

Why it is Important that Software Projects Fail

An essay on computer productivity

DRAFT

Dr A Berglas, February/October 2008
AXnthy@Berglas.org (Feedback welcome, remove X.)
Copy at will but provide attribution.

Abstract

This paper boldly challenges the long established misconception that the catastrophic failure of expensive software projects is detrimental to society. Historical analysis of bureaucracies such as the Australian Tax Office shows that massive software automation has not increased their real efficiency since the 1950s. Any increase in the efficiency of individual workers has simply been consumed by increased bureaucratic complexity, as predicted by Parkinson's law. As the primary net effect of software is to facilitate bureaucratic complexity it is therefore essential that software projects fail if society is to function effectively. In this way the heavy burden of guilt can be lifted from the shoulders of the numerous project managers that have subconsciously devoted their careers to ensuring that projects rarely, if ever, succeed.

Introduction

Many have bemoaned the fact that software projects rarely live up to their expectations, and many are completely abandoned at great expense. When engineers set out to build a bridge a bridge gets built. Our water ways are not cluttered with abandoned structures that cannot be made to work; bridge building is almost always successful. And the constructed bridges almost always function correctly and are usually built fairly close to budget. But when a large team of programmers sets out to develop software it is very uncertain whether anything of use will be delivered at all.

Some put this down to organizational culture. If an engineer were to deliberately under specify a load bearing beam in order to cut costs, they would be sent to jail. But when a programmer cuts a corner they get promoted. Others suggest that the problem domains are fundamentally different. But all commentators assume that failure is detrimental to society. This paper sets out to correct that misunderstanding.

Berglas's Corollary to Parkinson's Law

Parkinson (1955) is a seminal work that analysis the growth of bureaucracy. It proves the Law of the Multiplication of Work, and provides empirical examples that include the growth of the British admiralty compared to the decline in the number of ships, and the growth of the colonial offices during the decline of the empire. The paper develops scientific formulas that accurately predicts the growth of any bureaucracy depending on numerous parameters, none of which relate to the amount of work to actually be performed.

But writing in 1955, Parkinson could not have foreseen the massive impact that computerized automation would have in the following decades. This paper updates Parkinson's law with

Berglas's corollary, namely that no amount of automation will have any significant effect on the size or efficiency of a bureaucracy.

Empirical Evidence

To see the effect of the corollary in practice, we will consider bureaucracies whose essential function has remained unchanged since 1955, ie. before computer automation. A good example is the Australian Tax Office.

The first aspect to be automated was the basic processing of tax returns. This included calculating tax due, correlating them with employer documents and the printing and reconciliation of refund cheques. This was performed on large and expensive mainframes that had less computing power than is now found in basic mobile telephones. But even one ancient mainframe could perform the work of a thousand clerks reconciling accounts by hand.

Automation has now spread to all aspects of the tax office. Tax returns are entered electronically over the web, analyzed and processed by a number of complex computer systems, and refunds or payments are processed via direct bank deposits. Most returns are never touched by a human hand. The internal management systems are also highly automated, from the allocation and tracking of audits to processing their internal payroll and benefits systems.

Basic office processes have also been greatly improved. Back in 1955 writing a memo was a substantial undertaking. They had to be written out by hand or dictated to a stenographer. Then a typist from the pool had to be allocated, and any errors required the document to be retyped in full. Moreover only three or four copies could be made unless time consuming Gestetner machines were used, and all copies had to be distributed by hand.

The introduction of the word processor and photocopier in the 1980s meant that memos could be written and copied far more easily. E-Mail systems in the 1990s then provided a mechanism for the efficient and instantaneous distribution of memos to dozens if not hundreds of recipients. The effect of this is that in 1955 the average clerk would only receive a handful of memos each week, and rarely write one. Whereas today even the most junior clerk can receive many dozens of emails every single day.

So given this huge increase in automation, the question arises as to whether a modern bureaucracy could function without this automation? So for our example, suppose we took each and every computer out of the tax office, from personal computers to large mainframes, so that all the processing had to be performed completely manually. Could the tax office still function effectively within the same budget?

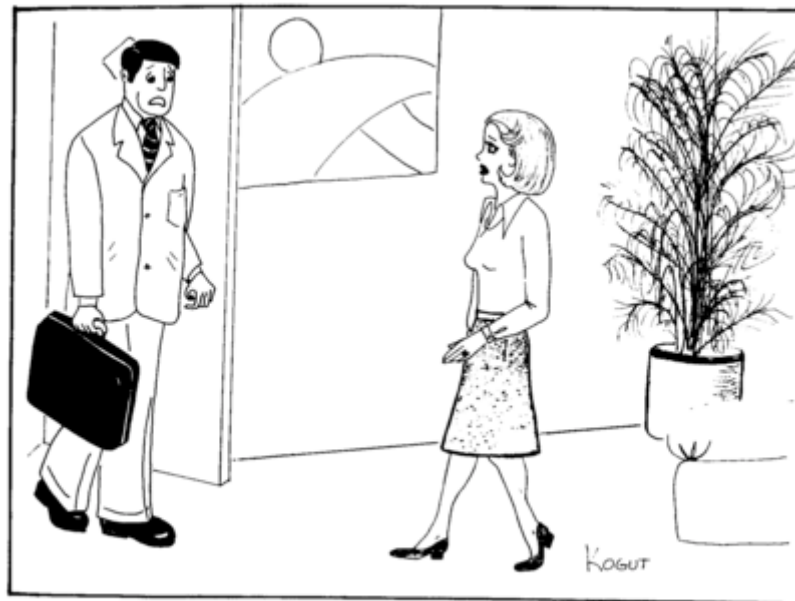
The answer is, obviously, yes they could. We know this because in 2007 the tax office's internal budget was AU\$11.4 billion, or 1.23% of GDP. In 1955 it performed essentially the same task without automation for A£66.7 million which was 1.33% of the 1955 GDP. The difference is not statistically significant. (Normalizing by GDP (essentially the sum of everyone's earnings) accounts for the growing population and inflation.)

To many this is a surprising result. How could the staggering amount of automation instigated

over the previous fifty years not produce any meaningful effect on productivity? This will be examined in the next section but it is an empirical and undeniable fact that bureaucracies have grown, not shrunk.

And for those that would be quick to blame government sloth, similar results can also be shown for private enterprises. For example the banking industry today performs essentially the same function as it did in 1955, when bank accounts were all reconciled by hand. Yet the banking industry has grown 189% as a proportion of GDP.

Effect on Society



The cartoon above was published in the early 1970s (in *Creative Computing*). It reflected the widespread fear and dismay at the time amongst bureaucrats who understood the enormous power of even a single mainframe computer. However, with an understanding of Berglas's Corollary they now know that they never had anything to fear. Indeed, office work now represents over twice the proportion of GDP that it did in 1955. This is in sharp contrast to agricultural productivity, where the steam tractor and the combine harvester slashed the proportion of GDP in agriculture from over 80% to under 5%.

This general lack of real productivity is clearly reflected in modern lifestyles. Hard working couples struggle to buy the basic food and shelter which their grandfathers had purchased while their wives stayed at home. If real productivity had risen just 1% pa, it would have almost doubled since 1955, which means that people could live a 1955 lifestyle working only three days per week. This is clearly not the case.

Analysis of Unproductiveness

To understand why we work harder than our grandparents despite the huge increase in productivity we have to examine the effect of productivity in detail. Consider the following data:-

	1955	2007
GDP	£6,670,000	\$11,400,000,000
Tax office Proportion of GDP	1.33%	1.23%
Pages in Tax act and Regulations	1,324	15,698
Raw Cost per Page	£5,030	\$726,000
Cost per kilo page per GDP	1.01%	0.08%

The key statistic appears in the third line of the table. The 1955 Australian taxation act and regulations was published in two volumes, consisting of 1,324 pages plus title pages and index. The 2007 act and regulations is over 12 dense volumes consisting of over 15,698 pages. We see in the following lines that there has in fact been a huge improvement in productivity as measured in relation to the size of the act and regulations that are being administrated. It would be quite infeasible to administer the current act with 1955 technology.

A good example of the benefit this extra regulation brings to society is the Superannuation Co-Contribution scheme which was introduced in 2001 to assist the poor to save for their retirement. It provides that the government will pay an additional \$1,500 into each person's superannuation account provided that they earn less than \$28,000, have a lazy \$1,000 to pay into their own superannuation account, are employed (ie. not house wives or students), and know about the scheme. Thus this marvelous scheme enables the government to be seen to be generously aiding the poor in a way that very few of the poor can take advantage of, and thus costing the government very little. A larger example is the new Goods and Services (VAT) tax, which is essentially the same as conventional income tax but is governed by a completely different set of regulations. Thousands of additions have been made to the act since 1955, and it is the duty of every Australian to comply with all of them.

Given that the size of a bureaucracy is not related to its function, one might ask why the size of the tax office has remained between 1% and 2% of GDP for over fifty years, regardless of the technology available to it. Why not 0.2%, or 35.7%? The answer is that society could not tolerate a value much higher than 2% — we would be paying taxes just to fund the tax collection process. Below 1% is easily affordable, so the bureaucracy will naturally grow beyond that size as predicted by Parkinson's law.

The Importance of Failure

We now see why it is so critical to society that software projects fail. The boundless creativity of politicians and bureaucrats to develop new and more complex regulation is bounded only by the

bureaucracy's inability to implement them. The absolute size of the bureaucracy is constrained by external factors, so the only effect of automation can be to increase bureaucratic complexity.

It would simply not have been feasible to implement the Superannuation Co-Contribution scheme in 1955, but automation had made it possible in 2001. Fortunately for society most tax office software projects have also failed, so the act and regulations have been limited to 15,698 pages. But imagine if all the projects had succeeded? We might need to deal with well over 150,000 pages of regulations, and society would be in danger of collapsing under their weight.

Again, one should not assume that these effects are limited to government. In 1955 banking products were very simple. Accounts were maintained, simple and transparent interest calculations were made, and few options were available. Accounts today are much more complex, with clever options to pay additional fees to have interest rates reduced under non-obvious conditions.

A major Australian bank recently undertook a software project to integrate a credit card loyalty scheme with home loan accounts. This promised customers huge savings provided they fitted exactly into very narrow and unlikely profiles. Fortunately the project failed due to the complexity of integrating disparate software systems, and so customers were never burdened with the need to understand, evaluate, and manage the scheme.

Another example was the development of a whole of government portal that the author was directly involved with. Its aim was to centralize and automate the issue and payment of government permits and fees, of which over two hundred had been identified. Fortunately the project was a complete failure, at a cost to the tax payer of a few tens of millions of dollars.

But what a disaster if it had succeeded! Once the two hundred existing permits had been automated hundreds more permits and regulations could have been easily created and implemented efficiently. In this age of "user pays" proposals had ranged from a hair dressing salon operating permit to a pedestrian permit to fund the construction of foot paths. Society was only spared this additional burden by the failure of the software project.

A naive reader might understand the damage caused by software systems, but might wonder if savings could be made by simply not attempting to build them in the first place rather than just having them fail. This of course completely misses the benefit that projects bring despite their failure to directly produce anything of value. Their development employs many bureaucrats, consultants and contractors and their expenditure supports an even larger number of people throughout society.

Conclusion

We have provided empirical proof of Berglas's Corollary, and clearly shown that software does not improve real productivity. Further, we have shown why it is essential that most software projects fail. No one need ever again be embarrassed by participation in a failed software project. Rather they should be proud to have spared society from yet another burden of complexity.

Some may misinterpret this article as satire. Surely it is not really desirable for software projects to fail. But the facts speak for themselves.

References

Parkinson's Law, The Economist, 1955. http://www.berglas.org/Articles/parkinsons_law.pdf

AUSTRALIAN ECONOMIC STATISTICS 1949–50 TO 1996–97
http://www.rba.gov.au/Statistics/op8_index.html

1301.0 - Year Book Australia, 1955
<http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/1301.01955?OpenDocument>

Do Computers Boost Productivity? Andrew Leonard.
<http://archive.salon.com/21st/feature/1998/04/24feature.html>

Australia's Strong Productivity Growth: Will it be Sustained?
http://www.rba.gov.au/PublicationsAndResearch/Bulletin/bu_feb01/bu_0201_3.pdf

Why Software Fails, Robert N. Charette. IEEE Spectrum article listing many failed projects and assuming that that is bad. <http://spectrum.ieee.org/sep05/1685>

Standish Group: Chaos. Suggests 31.1% of projects are abandoned, 52.7% are way over budget etc. <http://www.projectsmart.co.uk/docs/chaos-report.pdf>