00	9.00
28	Software Measurement and Metrics	2.00	8.50
29	Function Point Analysis: Normal	2.50	8.00
30	Overview of Software Inspections	1.00	8.00
31	Overview of Software Testing	1.00	8.00
32	COTS Acquisition	1.00	8.00
33	ERP Acquisition and Deployment	2.00	8.00
34	Sarbanes-Oxley Compliance	2.00	8.00
35	Multi-Company Outsourced Projects	2.00	8.00
36	Supply Chain Estimating	2.00	8.00
37	Six-Sigma Measurements	1.00	8.00
38	TSP/PSP Measurements	1.00	8.00
39	Metrics Conversion	1.00	8.00
40	Principles of Goal-Question Metrics	1.00	8.00
41	Balanced Scorecard Measurements	1.00	7.00
42	Earned-Value Measurements	1.00	7.00

	Data Base Courses	Days	Value
1	Principles of data security	1.00	10.00
2	Principles of data quality	1.00	10.00
3	Extracting and repairing bad data	1.00	10.00
4	Preventing data theft	1.00	10.00
5	Data and clouds, SOA, and SaaS	1.00	9.50
6	Principles of data organization	1.00	9.40
7	Principles of data architecture	1.00	9.40
8	Principles of data base design	1.00	9.40
9	Principles of data economics	1.00	9.40
10	Data bases and knowledge management	1.00	9.25
11	Building and maintaining a data base	1.00	9.20
12	Building and maintaining a data warehouse	1.00	9.20
13	Building and maintaining a repository	1.00	9.20
14	Building and maintaining web sites	1.00	9.20
15	Overview of commercial data products	1.00	9.00
16	Sharing data with supply chains	1.00	9.00
17	Acquiring data: benchmarks	1.00	9.00
18	Acquiring data: mailing lists	1.00	9.00
19	Acquiring data: customer preferences	1.00	9.00
20	Acquiring data: images, graphics, sounds	1.00	9.00
	TOTAL	20.00	

	TOTAL CURRICULA	Days	Classes
	Executive Education Courses	26.50	33
	Project Management Courses	26.50	43
	Software Development Courses	58.00	52
	Software Maintenance Courses	29.00	32
	Software Quality Assurance Courses	39.00	26
	Software Testing Courses	50.00	37
	Software Project Office Courses	59.00	44
	Data Base Courses	20.00	20
	TOTAL	308.00	287

Gack, Gary; Managing the Black Hole: The Executives Guide to Software Project Risk; Business Expert Publishing, Thomson, GA; 2010; ISBN10: 1-935602-01-9.

Galorath, Dan; Software Sizing, Estimating, and Risk Management: When Performance is Measured Performance Improves; Auerbach Publishing, Philadelphia; 2006; ISBN 10: 0849335930; 576 pages.

Garmus, David and Herron, David; Function Point Analysis – Measurement Practices for Successful Software Projects; Addison Wesley Longman, Boston, MA; 2001; ISBN 0-201-69944-3;363 pages.

Gilb, Tom and Graham, Dorothy; Software Inspections; Addison Wesley, Reading, MA; 1993; ISBN 10: 0201631814.

Humphrey, Watts, TSP – Leading a Development Team; Addison Wesley, Boston, MA; ISBN o-321-34962-8; 2006; 307 pages.

Jones, Capers; "Early Sizing and Early Risk Analysis";
Capers Jones & Associates LLC; Narragansett, RI; July 2011.

Jones, Capers and Bonsignour, Olivier; The Economics of Software Quality; Addison Wesley Longman, Boston, MA; ISBN 10: 0-13-258220—1; 2011; 585 pages.

Jones, Capers; Software Engineering Best Practices; McGraw Hill, New York, NY; ISBN 978-0-07-162161-8; 2010; 660 pages.

Jones, Capers; Applied Software Measurement; McGraw Hill, New York, NY; ISBN 978-0-07-150244-3; 2008; 662 pages.

Jones, Capers; Estimating Software Costs; McGraw Hill, New York, NY; 2007; ISBN-13: 978-0-07-148300-1.

Jones, Capers; Software Assessments, Benchmarks, and Best Practices; Addison Wesley Longman, Boston, MA; ISBN 0-201-48542-7; 2000; 657 pages.

Jones, Capers; Conflict and Litigation Between Software Clients and Developers; Software Productivity Research, Inc.; Burlington, MA; September 2007; 53 pages; (SPR technical report).

Kan, Stephen H.; Metrics and Models in Software Quality Engineering, 2nd

Radice, Ronald A.; High Quality Low Cost Software Inspections; Paradoxicon Publishing Andover, MA; ISBN 0-9645913-1-6; 2002; 479 pages.

Royce, Walker; Software Project Management – A Unified Framework; Addison Wesley, Boston, MA; ISBN 0-201-30958-0; 1999; 406 pages.

Wiegers, Karl E.; Peer Reviews in Software – A Practical Guide; Addison Wesley Longman, Boston, MA; ISBN 0-201-73485-0; 2002; 232 pages.