

Business is increasingly concerned with issues of sustainability, environmental sensitivity, and social responsibility.

A great deal has been written about sustainability and most aspects of the enterprise have been examined to discover policies and practices that will enhance overall sustainability.

One surprising exception: Information Technology (IT) with all those computers, software, and databases — along with all those people engaged in development and maintenance. Perhaps this omission arises from the fact that IT has become a commodity, like rent and utilities, merely something that every business must have. A sad contrast to the heady days of the 1970s when IT was a source of innovation, cost savings, and strategic advantage.

Whatever the reason, it is a mistake not to question the role of IT with regard sustainable business. Four aspects should be considered: cost, size, people, and adaptability.

### Cost

Consider some cost factors:

- Data centers (those we know about in the US and Europe) use 416 terawatts of electricity per year, **40% more than the entire electrical consumption of the United Kingdom.** Despite ongoing efforts, less than 10% of that energy comes from renewable sources.
- Known annual spending for IT, amounts to more than 2 trillion U.S. Dollars.
- Of every dollar spent on projects to implement new business capabilities; roughly 65 cents is waste. Contracted for software is not delivered at all, is delivered but is unusable, is delivered with less functionality, is delivered late and/or over budget.
- 8 million people in the U.S. are employed in information technology with 1 million unfilled positions.

Individually, each of these factors call into question the viability, the sustainability, of information technology as it is currently practiced. Collectively they suggest it is only a matter of time before the entire effort collapses upon itself.

To understand why so much money, so much energy and so many people are involved in information technology, it is necessary to consider the “size” of IT — or more accurately, the volume of software, the programs and the data, that actually constitute IT.

### Size

To contemplate size, it is necessary to have some sort of measure — some way to say this is 10 units of X and that is 20 units.

A flawed, but useful in this context, measure frequently applied to software is LOC; standing for Lines Of Code.

A program is a sequence of instructions telling the computer what to do. Colloquially, each instruction is a “line of code.” A program with ten instructions has 10 LOC, one with a hundred has 100 LOC.

Typical business programs have ten thousand LOC, and a typical company may have 10–100 such programs. The “size of IT” for that company then varies between 100,000 and one-million LOC.

The program that everyone knows as the Windows 10 Operating System has around 50 million LOC. The program known as Facebook has around 62 million LOC. Google — the programs that implement all of Googles services — search, maps, Gmail, Calendar, et. al. — is 2 Billion LOC!

Programs are not the only determinant of size. The data that those programs accumulate and operate on is an equal factor. The measure for a database is the number of “bytes” it contains, with a byte being roughly equivalent to a single alphabetic letter. *This sentence is 34 bytes in size.*

A typical business database will be several gigabytes (gigabyte equals one billion bytes) in size. A two-gigabyte database holds roughly the same number of letters (each one byte in size) as contained in a shelf of books 20 meters (66 feet) in length.

So large are the databases used by Google, Facebook, some government agencies, and even individual websites, that they are measured in Terabytes (trillions of bytes) and Petabytes (1,000 Terabytes).

Projected sizes of data stored or transmitted, e.g. Web traffic, are or will be measured in Exabytes (1,000 Petabytes), Zettabytes, Yottabytes, Xenttabytes, Shilentnobytes, and Domegegemetrottebytes.

Databases and programs, this massive, require very large, very powerful, very fast computers to house them.

Moreover, programs that large require hundreds to thousands of developers to create them — along with the costs of buildings, environmental controls, and local housing. Google employs 25,000 developers to write and maintain that 2 billion LOC software. Silicon Valley no longer has sufficient housing to accommodate the hundred-thousand plus developers working there.

Massive scale is the primary reason that IT is not sustainable.

## People

The size of programs increases the cognitive burden placed on those that write them. It is, quite literally, impossible for any individual or even teams of individuals to understand or comprehend the whole of programs of that size. This makes them prone to errors, security failures, and unpredictable consequences.

The amount of knowledge required to be a “full stack” software developer is equivalent to 2–3 years of college level education and can take a decade to fully acquire.

Human beings charged with creating software of such magnitude, writing all those thousands of programs, are simply not up to the task — suffer from significant cognitive overload. This, in very large part, accounts for that 65% waste figure in software development cited earlier.

Beyond the technical knowledge required of programmers and software developers are a host of “soft skills” like communication, teamwork, creativity, and insight.

An important factor in business sustainability centers on people, people who have the same sort of “soft skills” just noted for software developers. The business community has been aware of this since the early 1990s and numerous books and articles have been written about “learning organizations” and “modern polymaths.”

Unlike the business community, and especially that subset concerned with sustainability and social responsibility, the IT community is doing next to nothing to develop and support this kind of individual.

Typically, the kind of software currently developed by IT, severely constrain human abilities to a kind of clerical minimum; automate humans out of the systems wherever possible; and focus on replacing human with “artificial” intelligence.

Douglas Engelbart’s program to use computers and computing as a means to “augment human intelligence,” Alan Kay’s vision of the Dynabook, and even Steve Job’s vision of the computer as a “bicycle for the mind,” have been all but abandoned.

IT, as currently constituted, is a major barrier to “creation of more high-quality jobs with dignity and purpose,” goals expressed by B Corporations and others committed to sustainable and environmental and social goals.

### Adaptability

Becoming sustainable implies change; frequently rapid change, as current business practices and organizational goals are examined and alternatives defined when necessary. Technological innovations, new regulations, even new business opportunities; are additional factors that introduce change and mandate rapid responses.

It is possible for the business to develop means for innovating, adapting, and evolving in the timeframes required to be effective. That is seldom the case with IT, especially in large organizations.

The size of programs, those millions of lines of code, is directly responsible for the months required to isolate and make changes to existing software. Size is also the biggest factor accounting for the, on average, months and years it requires to develop new business software.

The average IT project still clocks in at a little over a year of elapsed time from inception to delivery. Two to three-year projects are common.

More often than not, especially since the advent of the Internet, required business change is measured in weeks to a few months. Clearly, IT cannot keep up.

Worse, businesses are bound to systems developed as long as a decade ago and reflecting the way that business was done then, not as it needs to be done today.

Those legacy systems tend to be large, complicated, poorly understood (no one that developed them is still around) and the challenge of replacing them is daunting.

### Conclusion and Alternative

IT is not, in and of itself, sustainable. IT is a major, if not the primary, inhibitor of business sustainability.

But is this as it must be? Do we simply accept the way things are and do our best despite the limitations imposed by IT as currently practiced?

Alternatives are possible.

1) Reduce the LOC of programs. Alan Kay initiated a program addressing the LOC problem and demonstrated the possibility of reducing a million LOC program to ten-thousand LOC — a two order magnitude reduction. This would not only reduce the data center demands, but would make programs more comprehensible and easier to maintain and evolve.

In the early 1990s it was demonstrated, consistently, that the right perspective for designing software, at that time it was object orientation, could reduce program complexity, and therefore the time and cost required to create new software by at least an order of magnitude.

2) Find a way to eliminate large centralized databases. Keep the data distributed and in place where it is created. For example, patient data on the patient's cell phone (or other personal device) instead of copied (multiple times) and centralized in the databases of drug and insurance companies who exploit that data for their own, not the patient's, benefit.

Decentralizing data would offer all kinds of social benefits. Hacking to steal identities and credit card information would not be profitable if it had to be done piecemeal. Privacy and freedom from commercial exploitation would be another.

3) The smart phone most of your employees carry around has the same computing power as an IBM mainframe computer of the 1990s.

It is possible to conceive of and design a collection of small programs, running in phones or tablets, that could replace the millions of lines of code running in a data center.

While you are doing this, eliminate the operating system that drives your phone. That operating system is a relic of the 1960s, replicating the structure and purpose of large mainframe computers of that era.

Radical? Certainly! But quite possible.

4) Instead of focusing on the computer and what goes inside it, the direct cause of those massive, immensely complicated, artificial systems, focus on the complex, dynamic, business system that is your business and figure out how to directly modify that system.

Focus on the natural (outside the computer) instead of the artificial (inside the computer).

Take the ideas (not the implementation) of Objects, Domain-Driven Design, and Extreme Programming seriously and find a way to build and deploy small special purpose bits of software; each of which directly supports one element of the business.

5) Use software to augment human intelligence and abilities instead of trying to replace them with inferior artificial substitutes.

Organizations and businesses serious about sustainability, environmentalism, about human integrity, and social well-being are well advised to start exploring alternatives to the way that IT is currently constituted and practiced.