Cutter IT Journal

The Journal of Information Technology Management

Vol. 26, No. 4
April 2013

"In Lean, value — not resource utilization — is the key driver of work."

— Jim Sutton, Guest Editor

Is Lean the Path to Releasing the Competitive Business Potential in Knowledge Work?

Opening Statement
by Jim Sutton 3

Lean Thinking and Knowledge Work
by Daniel T. Jones and Steve Bell 6

Dissolving a Dangerous Enthusiasm: Taking a Systems Approach to IT Systems
by John Seddon 11

Cooks in a Messy Kitchen
by David Snowden 18

The Lean Systems Framework: Extending Lean for Knowledge Work
by Frode L. Odegard 23

Implementing Organization-Wide Gemba Using Noninvasive Process Mining
by Saulius Astromskis, Andrea Janes, Alberto Sillitti, and Giancarlo Succi 32

Strategy Formulation for Lean Environments
by Kevin Brennan 40

NOT FOR DISTRIBUTION
For authorized use, contact
Cutter Consortium:
+1 781 648 8700
service@cutter.com

CUTTER CONSORTIUM

Cutter IT Journal

The Journal of Information Technology Management

Get global perspectives and solutions to some of the most critical business technology issues facing organizations today!

Your Front-Row Seat to IT Management Debate at the Highest Level!

Every day, your organization is confronted with the stark reality of having to achieve more aggressive goals with a shrinking budget, ever-changing requirements, and impossible deadlines.

Few of you have the time to develop well-supported arguments on how to get your organization to improve its IT operations. It's a tough trap: you know solutions are out there, but you're too busy to identify them and convince your organization to implement them.

Advice, Solutions, and Experience That You Can Rely On

A ***Cutter IT Journal*** subscription helps you break out of the trap. Every month, ***Cutter IT Journal*** features a select Guest Editor who articulates the controversial issues, offers his or her opinion on them, invites others to introduce opposing viewpoints, and sparks a lively debate.

Cutter IT Journal provides you the opportunity to experience a variety of perspectives: viewpoints that will be instrumental in advancing the cause of better software development. No matter where you stand on these issues, the thoughtful discourse delivered in ***Cutter IT Journal*** will certainly help you clarify your position.

In addition to your monthly journal issue, you will also receive its companion ***Cutter IT Advisor***. Each ***Advisor*** brings you practical advice and thoughtful analysis from well-known and respected experts in the IT field. Learn from the experiences of others, including what you should avoid and what you should consider implementing.

As a subscriber to ***Cutter IT Journal***, you'll stay up to date on important IT issues such as project management, security, risk management, business intelligence, sourcing, enterprise architecture, requirements, trends in technology, and more. Whatever the topic, you can be sure you'll receive frank, unbiased opinions, in the no-holds-barred manner ***Cutter IT Journal*** is known for.

Don't miss upcoming issues on:

- Creating High Performance Teams
- Disciplined Agile Delivery
- Is IT Still Relevant?
- Security and the Internet of Things
- The API Economy

Special offer for new subscribers!

Begin your subscription to ***Cutter IT Journal*** today and save $100 off the regular subscription rate! Plus receive all 2012 ***Cutter IT Journal*** issues on CD-ROM.

To subscribe for just $385 ($485 outside N. America) and receive your Free CD-ROM, go to **bookstore.cutter.com** and enter Coupon Code **CITJSAVECD** at checkout. Or complete and return the form below by fax +1 781 648 8707, phone +1 781 648 8700, or email service@cutter.com. For more on ***Cutter IT Journal***, please visit www.cutter.com/itjournal.html.

Visit bookstore.cutter.com to order individual ***Cutter IT Journal*** issues.

Special Offer: Save $100 on a New Subscription!

CUTTER CONSORTIUM

☐ ***YES!*** Please start my new, one-year subscription to ***Cutter IT Journal*** for just $385 (US $485 outside North America) — I save $100 off the regular price of $485/US $585! Plus send all 2012 issues on CD-ROM!

Name	Title
Company	Dept.
Address/P.O. Box	Mailstop/Suite
City	State/Province
ZIP/Postal Code	Country
Phone	
Fax	
Email	

Order online at bookstore.cutter.com (enter Priority Code CITJSAVECDto save $100). Or fax form to +1 781 648 8707, call +1 781 648 8700, or send email to service@cutter.com. Mail to Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA.

Dissolving a Dangerous Enthusiasm:
Taking a Systems Approach to IT Systems

by John Seddon

FOOL ME ONCE, SHAME ON YOU. FOOL ME TWICE …

In their aptly titled book *Dangerous Enthusiasms*,[1] Robin Gauld and Shaun Goldfinch summarize the shocking truth about large-scale IT projects: 30% fail completely and a further 60% fail in that they require significantly more time, resources, and effort than planned, and even then they often fail to meet requirements. Despite ritual invocations of IT's many project failures, the IT industry always insists that its latest reinvention "will work this time." What is also true is that we cannot run our organizations without IT systems, so this is a problem that we have to solve.

Prince 2 Has No Clothes

Prince 2, a method for managing projects that is particularly prevalent throughout the UK government, began life in the 1970s with the expressed goal of improving the delivery of large-scale IT systems through better project management. Despite the tsunami of continuing failure, Prince 2 advocates still believe better project management is the answer — that what is needed is to "do the wrong thing righter." As I shall argue, the complexity that draws us to project management as a solution must itself be removed for IT development to make a positive contribution to organizational performance.

Is Agile Doing the Wrong Thing Faster?

The Agile Manifesto[2] represented a bold departure from the conventional "waterfall" method for large-scale IT projects. Engaging more closely with the work and building solutions in iterations makes IT development happen live, and it brings development out of the long-term black box that was waterfall. But even when we are "agile," we are still using IT to solve the problems management believes it has (see below), which is why I argue that agile "does the wrong thing faster."

Agile does have many positive features: changes are implemented more quickly, iterative delivery avoids the perils of large-scale batching and queuing, and learning is both key and routine, improving what is delivered as well as future deliveries. Nevertheless, these features only become benefits *where agile is employed to solve the right problems*. So often, because the real problems are not understood, agile techniques are employed to speed up doing the wrong things. Where this is the case, agile's features are not benefits. Further, agile sits as a component in, and offers no challenge to, the wider IT development infrastructure. I shall return to this idea shortly.

WHAT PROBLEMS ARE WE TRYING TO SOLVE?

I work in service organizations. Managers of service organizations have, over the last 20 years, industrialized their operations. Call centers, back offices, and standardized and specialized work are the norm. Management of activity, control of processes, and reporting performance against measures like SLAs, standard times, and targets are the basis of management's thinking about the problems they have to solve, and thus they become the basis for specifications in IT development. Managers of these industrialized designs want their IT folk to supply workflow solutions, CRM, dashboard metrics, activity reports on individual workers, work-state reporting, scripts, access controls, and so on.

The Problem Is Industrialized Management

But here's the thing: managing service operations in this way is dysfunctional. The first clue is failure demand: *demand caused by the failure to do something or do something right for the customer*.[3] Failure demand is not apparent to managers of industrialized services because they treat all demand as work to be done. Codify that into IT systems for managing resource planning, and you're instantly blind to a major lever for improvement.

Failure demand is caused by the way the higher-level system (i.e., organization) that includes IT is designed:

- Front and back offices take different views of customers.

NOT FOR DISTRIBUTION • For authorized use, contact Cutter Consortium: +1 781 648 8700 • service@cutter.com

- The standardization of work fails to absorb the variety of customer demand.
- The specialization of work increases the number of handovers and thus duplication of work.
- Activity management — working to targets or standard times — fails to take account of variation and stifles agents' abilities to serve customers, and scripts and procedural controls have the same effect.

When we codify all that into IT systems, we turn a difficult problem into an intractable one.

The ultimate test — does this make service to customers better? — is not in anyone's frame.

On top of all that, management is distracted by problems that are apparent and seemingly pressing. Some are concerned with day-to-day bothers and issues that cause disruptions to service agents trying to do their jobs. Other managers are concerned with bigger projects, such as automating activities and introducing new applications or technology to solve problems they think they have. These problems and their associated activity become the foci for management attention. These larger projects are, inevitably, based on the industrialization logic I described above. We try to manage these problems by steering groups or what are sometimes called "investment" or "improvement" boards — and I put the labels in quotation marks for good reason.

IT projects get to the decision-making groups by firstly using business analysts, sometimes called architects, who listen to management's descriptions of their problems, agree on solutions, and come up with specifications and budgets. The decision-making group's task is then to allocate resources according to their view of the priorities; proposals for IT-based changes compete for budgets. Business plans, cost/benefit analyses, project plans, and milestones become the lingua franca for decision making. IT projects are prioritized, authorized, monitored, and reviewed against the annual planning cycle, which, in turn, is aligned to the annual budget/financial calendar. "Green light" status is achieved by conformance to plans, regardless of the project's impact on operations. But what if this development work is only compounding flawed thinking about how we should design and manage service operations?

Industrialization Has Spread to IT Development

It gets worse. In parallel with industrializing our service organizations, we have industrialized IT development. Few large companies maintain their own IT resources. Just as we have outsourced service work to low-cost providers, we have outsourced IT development. In many large companies, the only activity retained in the IT department is the analyst/architecture function. While the analysts will build specifications, the design, building, and testing of IT development projects are contracted out. Hundreds of people may remain in house, but many thousands of people work on IT development in outsourcing providers. And the only focus for providers is to deliver against the specification. Inhouse IT people now manage problems that didn't exist before outsourcing occurred: agreeing on specifications with suppliers, managing individual suppliers, and solving problems of how the various suppliers will work together. They become actors in a planning, monitoring, and reviewing activity against SLAs — they no longer do any value work.

In my business, when we try alongside our clients to get access to what's going on in outsourcing providers, and to study the "what and why" of the real performance being experienced, we find strong barriers. The attitude is, "Stay away. We've signed a contract, and we will deliver against that. How we work is no concern of yours."

Of all the outsourced activities, testing consumes the most resources and takes an incredible amount of time. If we reflect on the situation I described above, it is easy to see why. Service organizations operate in functional silos, all of which might be impacted by new IT development. Further, because development has been fragmented and outsourced to multiple providers, testing has to work across those boundaries too. While that will add to complexity and thus make testing a long and arduous process, we should also bear in mind that testing is based on features that represent the original analysts' views of the problems and requirements. The ultimate test — does this make service to customers better? — is not in anyone's frame.

Budget Is King

Finally, what we see when we study IT development is that the whole of this complicated world is dominated by an ethos of working to budgets and plans. Bonuses and other "performance" metrics depend on it, which means we deliver things we know aren't working and/or reduce specifications to meet either budget

NOT FOR DISTRIBUTION • For authorized use, contact Cutter Consortium: +1 781 648 8700 • service@cutter.com
©2013 Cutter Information LLC

or schedules. IT people in the UK describe this as "cut and push," the priority being to make the numbers at all costs. Development managers pride themselves on getting IT out on time and then being good at dealing with the problems that ensue; they become conditioned by their world, oblivious to reality.

MAKING TOAST THE AMERICAN WAY

The consequences of all this is massive failure. The most obvious sign is the creation of "war rooms" whose job it is to deal with postimplementation issues. As war rooms become overloaded, "control rooms" are established to manage the activity of the war rooms. We see a growing backlog of things to be fixed, and management's attention is entirely focused on solving those problems — no one is asking why we get it so wrong. It is as W. Edwards Deming[4] observed about American manufacturing in his time: there's lots of rework at the end of the line, often costing more than the original production. He joked, "Let's make toast the American way: I'll burn, you scrape."

It is all the foregoing that leads inevitably to the abundant failure evidenced by Gauld and Goldfinch. The IT industry has become industrialized and massively complicated. The big IT companies have a vested interest in its staying that way. The number of people involved in IT development is staggering, and the proportion involved in scraping burnt toast is alarming.

As Deming observed, improvement is all about method. There is a better way to run our organizations, and, similarly, we have to find a better way to develop IT.

DOES YOUR HELP DESK HELP?

If we go back to the original problem — that managers of service companies run their operations in suboptimal ways — IT developers can get an immediate understanding of this phenomenon by studying their organizations' IT help desks. When you do, you usually find that they don't help. It might be better to call them "allocation of number" desks, because that is what they do. Customers of help desks might understandably expect them to help, but what customers get is a job number. Of course, that is useful if you want to progress-chase your problem resolution (which is failure demand, and therefore produces no end benefit), but otherwise it is of no value.

The service in help desks is usually designed on the basis of "levels," with the most expert specialists at the higher levels. After all, managers think, you want to keep your most expensive resources away from the dross. What this does is increase the number of times a customer has to explain his or her problem. Often decisions about who has to pay become a gate, and getting through the gate — or, more correctly, convincing the service people that you have the right credentials — interferes with solving problems. Support personnel at all levels are incentivized to "close" jobs in order to comply with SLAs about the time taken to respond to customer issues. Well, you know what that does.

The obsession with industrial design principles such as maximizing resource loading must be seen for what it is: a plausible but wrong-headed ideology.

Studying the Help Desk

When you study IT help desks, you learn that the features of industrialized design are the causes of poor performance. Studying demand reveals high levels of failure demand, all passed into the system for action — activity of no value that is often institutionalized with CRM software. You learn that the true end-to-end time for problem resolution is enormous and dreadful. It is hardly a surprise that the system is full of failure demand. When you follow individual requests for service and see how the current design fails to do the value work — diagnosing and fixing the problem — and how the preoccupation with activity management and service levels causes that failure, you quickly come to the conclusion that the system has to change.

But you can also appreciate that you won't be able to change the system without rethinking your theory of management. Functional specialization, activity management, service levels, and the like have to go. The obsession with industrial design principles such as maximizing resource loading must be seen for what it is: a plausible but wrong-headed ideology.

What is vital about this revelation is that it is achieved through studying the system.

Designing Help Desks to Help

The knowledge gained though studying the help desk as a system can be used to design a better service. Thorough knowledge of demand means you can build

NOT FOR DISTRIBUTION • For authorized use, contact
Cutter Consortium: +1 781 648 8700 • service@cutter.com

the expertise required to solve all high-frequency, predictable demands quickly. When lower-frequency demands occur, help can be "pulled" to the front line. Measures should be derived from what matters to customers: one-stop resolution. End-to-end times and/or on-time-as-required are used for understanding and improving performance. These measures are used where the work is done.

The job of management changes from managing activity, work states, service levels, and the like, to acting on the system. That means things like ensuring the service has intelligence about demand and the capacity to respond to it. In systems-speak, this requires ensuring the system can absorb the variety of customer demand. When such organizational designs are implemented, service improves and costs fall out.[5] The help desk truly helps.

SOFTWARE DEVELOPMENT: DOING THE WRONG THING BETTER

The features of industrial design also affect software development organizations, whether internal or outsourced. Managers of software development functions are preoccupied with utilization, budget, and due dates. Studying software development projects as they go through this typical organizational design reveals massive amounts of duplication, rework, errors, delays, work that starts but then stops, and work done that is subsequently thrown away. Then there are "son-of" projects, where we didn't get it right the first time so we need to do it again, or work is required to tidy up the unforeseen problems that were caused by implementing the new IT.

Managers of software development think "the more you start, the more you will finish"; they try to do as much concurrent project delivery as possible. The practical consequences are lots of sharing of resources, multitasking of individuals, fragmented delivery of work on every project, and bags of resource coordination and management overhead. Managers couldn't be more wrong**:** the truth is the more you start, the *less* you will finish.

As with help desks, software development can be redesigned, focusing in the same way on demand, value work, and flow. But software development differs in important ways from help desks. A help desk is dealing with transactions; a software development group is developing a solution — it is a complex project environment.

One bright light in the software development world is David J. Anderson's *Kanban*[6] (despite the fact that his use of the label "kanban" doesn't mean the same thing as its originators meant). Anderson rightly asserts that inventory must be withdrawn from these project-stuffed systems. It is an essential prerequisite for learning how to manage flow. When the system becomes stable, it can be managed on the basis of one-in, one-out. While "utilization" may fall, focus on the value work rises, and the system produces better software while consuming fewer resources. But reducing inventory is only a tactic in improving software development; to achieve sustainable gains, managers need to understand how their focus on utilization, budgets, and due dates is central to the problems and not the solution.

However, to return to the broader thesis offered here, while we can improve software development, if it is developing solutions for people who have the wrong conception of their problems, improvement only amounts to doing the wrong thing better.

Is "Lean" the Answer?

Kanban, a sound idea, grew on the back of "Lean," a terrible idea in the way most Western consultants and tool vendors have taught it. In the West, Lean has often been promoted as the Toyota System in a box.[7] Taiichi Ohno, the man who developed the Toyota System, insisted it shouldn't have a name at all lest managers expect it to literally come in a box. Ohno was quite prescient, as this was the very marketing opportunity many early and recent Western Lean promulgators tapped into. They capitalized on the bias of most managers since the mid-20th century to achieve change through tools and projects. Ohno, in contrast, taught his managers to study the overall system, an orientation many Western "Lean" practitioners still ignore.

DOING BETTER THINGS

If our organizations are to do better things and IT development is going to contribute to that, we need to take a leaf out of Ohno's book. We need to *study*. The first step in IT development should never be a specification. In fact, if we think we have something that could be improved, the first order of things should be to study the problem, for so often we will find that the problem we thought we had was not, in fact, the problem.

Changing Our Understanding of the Problem

A good example of understanding the problem and only subsequently developing IT, one that is easily understood by people who have no knowledge of the service, is housing repairs.[8] As with many "break-fix" services, housing repair services in the UK social housing sector are conventionally designed according

NOT FOR DISTRIBUTION • For authorized use, contact Cutter Consortium: +1 781 648 8700 • service@cutter.com
©2013 Cutter Information LLC

to the industrial principles I discussed earlier. First, a call center receives a call from a person who needs a repair. Then the call center agent decides the relevant target and the work to be done, using a schedule of rates — a specification of the work to be done, the time it will take, and the materials that will be used. Jobs are sent to operatives electronically, and the operatives are measured for productivity.

When you study such systems, you learn that few repairs are completed on the first visit. You find that the call center operative will almost always fail to understand the true requirement, and you discover that productivity measures and bonuses damage rather than encourage improvement. In short, studying reveals management thinking to be the primary cause of poor performance — which is, of course, why managers should do the studying.

Studying also reveals the predictability of demand from properties and the predictability of materials required to repair things. Managers who have studied the failings of the conventional design realize that the only way they can absorb variety (a key feature of effective service design) is to start the flow of work only when the tradesperson is in front of the problem. These two things — understanding demand and providing the right expertise to service the variety of predictable demand — are the keys to managing value. Furthermore, managing value drives costs out of the system. The frequency with which problems are repaired on the first visit rockets to almost 100% and the costs of materials falls, because material flow is designed to be pulled by the rate of use in the properties. A different organizational design leads to the provision of a service on the day and at the time the tenant wants, and that amazing service is delivered at half the original cost.

IT DEVELOPMENT AS LAST, NOT FIRST

It is time to turn our thinking about IT development's role in change on its head. Instead of IT being the driver and necessary means for change, it should be relegated to last in the order of things. The first step is study, which changes management's perception of the problem(s). The second step is to improve the service design so that it works for customers. The third and final step is to pull in IT development to provide the means to run the service in the new way.

In the conventional repair service designs commonly used by social housing landlords in the UK, IT is used by call center agents to record targets and schedule-of-rates decisions, monitor tradespeople, control materials, and so on. These IT systems conventionally cost hundreds of thousands of pounds. In the unconventional but far more effective designs, IT is used by tradespeople to control their availability (they tell the system how long jobs will take) and to purchase materials at the rate of use. These better IT systems cost far less — in the tens of thousands — and are central to achieving better performance.

Only when the new design is stable and demonstrably better does attention turn to IT development.

Study – Improve – Pull IT

The first step, to study, is as far removed from business analysis as it is possible to be. It begins with no articulation of the problem; in fact, all current perceptions of problems must be put to one side. No plan, no specification, no requirements other than the seed question, "Can this be improved?" Studying reveals the causes of poor performance and the means for improvement.

The second step, to improve, also makes no demands on IT. Any IT system in use may be either treated as a constraint to be worked around or can be turned off. Often in the improvement step, it is necessary to create paper-based or desktop applications, with spreadsheets and the like, to aid the improved service design as it is being developed.

Only when the new design is stable and demonstrably better does attention turn to IT development. The questions now are "What can IT do to help this new design work even better?" and "How can we hard-wire this better design?'

A TRANSACTIONAL SERVICE ARCHETYPE

Studying and redesigning transactional services reveals a completely different archetype from the industrial designs we have built over recent years (see Figure 1).[9]

With knowledge of demands from customers, it is possible to train workers in the expertise required to handle the high-frequency, predictable value demands (things we know we are going to get a lot of) that are hitting the system. The consequences are shortened training times and more productive employment of the worker. When the worker receives a customer demand for which he or she is not trained, the required expertise is pulled as

NOT FOR DISTRIBUTION • For authorized use, contact Cutter Consortium: +1 781 648 8700 • service@cutter.com

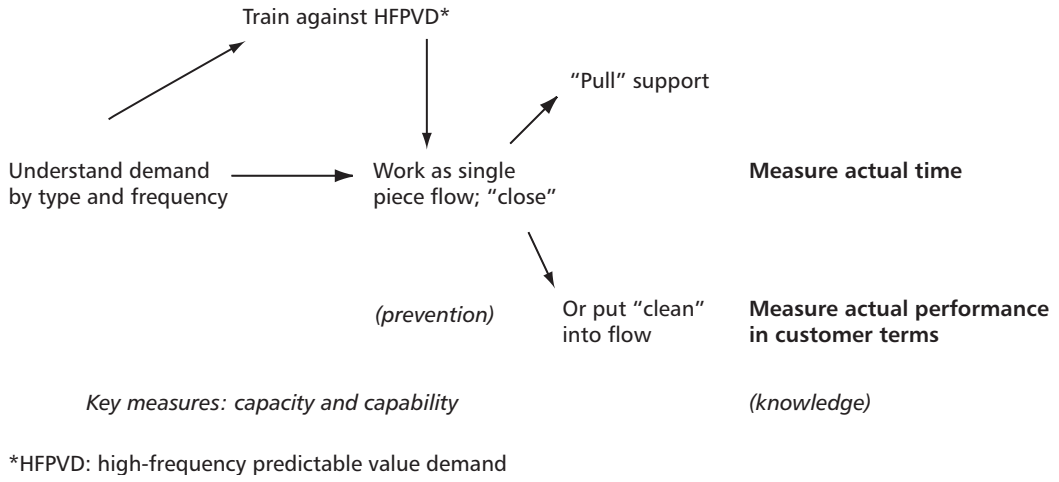


Figure 1 — The systems archetype for transactional service systems. (Source: Seddon, O'Donovan, and Zokaei.)

needed. In this way, worker training is directly related to the requirements of the work. The worker aims to achieve single-piece flow, dealing with each demand as it enters the system right through to resolution for the customer before beginning with another demand. If the work *does* have to be handed on to a flow, then the worker focuses on passing it "clean" — it must be in such a state that the next person has everything he or she needs to take the next step. Workers have in their hands measures that relate to the customer's purpose (one-stop capability, measures of end-to-end flow) and hence, like Ohno's workers, have the latitude to experiment with and improve the work design.

Training workers against demand and ensuring they are responsible for what they do is preventative (the better alternative to inspection). All arbitrary measures — standard times, cost, targets, and standards — are removed from the system, and instead real measures are used to help managers and workers alike understand and improve the work. It is better, for example, to know the actual time it takes to complete transactions in a one-stop manner; this improves resource planning. Similarly, it is better to know the true experience of the customer for any work that goes through a flow (end-to-end time or on-time-as-required) in order to improve the flow and, consequently, reduce costs.

At its heart, the systems archetype is concerned with designing according to customer demand, managing value rather than cost. And this is the heart of the paradox: when managers manage costs, costs go up; when they learn to manage value, costs fall. It is a counter-intuitive truth.

Because the archetype in Figure 1 is based on working outside-in, it removes complexity from the customer experience. In simple terms, it gives customers what they want. Contrast this with typical customer experiences in industrialized service designs, where, for instance, customers receive bills that reflect the fragmentation of service design and the associated fragmentation of IT systems. (For an example, just take a look at your telephone provider's bill.)

DISSOLVING THE DANGEROUS ENTHUSIASM

In service organizations, it is a relatively simple task to understand the customer experience in buying a service and, with the right IT skills, to develop a working solution — for example, a bill customers can understand and pay. When managers first see the elegant simplicity of the method, their immediate concern is, "How can this be scaled?" By that they mean, "How can we fit this solution into the IT infrastructure that we have?" And here's the rub: that's not the right question. It is not an IT problem; it is (or was) a business problem. It is a problem we have to dissolve, not solve. We have to dissolve the way marketing offers complexity to customers. We have to dissolve the functional industrial designs. In this way, we can dissolve the need for complex IT systems.

The transactional service archetype exposes the folly of the current approach to large-scale IT projects and obviates the need for the current complex outsourcing arrangements. The contrast between the IT resources needed in conventional IT-led change and the IT

NOT FOR DISTRIBUTION • For authorized use, contact
Cutter Consortium: +1 781 648 8700 • service@cutter.com
©2013 Cutter Information LLC

resources required to solve the real problem is nothing short of breathtaking. It's the route that IT development has to go, but we shouldn't expect the IT industry to take the lead.

ENDNOTES

[1]Gauld, Robin, and Shaun Goldfinch. *Dangerous Enthusiasms: E-Government, Computer Failure, and Information System Development*. Otago University Press, 2006.

[2]Agile Manifesto (http://agilemanifesto.org).

[3]Seddon, John. *Freedom from Command and Control*. Vanguard Press, 2003.

[4]Greisler, David S. "William Edwards Deming: The Man." *Journal of Management History*, Vol. 5, No. 8, 1999, pp. 434-453.

[5]Lane, Jo, and Phil Badley. "Stockport Metropolitan Borough Council: Implementing Systems Thinking in IT and HR." Chap. 3 in *Delivering Public Services that Work*, Vol. 1, edited by Peter Middleton. Triarchy Press, 2010.

[6]Anderson, David J. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.

[7]See "It's the Toyota System … in a Box!!" (video) at www.vanguard-method.com/content/380.

[8]More details on this example can be found in "Re-Thinking IT," my keynote presentation to the *Øredev 2010 Developer Conference*, Malmö, Sweden, 8-12 November 2010 (http://vimeo.com/19122939).

[9]Seddon, John, Brendan O'Donovan, and Keivan Zokaei. "Rethinking Lean Service." Warwick University, February 2011 (https://www.vanguard-method.com/v1_lib.php?current=448).

John Seddon has received numerous academic awards for his innovative contribution to management science. Inspired by W. Edwards Deming's observations that mankind invented management and conventional management doesn't work very well, Prof. Seddon pioneered a systems approach to designing and managing service organizations. He developed the "Vanguard Method," the means by which service organizations can change from a conventional command-and-control design to a systems design. Prof. Seddon is the founder of the Vanguard organizations, which operate today in nine countries. He can be reached at John.Seddon@vanguardconsult.co.uk; website: www.vanguard-method.com.

NOT FOR DISTRIBUTION • For authorized use, contact Cutter Consortium: +1 781 648 8700 • service@cutter.com

Cutter Consortium

Executive Education+

SUMMIT 2013

4–6 November 2013
Cambridge, MA, USA



Intense, Interactive Instruction. Effective Learning.

Register today and save!

Single Seat: $1995 for a limited time (save $500!)

Team Builder: Buy 1 seat for $2495 and bring a colleague at a deep discount. (Save 30%)

Executive education on IT leadership and emerging trends. A year's worth of professional development and personal enrichment in 3 invigorating days.

From an in-depth case study taught in the popular business school style, to interactive small group exercises and keynotes that help you identify business opportunities made possible by emerging technologies, you'll enjoy truly unbiased discussion and meaningful debate on today's IT opportunities and challenges at the Cutter *Summit*. Discover and learn about new strategies, technologies, and leadership skills — from Cutter's exceptional lineup of experts —that will help you embrace the ever-unfolding opportunities and challenges of the SMAC business environment.

In addition, you'll benefit from hands-on seminars and roundtables led by Cutter's Practice Directors and Senior Consultants on topics such as software engineering and agility, business and enterprise architecture, CIO/CTO issues, and data insight and social BI, to name a few.

You'll enjoy (and join in on!) raucous panel debates; networking at lunches, breaks, and entertaining evening events; and get one-on-one guidance and input from expert presenters and participants.

Monday, 4 November 2013



The Evolving Role of 21st-Century Technology Leaders

Keynote by Robert D. Scott

Cutter Fellow; Director of the Information Systems Executive Forum, Ross School of Business, University of Michigan



Lightning Talks

Short and to-the-point presentations around a single strategy, technique, or success story.



Big — and Fast — Data Analytics

Case Study with Vince Kellen

Cutter Fellow; Senior Vice Provost for Academic Planning, Analytics & Technologies, University of Kentucky



Evening Cocktail Party

Unwind and socialize with the speakers and your fellow attendees while enjoying some of Boston's tastiest regional specialties.

Learn more at www.cutter.com/summit.html

Executive Education+
SUMMIT 2013

4–6 November 2013
Cambridge, MA, USA

Cutter Consortium

Tuesday, 5 November 2013



Sustainable Growth: Achieve It with Highly Motivated Teams

Roundtable with Lynne Ellyn

Cutter Fellow



Agile in the API Economy

Keynote by Israel Gat

Fellow and Director, Cutter Agile Product & Project Management Practice



Digging for Gold in the Emerging Technology Pile of Hype

Keynote by Lou Mazzucchelli

Cutter Fellow



Putting Your Leadership Skills to the Test

Active Learning Exercises with Michael Roberto

Cutter Fellow; Trustee Professor of Management at Bryant University

Wednesday, 6 November 2013

Choose a Track ▸

BUSINESS TECHNOLOGY STRATEGIES TRACK



CIO/CTO Roundtable

Forum with Ron Blitstein

Fellow and Director, Cutter Business Technology Strategies Practice

AGILE TRACK



Agile, Software Engineering, and Product Development

Workshop

TBA

DATA INSIGHT & SOCIAL BI TRACK



Data Insight and Social BI

Workshop

TBA

BUSINESS & ENTERPRISE ARCHITECTURE TRACK



Business and Enterprise Architecture

Workshop

TBA



A Theory of Practice: Soft Decision-Making in the Context of a High-Pressure IT Organization

Keynote by Tom DeMarco

Cutter Fellow

See the full program at www.cutter.com/summit.html

Cutter
IT Journal

About Cutter Consortium

Cutter Consortium is a truly unique IT advisory firm, comprising a group of more than 100 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, metrics, and sourcing.

Cutter offers a different value proposition than other IT research firms: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts, experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats, including content via online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or sales@cutter.com.

The Cutter Business Technology Council

The Cutter Business Technology Council was established by Cutter Consortium to help spot emerging trends in IT, digital technology, and the marketplace. Its members are IT specialists whose ideas have become important building blocks of today's wide-band, digitally connected, global economy. This brain trust includes:

- Rob Austin
- Ron Blitstein
- Tom DeMarco
- Lynne Ellyn
- Israel Gat
- Vince Kellen
- Tim Lister
- Lou Mazzucchelli
- Ken Orr
- Robert D. Scott