

Post-Industrial Quality Software

by Allan Baktoft Jakobsen

Hacking software has always been regarded as something chaotic and shady by professional managers. The only problem with this attitude is that the hackers of the Linux net society are now making large software systems of better quality than most traditional companies. The Linux society is challenging Microsoft just as Microsoft challenged IBM. "We're not afraid, the net people say, we're 100.000 brains against 40.000!"

A new paradigm is breaking through with software at its heart: The knowledge society. Things are being turned up-side-down. What is the role of software quality and how will it be achieved in the future? Perhaps the management thinking of the industry society is being confined to traditional production and a new kind of leadership will dominate post industrial software development putting people in the center instead of organizations.

Management versus Leadership

Management and leadership are concepts that reflect two different attitudes to working with long-term ideas. A nice way of explaining the difference is the following [Ledelse af mennesker]:

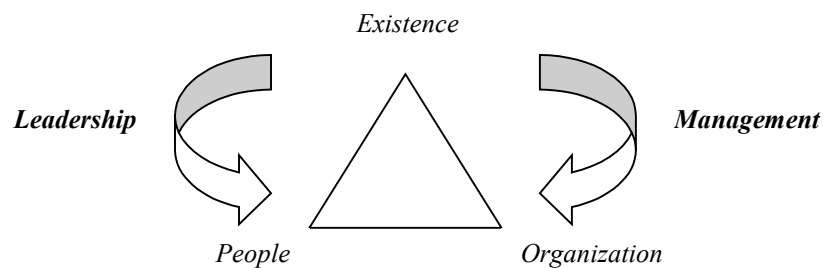


Figure: Management and leadership

The basis of both management and leadership is the existence of long-term ideas, traditionally manifested in the existence of a company. Now, *managers* will setup the organization before reacting to the people's conditions (Clockwise order in the above figure) while *leaders*, on the other hand, will proactively listen to the people before setting up the organization (anti-clockwise order).

In practice, things are not so categorical, but let us consider for while management and leadership in their purest form:

Managers think in terms of organization, that is, hierarchies (divisions, departments), procedures, systems, roles, and responsibility. The employees are resources represented by cells in a spreadsheet (*Excel management*) or entities in a database.

I once worked in a company with a strong management attitude, and the sure thing to get the *group boss* running was forgetting the time-registration. The only way to attract the *department boss* to our office was making problems with overtime payments. The basic philosophy was indeed *command and control*.

Leaders think in terms of people, that is, expertise, talent, commitment and personal character. Leaders will say: ok, here are some people - let's see what we can do with them. Managers will say: Ok, here is the organization - let's see what we can do with it. Consequently, managers will say that people *are* resources - leaders will say that people *have* resources.

Leaders regard brains as resources of the people, where managers regard them as resources (organs) of the organizations. Leaders will focus on the interaction of people with products, while managers focus on the interaction between process and products.

Management thinking

CMM (Capability Maturity Model), ISO 9001 (European standard for production), and BSC (Balanced Score Card) are all successful models for organizing companies. They are used extensively in the software industry.

The CMM, for example, is dominating the way the software industry is working with process improvements (SPI). The levels of maturity are bench-marking the companies' capabilities externally in terms of processes and are internally used as a ladder for improvements. The second highest level, for example, is called the *managed* level, indicating the capability of quantitative control.

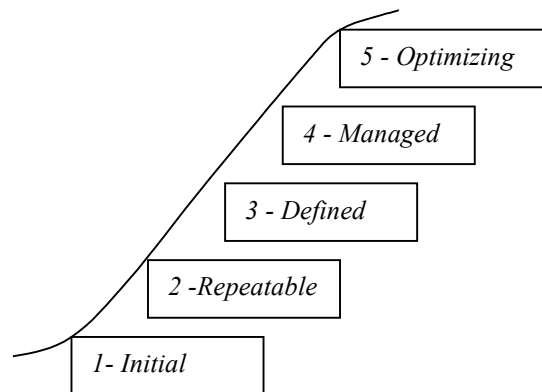


Figure: The maturity levels of the CMM

The ISO 9001 is a European industry standard for production originating from 2nd World War's production of ammunition. The goal of the standard is common procedures to ensuring the uniformity of the products. Clearly, in mass-producing ammunition *quality* means identical functionality of each bullet.

Ideas from the CMM and ISO 9001 are merged in the European maturity models called BOOTSTRAP and SPICE. It comes as no surprise that research has shown that ISO 9001 certified companies are performing better at BOOTSTRAP assessments. [Does ISO 9001 Increase Software Development Maturity?]

The traditional financial control of a company has inspired many theories of similar annual accounts for quality, knowledge, environment, etc. The Balanced Scorecard, for example, reflects this and the lesson learned is often that the challenge is to "re-educate the organizational employees".

Of course, all successful ideas are bound to be criticized.

James Bach [The Immaturity of the CMM] stated that the

- CMM reveres process but ignores people
- CMM reveres institutionalization of process for its own sake
- CMM contains very little information on process dynamics
- CMM encourages displacement of goals from the mission of improving software to the mission of achieving higher maturity level

Although the CMM should be an ideal vehicle for software process improvements, it isn't necessarily in practice. There are quite a few companies where management decides to initiate improvements by making maturity assessments. Some of them fail. Even if they manage to get started, subsequent assessments may show little or no progress on the maturity scale.

Working with SPI as a consultant can be very frustrating leaving you with the heretical feeling that much of your success is due to the Hawthorne effect (or *Placebo* SPI) The psychological mechanisms of the people cannot be underestimated in this game. SPI is no ordinary science experiment and some scientists consequently prefer calling their work *action research*: "We throw a bomb into a company and observe what happens."

Having studied several hundred British companies, business psychologist John Seddon [The case against ISO 9000] stated that

- ISO 9000 encourages organizations to act in ways which make things worse for their customers (Service not in focus)
- Quality by inspection is not quality

- ISO 9000 starts from the flawed presumption that work is best controlled by specifying and controlling procedures
- The typical method of implementation is bound to cause sub-optimization of performance
- ISO 9000 promotes, encourages, and explicitly demands action which cause sub-optimization
- When people are subjected to external controls, they will be inclined to pay attention only to those things which are affected by the controls
- ISO 9000 has failed to foster good customers-supplier relations
- As an intervention, ISO 9000 has not encouraged managers to think differently

He suggests *quality thinking* instead of top-down command and control.

Weinberg [Anticipating Change] says the CMM doesn't cover half of what has to be done, and he suggests *problem solving leadership* rather than *canned processes*.

Kelly [The Complexity Advantage] says that old-fashioned bureaucracies - commanded and controlled by a few leaders [managers] - cannot respond to today's rate of change with the speed and precision of numerous coherent, intelligent, and self-disciplined agents who self-organize with integrity and overtly coordinate their co-evolution.

Adizes [How to Solve the Mismanagement Crisis] suggests the following life-cycle of a company:

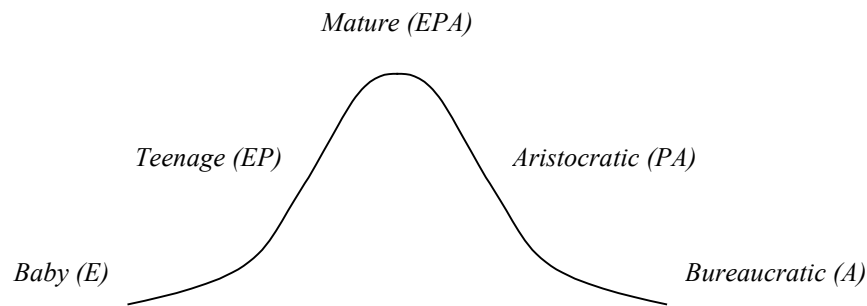


Figure: Adizes company life cycle

The company is regarded as a *super individual* (a kind of *local Gaia theory* c.f. Lovelock) being driven by components of *entrepreneurship* (E), *production* (P), *administration* (A) and *social integration* (I). At the *teenage* stage, for example, the company is dominated by new ideas and growth but also many internal conflicts. At the *aristocratic* stage, the growth has declined and company rests on its laurels. The question is obviously how this idea of (over)maturity maps to the CMM.

Even Humphrey [Managing Technical People], the father of the CMM, sums up 50 years of experience by stating that: "The real thrill of leadership is discovering that you really need to do very little managing. When people are motivated and properly led, they will generally manage themselves. Your job is to manage yourself so you will be a truly superior leader." In practice, it seems that the CMM has been interpreted as a management model, where it was perhaps originally was thought as a model of human behavior.

To summarize the above ideas and objections, I think that what we are witnessing is a struggle between two forces: *Management thinking* versus *Leadership thinking*. Management is based on organizing people, while leadership is based on motivating people. Thus, *mission* is essential to the first where *vision* is essential to the latter.

Motivating people

The above management theories tell us all there is to know about organizing. To examine the secrets of leadership, on the other hand, we have to look at the concept of human motivation.

Maslow classical theory of motivation based of the pyramid of personal needs is being refined by the modern theory of Herzberg.

Herzberg's main point is that the opposite of being *satisfied* is not being *dissatisfied* and that opposite of being *dissatisfied* is not being *satisfied*. The factors of motivation split in two:

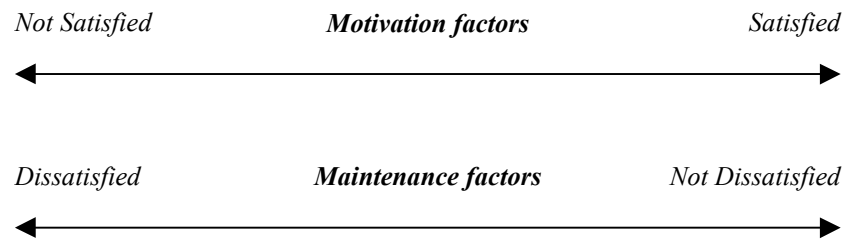


Figure: Herzberg motivation and maintenance factors

A happy person is both satisfied and not dissatisfied. He is happy because he is

- Satisfied about his: job assignments, mutual respect, acknowledgement, influence, and freedom
- Not dissatisfied about his: payment, bonuses, social relations, and title

Herzberg describes the *KITA* approach to motivation carried out by many managers: "Accept the challenge or be *kicked in the ass!*" [What motivates people at work?] This approach is both inelegant, often in contrary with company values, and a physical attack on the person's nervous system, he says. Who wouldn't avoid KITA situations?

This relates to the mechanism mentioned by John Seddon: "When people are subjected to external controls, they will be inclined to pay attention only to those things which are affected by the control." The work is then about *verifying* the mission instead of *validating* the vision. This is the cause of sub-optimization.

In a large European SPI project, a bureaucratic *measurement framework* was set up to control the participants. When they complained about counting the number of people at their meetings etc. instead of focusing on coaching and raising the awareness of SPI, the sponsors called it an *attitude* problem. Which was exactly the truth!

When a big American city focused on the crime rate, it dropped. But the story went that the officers simply didn't report all incidents - the short cut to success.

To leaders, the *diversity* of people is the key to motivation. The relevant theories are the soft ones about people. This could be Adizes' on people's success criteria classified in terms of E, P, A, and I types (see above) [Bottom-Up Process Improvement Tricks]. Or Meyers-Briggs, classifying people in terms of how they get energy, get knowledge, make decisions and act. [Congruent Actions] Having these models in mind, motivation and coordination can be optimized.

Although many companies seem to regard organizing people as more important than motivating them, there is one group of people that few companies have managed to organize: The customers! As long as there is competition all a company can do is trying to motivate customers to keep buying their products or services.

The BPR theory (Business Process Reengineering) suggests shifting the focus to the customers by reengineering the processes on their premises. This kind of customer motivation is a dramatic shift for a company based on production but more obvious for a company based on service.

Applying Herzberg to customer motivation perhaps *quality* is just a maintenance factor.

Creativity and Innovation

It is common to distinguish the production of a company from its creative and innovative activities. *Dreamland* is not to be mixed with *Productionland*.

Creativity is often defined as the generation of new ideas by rearranging knowledge. Innovation, on the other hand, is the first concrete implementation of such new ideas [Creativity, Innovation, and Quality].

Most people consider the creative process hard: Combining the right knowledge requires a lot of excess input, context and domain knowledge, unconsciously information and exformation (aggregated information during

time) - all the things that are so hard to share. There are so many combinations and only so few potential profitable ideas.

Some will argue that the concept of a process makes little sense here. Traditional processes are valuable by their repeatability and their standard definition as basis for common control metrics. New ideas, however, are unique creations that cannot be mass-produced.

The creative process has low quality in the sense that the *yield quotient* (amount of output/amount of input) is low. Then again, knowledge is (still) free and the *absolute* output is potentially very valuable - a good idea is like gold if implemented successfully. Creativity has a *mysterious gleam* and is therefore often left to artists and gurus.

At the individual level, take for example Edward de Bono's ideas of *lateral thinking*. Lateral thinking is about generating and *including* possibilities, in contrary to *vertical* thinking that is about *excluding* possibilities. *Richness* is the issue in lateral thinking - *rightness* the issue of vertical thinking. Creative people say *yes* where others say *no*. Generating problems is sometimes more sophisticated than generating solutions. In his "Six thinking hats" de Bono suggests techniques for individual innovation, that is, how to get on with the bright ideas.

While individual creativity can be fun and may enrich your life, organizational creativity must be implemented in order to be valuable. Innovation is the key here since it involves both ideas and implementation.

In the early 1980's, the US industry was threatened by Japanese competitors. It was believed that only innovation could save the nation, and the big issue was how to stimulate it. Rosabeth Kanter concluded that innovative companies are characterized by a *open culture* integrating relations across the organization. The opposite, a *segmentalistic* (non holistic) organization where departments and divisions are separated with waterproof shutters, are not very innovative. She recommended that innovation should not be *masterminded in a top-down process*, but should be initiated by gathering teams of talented people. Comparing the premises for creativity and innovation, organizations is obviously much like super individuals.

In changing over-mature organizations, Adizes recommends decentralization and separation of short-term EI teams from the traditional PA hierarchy. This should countermeasure the excess administration and the scarcity of new ideas. Of course, the synergy teams (=mature groups) are super individuals themselves and will eventually dissolve, but that is ok: They were never meant to be long-termed.

This reminds me of a high-tech company that decided to introduce a matrix-organization instead of a traditional line-organization. I expected that loosen up the organization by introducing a pool of resources perhaps could create an environment for more innovation. But after only a year, the new organization collapsed: Nobody was taking *responsibility* it was observed and the one fatal question "Who is my boss?" was haunting the organization. It was, however, the same organization I described earlier having a lot of managers, but very few leaders. Perhaps innovation requires leadership where production requires management.

Often the word "innovation" is related to *product* innovation, that is, new fancy products. Software process improvements (SPI) is a good example of *process* innovation, that is, new ways of doing things. If innovation requires leadership, this means that successful SPI should be introduced using the diversity of the people and not through the hierarchy of the organization. [Bottom-Up Process Improvement Tricks] [Enough About Process: What We Need are Heroes]. In other words: to *change* you need leadership - to maintain *status quo* you need management.

Communication

Thinking about what goes wrong in software development, there is one word that comes to my mind: *communication*.

Communication and it's relative, *coordination* (communication of actions) is what it is all about. It's not that it is hard to communicate with computers, it is of course, but at least you know where you have them. It's different with people. Much of the communication between people is based on common assumptions (exformation). That trick saves a lot of words but it is also the root of many problems when the understanding of common ideas is put to the test. Furthermore, cooperation is based on common decisions, which require not only common assumptions but also common ethical values.

Fred Brooks [The Mythical Man-Month] stressed the communication problem in his provoking statement that if your software project is going bad adding resources is *the sure way* to make it worse. The *Babel Tower Syndrome* he called it, referring to the quadratic communication overload when adding resources.

The challenge is, however, that some software systems are now so big that many brains have to work together to build them. Brooks refers to organization as the consequence of the communication problem. Only a tight organization can produce high quality software, he says. He describes *aristocracy* versus *democracy* in software development pointing to the superiority of the first and proposing formation of *surgical teams* dominating the development.

But what about synergy? Many experiments have shown that if you let groups of people solve problems, the solution made by the group together is usually better than the average of the individual solutions. Synergy is when $1+1=3$, that is, when there is an *added value* of working together.

Raymond [The Cathedral and the Bazaar] discusses this in his description of the net society quoting Linus Torvalds: "Somebody finds the problem ... and somebody else understands it. Both things tend to happen quickly."

New paradigm

Which attitude is the better: Management or leadership? Which will prevail? IBM was beaten by Microsoft and perhaps Microsoft will be beaten by the Linux society. Perhaps we are facing a paradigm shift as we move from the industry society into the knowledge society. From production to innovation, from management to leadership.

The old paradigm is not dead, it is just being confined. The validity is being limited as focus shifts. Fire-making is still important but not in focus. The validity of classical mechanics is limited by relativity theory. The secret of the industry society was in fact not the inventions of new ideas such as the automobile or the light bulb but the process of *mass-production* that made the products available to the public and thereby profitable. In the knowledge society, perhaps *mass-invention* will be the big trick.

Focus may shift from products and processes to people. Profit will be made in fulfilling Herzberg needs of immaterial motivation rather than Maslow basis needs of material products. BPR's customer motivation is a fine idea, but why not take the plunge and focus on the people: users as well as developers as in the Linux society? Why re-educate the organizational employees - let the employees re-educate the organization.

Some are comparing the late days of the industry society (= information society?) with the Middle Ages. The world has become a global village where we all have information about what's going on. Few of us, however, are yet in a position to exploit that information. Like the medieval aristocracy had superior communication skills due to the mobility of their knights, there is today a media aristocracy dominating the society by their superior two-way communication skills. The more powerful is the one who is better in communicating. The Middle Ages didn't last, however, and perhaps the paradigm shift to the knowledge society will have parallels to the Renaissance and the Age of Enlightenment. The winning society is the one that enhances the communication of its people's talents. (An example of this is *the Student Help* - an internet service where students at Copenhagen University are offering their individual skills part time to private companies.) Until then, *noble families of knowledge* will dominate.

The internet is a revolution to communication and much of this is due to software. In software development, things are being turned up-side-down as various net societies are emerging. The Linux society is perhaps the best known at this moment because it is directly challenging Microsoft's otherwise dominating computer operating system.

For software, innovation is much more valuable than production. The internet can make the software available for the entire world for download within seconds. Time-to-market is more important than quality for commercial software products as you can instantly dominate the global market if you are first.

Moreover, *riding* the internet is great for collecting pieces for creative ideas and connecting talented people in synergy teams to implement them. The internet environment matches De Bono's and Kanter's recipes for creativity and innovation, and the ultimate separation of EI teams and PA structures suggested by Adizes is today a reality in the net society.

Post industrial software quality

Most of us acknowledge that CMM and ISO 9001 have been useful instruments for raising the awareness of software quality, but perhaps they are just the *crown-jewels* of the industry society's production management thinking. If so, what will software quality mean in the new paradigm and what will be the key areas to improve?

It seem to be the general opinion that the Linux operating system is of very high quality. Raymond describes a quality mechanism calling the *Linus Law*: "Given enough eyeballs, all bugs are shallow." It seems that mass-invention is the trick behind the quality. It is interesting since it seems that mass-production raised the problem of quality in the industry society (c.f. the history of ISO 9001). The challenges of *completeness*, *correctness*, and *consistency* may have obvious solutions in a software inventing net society. *Lateral quality assurance* is perhaps the buzz word.

The method reminds me of a trick I used as a math instructor at the university: When I corrected the homework of the students, I always compared my solutions with the work of the best students in the class thus ensuring perfection again and again.

What about the communication problem? Raymond describes Linus Torvalds's development method: "Release early and often – delegate everything you can." He adds: "If you have the right attitude, interesting problems will find you." The common language of these hackers and nerds is computer code. When exchanging ideas on computer system features, source code is a very precise language. Open source code is consequently of extreme importance for the net society since it supports the building of a common idea that communication is all about. Releasing early and often is an essential communication trick in this iterative evolution of quality software. Brooks's Babylon Syndrome and his ideas of software evolution as an entropy increasing entropy eventually leading to collapse of the product are apparently being turned up-side-down. Maybe only a loose net can create high quality software.

Raymond stresses the importance of having users. Beta-testers and co-developers seem to work in perfect symbiosis: "Linus was constantly keeping his hackers/users stimulated and rewarded – stimulated by the prospect of having an ego-satisfying piece of the action, rewarded by the sight of constant (even daily) improvement in their work." This almost sounds like animal training.

What enables so many people to work together building large computer systems? Raymond says: Motivation, and leadership, not based on power, but common understanding.

In an interview on what motivates the free developers in the net society, Linus Torvalds says, that it *feels good* to have given others something useful. When 10 people work 1 hour at the project, they all get 9 hours for free since they share the results. Both developers and users are contributing this was. Linux is free since the traditional consumer/producer distinction makes little sense here - giving the same thing to a thousand consumers is not more expensive than giving it to just one, he says. This suggests a kind of natural economy in the net society based on brain time (or intellectual capital).

Studying the net society in details may reveal a handful of basic mechanisms which suggest the application of complexity theory [The Complexity Advantage] to explaining it's *macro* behavior. The analogy to self-organizing biological systems is sometimes mentioned. It would surprise me if these mechanisms were far from the ideas in for example Herzberg's motivation theory.

The coordinators of the net society offers pure leadership to the people, who in return donate expertise. Not in a top-down or vertical hierarchy but in a lateral way. Hackers and nerds by circumstance share a common vision. Because if they don't, they will just diverge in the net with no hard feelings.

In the mean time, traditional managers can only regard the *codex* of the net society as fanaticism. Philosophers, on the other hand, could see this dialectical *self-organizing* as the *synthesis* of the management *thesis* and the leadership *anti-thesis* (cf. Kant and Hegel). Some day, perhaps, aristocracy and democracy will be working hand in hand and traditional management thinking will be confined to industry production of material goods. The new immaterial products will emerge from the talents of people that will organize exactly as much as necessary.

References:

1. James Bach: The immaturity of the CMM. American programmer, Sep. 1994.
2. James Bach: Enough About Process: What We Need are Heroes. IEEE Software, Mar. 1995, pp. 96-98.

3. John Seddon: In Pursuit of Quality - The Case Against ISO 9000. Oak Tree Press 1997.
4. Robert S. Kaplan: The Balanced Scorecard. Harvard Business School Press 1996.
5. Gerald Weinberg: Quality Software Management Vol. 3: Congruent Action. Dorset House 1994.
6. Gerald Weinberg: Quality Software Management Vol. 4: Anticipating Change. Dorset House 1997.
7. I. Adizes: How to Solve the Mismanagement Crisis. MDOR Inst., Los Angeles, 1979.
8. Watts S. Humphrey: Managing Technical People. Addison-Wesley 1997.
9. Jørn Johansen: Does ISO 9001 Increase Software Development Maturity. Proceedings of the 24th EUROMICRO Conference 1998. Vol. 2, IEEE Computer Society.
10. Paul E. Plsek: Creativity, Innovation and Quality. ASQ Quality Press 1997.
11. Danish Defence Centre for Leadership (Forsvarets Center for Lederskab): Ledelse af mennesker. 1998.
12. Mogens Stiller-Kjærgaard: The Unified Company - A Revitalisation Concept. PA Consulting Group, Denmark. 1995.
13. Nils Bech: Synergy or Just Symbiosis? Common Goals Make the Difference! Proceedings of the Fourth European Conference on Creativity and Innovation. Geschka & Partner 1993.
14. Edward de Bono: Lateral thinking - Creativity Step by Step. Harper & Row 1970.
15. Rosabeth Moss Kanter: The Change Masters - Innovation and Entrepreneurship in the American Corporation. Simon & Schuster 1985.
16. Frederick Herzberg: What Motivates People at Work? Harvard Business Review, Jan./Feb. 1969.
17. Frederick Herzberg: The Motivation-Hygiene Concept and Problems of Manpower. Organization Development: Managing Change in the Public Sector, 1976, pp. 178-182.
18. Abraham Maslow: Motivation and Personality. Harper & Row, 1954.
19. J. Daniel Couger: Motivating and Managing Computer Personnel. John Wiley & Sons, 1980.
20. Gerald Weinberg: The Psychology of Computer Programming. Silver Anniversary. Dorset House 1998.
21. Frederick P. Brooks, jr.: The Mythical Man-Month (Anniversary Edition). Addison-Wesley 1995.
22. Eric S. Raymond: The Cathedral and the Bazaar. <http://www.redhat.com/knowledgebase/cathedral-bazaar.html>.
23. Interview with Linus Torvalds: What motivates free software developers? First Monday, http://www.firstmonday.dk/issues/issue3_3/torvalds/index.html
24. Susanne Kelly: The Complexity Advantage: How the Science of Complexity Can Help Your Business Achieve Peak Performance. McGraw-Hill 1999.
25. Allan Baktoft Jakobsen: Bottom-Up Process Improvement Tricks. IEEE Software, Jan./Feb. 1998, pp. 64-68.
26. Allan Baktoft Jakobsen: The Software Prince. Proceedings of the Machiavelli at 500 seminar at Manchester Metropolitan University 1998.

Allan Jakobsen has an educational background in mathematics and physics from Copenhagen University and Dartmouth College. He has been working in the software business with maintenance, design, coding, test, quality assurance, and as an SEPG member. He is working for the Danish company DELTA as a consultant in best software practices and process improvements and runs his own company MindMate in his spare time.