

# THE VIEW FROM THE CUTTING EDGE 3

or

*Things I Learned in the Great Software Wars*

by Larry Bernstein

On Design

On Programmers

On Programming

On Object Oriented Design

On Managing People

On Manufacturing

On Maintenance

*Ten Steps to Software Maintenance*

## ON DESIGN

Along with features and structure in written design specifications, include estimates of reliability, response time and throughput with estimates of computer sizing.

Decouple conversion system databases from application system databases by using application system transactions to initialize the database. Otherwise, unnatural constraints of freezing the data base design prematurely seep into the development program. This has the tendency to prevent redesigns aimed at eliminating bottlenecks that can result in missing the throughput, sizing, and response time commitments.

“Designs of different vintages, built at different times, built on existing systems, lead to complex tradeoffs.”

*Sumner*

“Before committing the design of a complex system be sure solutions are available for the basic technical problems that limit the system; leadership in [software] engineering requires understanding basic technical processes; and a good [software] engineer personally validates the analysis [in a prototype].”

*Elmendorf commenting on Friis's Seventy-Five Years in an Exciting World.*

“Today we design data bases by creating a bunch of boxes and then mashing the data into the boxes.”

*Vyssotsky, 1980*

“Software is only one interpretation of the reality of the problem it is solving.”

*Jackson*

Structure clashes between the problem and the solution or careless use of data structures leads to unacceptable throughput and response time.

“One of the problems associated with large data base systems is the lingering death associated with corrupted data bases. Unfortunately, the program that first discovers the corruption is usually the victim and not the perpetrator. Two tedious processes then ensue. First, the result must be corrected, and, second, the cause must be discovered. This latter problem is very complex in that usually the perpetrator is long since gone and has left few to no clues behind. Some suggestions:

1. Put in early alarm systems to detect some problems as they occur and try to contain them; for example, validity check all records as they are entered into the data base.
2. Instrument the system to keep clues, for example, store the program ID and date of update.
3. Reduce the potential problems by consolidating data base access routines which handle inverted files.”

*Martin*

Software designers need to visit customers.

“Build a tree, not a pile of leaves”.

*Mashey*

“It isn’t so astonishing, the number of things that I can remember, as the number of things I can remember that aren’t so.”

*Twain*

Post-operational design evaluation is essential.

Hold design reviews. The most successful ones result in simplifying the design.

When in doubt, throw it out.

### **ON PROGRAMMERS**

Programmer productivity varies by as much as 25:1 depending on the individual. Program productivity techniques may each result in a maximum of 2:1 or 3:1 individual improvement.

Appoint module owners.

Programmers perform best from sun-up to sundown. Machines work round the clock.

Programmers close to the machine are productive, happy programmers.

Programmers want to document, comment, and control – bosses don't provide the tools or time.

Programmers like to learn and grow; they need change.

Programming to improve maintainability results in maintainable software.

## **ON PROGRAMMING**

The “development environment” is the collection of software and hardware that programmers use to control the official source, edit, compile and finally test the programs. It is sometimes called the development machine. The “execution environment” is the collection of software and hardware that houses the working application. It is sometimes called the “target machine”.

Do not make flow charts deliverable documents. Although they may be useful in the initial design, they are impossible to keep up-to-date. There is some evidence that they lead to poorly structured code. Design documentation should be in the program comments. Comments should follow a standard structure.

“High level languages are useful:

- to concentrate on design issues rather than machine quirks
- to shorten program text leading to fewer coding errors
- to permit coding in a machine independent way
- to make change easier and maintenance easier because of
  - more understandable code
  - well defined interfaces
  - data structuring capability
  - reliable flow control structures ”

*Pinson*

A program is something you build for yourself. A software product is something you build for somebody else.

## **ON OBJECT ORIENTED DESIGN**

Tips for adopting object-oriented methods:

1. Get expert help early.
2. Invest \$10k in tools and six months of training for each member of the development team.

3. Limit the number of Object Oriented Architects to 20% of the staff.
4. Limit the number of object classes to 0.5% of the total functions.
5. Limit the number of object classes that have inheritance greater than three to no more than one-third of the total.
6. Increase the design phase to 30% of the life cycle.
7. Model performance early
8. Pilot before deploying.
9. Use unit test drivers.
10. Assign work by feature teams.
11. Expect to break even after one year of the first release; afterwards, expect a 3:1 productivity gain.

### **ON MANAGING PEOPLE**

Tell all the Truth but tell it slant -  
 Success in Circuit lies  
 Too bright for our infirm Delight  
 The Truth's superb surprise  
 As Lightning to the Children eased  
 With explanation kind  
 The Truth must dazzle gradually  
 Or every man be blind –

*Emily Dickinson 1868*

People need lots of stroking with just a little poking.

“If the desire to kill and the opportunity to kill come always together, who will escape hanging?”

*Twain*

The quality of your managing ability is directly proportional to the quality of your questions.

People whose values do not match those of the organization become malcontents.

Make sure that incentive compensation is linked to performance over which the beneficiaries have control.

*Fortune Magazine*

Don't underestimate the power of bigger bucks to attract higher-quality workers and improve service.

*Fortune Magazine*

There is much injustice in the name of equality.

“Since customer service and innovations are increasingly important competitive weapons, it doesn’t pay to create a sullen, dispirited or burned out work force. Some ways to turn down the heat:

1. Don’t let work turn into an endurance contest
2. Be flexible in work hours
3. Ask people how to restructure the work
4. Keep punishing hours (overtime) reserved for crises, not part of the standard drill.”

*O’Reilly, Fortune Magazine, March 12, 1990*

To be challenged is to be uncomfortable. Tell your people you believe in challenge and they can expect to change jobs when they master their current jobs. Explain that with this opportunity for growth they have the obligation to help out if there is trouble in their former jobs that their successors can’t solve. So, rotate your people’s assignments. Software people should not be doing the same thing for more than three years.

Too many software people have one year’s experience ten times, not ten year’s experience.

“When I reflect on the number of disagreeable people who I know have gone to a better world, I am moved to live a different life.”

*Twain*

“Why is it that we rejoice at a birth, and grieve at a funeral? Is it because we are not the person concerned?”

*Twain*

“Why do you sit there like an envelope without any address on it?”

*Twain*

“It takes your enemy and your friend, working together, to hurt you to the heart; the one to slander you and the other to get the news to you.”

*Twain*

“Ain’t we got all the fools in town (or on the project) on our side? And ain’t that a big enough majority in any town?”

*Twain*

“Noise proves nothing. Often a hen who has merely laid an egg cackles if she has laid an asteroid.”

*Twain*

“It used to be a good hotel, but that proves nothing - I used to be a good boy.”

*Twain*

“I can stand any kind of society. All I care to know is that a man is a human being - that is enough for me; he can't be any worse.”

*Twain*

“I would rather have my ignorance than another man's knowledge, because I got so much more of it.”

*Twain*

“Few things are harder to put up with than the annoyance of a good example.”

*Twain*

“Barring that natural expression of villainy which we all have, the man looked honest enough.”

*Twain*

“April 1. This is the day upon which we are reminded of what we are on the other 364.”

*Twain*

“You made me like him, and did it without trouble...you made me better satisfied with myself than I had ever been before.”

*Twain*

“Nothing so needs reforming as other people's habits.”

*Twain*

“Duties are not performed for duties' sake, but because the neglect would make the man uncomfortable. The man performs but one duty - the duty of contenting his spirit, the duty of making himself agreeable to himself.”

*Twain*

“If you pick up a starving dog and make him prosperous, he will not bite you; that is the principle difference between a dog and a man.”

*Twain*

“His answers were so final and exact that he did not leave a doubt to hang a conversation on.”

*Twain*

“Compliments always embarrass a man. You do not know anything to say. It does not inspire you with words. There is nothing you can say in answer to a compliment. I have been complimented myself a great many of times, and they always embarrass me - I always feel that they have not said enough.”

*Twain*

“ I can live for two months on a good compliment.”

*Twain*

“ When you can not get a compliment any other way, pay yourself one.”

*Twain*

“I like criticism, but it must be my way.”

*Twain*

“ War talk by men who have been in a war is always interesting; whereas moon talk by a poet who has not been to the moon is likely to be dull.”

*Twain*

“He had only one vanity; he thought he could give advice better than any other person.”

*Twain*

“At 50, a man can be an ass without being an optimist but not an optimist without being an ass.”

*Twain*

“Participatory management, while an improvement, takes more time...the boss has to spend hours providing general direction, guiding sharp decision making and wandering around to keep an eye on things...which increases the level of ambiguity and stress a manager must live with.”

*York (Citibank)*

When someone comes to you and says they want a change of assignment, help them consistent with your short-term needs but don't keep them for more than six months. Do not keep them against their will.

Find the indispensable people and reassign them.

Purge weak performers.

The highest form of charity is to help another become productive.

“A good manager has good people.”

*Casulli, private conversation*

Career development is the employee's responsibility. Providing an environment for growth and the opportunity to learn is manager's responsibility.

“Last year's success is this year's mediocrity”.

*Kellogg*

Share with your people what is expected of you and your view of how well things are going.

Admit your mistakes.

***To De-motivate:***

Don't allow people to do what they are paid to do.

Don't allow people to see the results of their work.

“When you employ a precision inconsistent with the inherent accuracy of the management process, you engender mistrust.”

*Dacey*

Protect your people from politics.

Get behind the data to the facts.

Seek respect, not friendship.

Good managers focus on people and not controls or procedures.

“Genius is one percent inspiration and ninety-nine percent perspiration.”

*Edison*

Set a theme for the organization and tell your people what it is. Repeat it often and consistently. Establish specific short-term goals so people can measure their own progress.

A manager should talk daily with those directly reporting and visit those two levels below weekly. Gather information and show involvement.

When time is short and several people want to see you, give first priority to those lowest in the organization. Their problems are probably quickly solved and most pressing. Let your boss wait.

When there is a decision to be made, force your people to present several alternatives with quantitative data as well as qualitative options. Seek their recommendation and, in most cases, accept it. If most of the time you find yourself unable to accept their recommendation, replace them.

Treat each person with respect and dignity; value individual and cultural differences. Communicate frequently and with candor. Give people the authority to use their capabilities to the fullest to satisfy customers.



When the values comprising AT&T's "Our Common Bond" were determined, "Respect for Individuals" was listed first. That's because respect toward one another is the foundation for everything – mission, strategies, and goals.

Respecting one another isn't just a creed or motto; it is a clearly demonstrated behavior. It's saying "please" and "thank you" to one another; it's seeking and using the opinions, ideas, and expertise of all people. It's personally affirming the value of one another daily by:

1. Acting with integrity - keeping our promises and admitting our mistakes.
2. Fostering and conducting open, non-judgmental, two-way communications with others – regardless of level or position.
3. Considering people's needs before implementing something that affects them – and always communicating ahead of time why it is being done.
4. Taking time to talk personally and getting to know the people with whom you work – their skills, experience, needs, and goals.
5. Including people in decision-making when the decision will affect them or when they have something to contribute. This includes sharing problems that affect your group and dealing with them together.
6. "Actively listening" to others by paying attention and showing that you understand their viewpoints.
7. Encouraging and being objective about upward feedback and constructive criticism.
8. Building up people's self-confidence so they are better able to respond and act on their empowerment.
9. Balancing business/work place needs with personal needs, and encouraging and allowing others the flexibility to do the same.

"People perform to the Standards of Their Leader."

*Bechler, Practical and Quality Management Workshop.*

"The difference between the Japanese teian system and the American suggestion system is in the placement of emphasis. When participation and guidance are emphasized, it becomes the teian system. When the emphasis is on effect and result, then it becomes the suggestion system."

*Tozawa*

Tozawa asserts that both the teian and the suggestion systems are based on motivational ideas of Douglas McGregor:

- The average human being does not dislike work.
- People will exercise self-direction and self-control in service to goals to which they are committed.
- Commitment to goals is an outcome of rewards. The highest rewards are ego-satisfaction and self-actualization.
- The average human being learns, under proper conditions, not only to accept but also to seek responsibility.

- The capacity to be creative is widely distributed in the population.
- Under the conditions of modern industrial life, the average person's intellectual potential is only partially utilized.

## **ON MANUFACTURING**

Software manufacturing is a systematic approach to system building, deliverable documentation production, configuration identification, change control, and packaging for delivery.

Manufacturing is in-line, not overhead.

Production skills must be valued in manufacturing organizations.

Staff manufacturing at 5% of the programming staff.

Use technicians, not program designers, to manufacture software.

Let manufacturing technicians run the computers. Call it a software factory.

All releases to system integration and the field must go through manufacturing. When schedules get tight, don't bypass manufacturing; expedite it.

## **ON MAINTENANCE**

A glitch is a bug that has not been escalated; a bug is a fault that has not become a crisis.

When something doesn't work today that worked yesterday, ask, "What changed?"  
That's where the problem is.

"Assigning repair responsibility to the 'green horns' is ... the greatest fallacy of all."

*Belady and Lehman*

### ***Ten Steps to Software Maintenance***

1. Maintenance is driven by formal Change Requests (CRs).
2. Each module has a single owner who is responsible for maintenance and on-going development.
3. Each CR is assigned to a single person.
4. Each step of CR resolution is directed/tracked as an action item.
5. All CR investigations are documented.
6. Decision to accept/reject each CR is made by one person, the manager responsible for the release.
7. Child CRs direct the implementation when multiple modules are involved.  
Each module affected has a unique CR number.

8. Have independent test and manufacturing groups.
9. Employ identical Change Request processes to controlling changes to user documentation
10. To get your best people to maintain programs, ask them and then reward them for doing it. Rotate people between design, test and maintenance.

Once a critical organization size is reached, the rate at which problems are found and fixed tends to be relatively constant. Increases in this rate can be realized from improvements in tools, not by adding people.

At any instant about one-fifth of the modules should be undergoing reprogramming to ease maintenance without adding features. This frees those adding features from harsh performance constraints.

When the size of the maintenance staff equals the size of the development staff, cap the system. Start to develop a replacement system.

Invent a name such as “Version 2” or “Next” or “Upgrade” and begin development of a new system. Anything too hard or too large for the maintenance staff can then be deferred to the upgrade, allowing the system to mature incrementally.

“Leave the program better structured than you found it.”

*Jackson*

“When fixing a program bug, make the largest change you can to improve maintainability.”

*Jackson*

Half of all changes will be due to design flaws, half to programming errors.

A product is mature when enhancement, not reliability, is the main topic of discussion. Once the product is mature, reduce the size of the maintenance staff 10% per release.