Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

the way we see it

Agile iSAP Improves Project Outcomes with Richer Customer Collaboration

Applying industrialized SAP project management to Agile development enhances collaboration with customers, thereby cutting project timelines and wasted effort.

When Capgemini invented iSAP it brought Lean manufacturing principles to the practice of implementing SAP projects. In doing so it also created an ideal solution for a problem that has challenged SAP project teams for a long time — how to apply Agile on an industrial scale. It turns out that the same methods and tools iSAP uses for Lean are also well suited for Agile. For example, they are designed to make highly visible those activities that either add value or contribute waste and also those critical decisions that customers and project leaders still need to make. (In fact, one of iSAP's key practices is called "visual management.") Once identified, that information is a natural input to Agile teams looking for rapid and holistic feedback — holistic in that the feedback may come from anyone on the project, especially customers who have new requirements.

iSAP's inventors call this Design By Acception®— focusing on the relatively few difference makers that truly add value for the customer — which is a turn on the Lean phrase of "management by exception" — which is to only focus on (and eliminate) things that make waste. Design By Acception® also dovetails naturally with Agile, which is all about finding opportunities for adding value in short iterative bursts of development. In both Lean and Agile, value is what the customer says it is, so the only way to know for sure is by asking the customer. Agile assumes that what the customer wants will constantly change in light of new information (including the arrival of new project deliverables). So by highlighting the "waste makers" iSAP can drastically enhance the quality of collaboration with the SAP customer and thereby drastically streamline Agile.

Why this is important is for the same reason that incorporating Agile into SAP has been so challenging. By nature, SAP projects are highly structured (in other words, very un-Agile like). They were engineered that way so as to bring order to projects of huge size and with many moving parts. Work is done by interdependent teams, in discrete phases (blue printing vs. realization vs. testing, for example) often over several years, and on well-ordered process hierarchies (such as end-to-end process vs. major process vs. business process). Having each team operate in long highly compartmentalized phases (called a waterfall) makes sense if you are trying to coordinate all this activity without an easy way to quickly correlate outcomes across teams (including customer feedback and decisions in response to feedback).

But iSAP addresses that missing element. More than a philosophy, iSAP includes tools that allow consultants to easily mix and match standard reusable project components so as to quickly identify gaps where customer needs still must be met. That visibility is project-wide, enabling the effective cross collaboration in the shorter iterative sprints that Agile imagines — but which, more importantly, gets more value to the client faster.

Lean and Agile — Natural Partners

The idea that Lean and Agile are natural partners for large software development projects has been around for some time. In a January 2010 Forbes article, titled, **"Why Lean And Agile Go Together,"** Dan Woods, an IT research consultant, wrote that:

> "Large organizations that are using Agile are applying a value stream approach in which various development teams are organized in sequential and parallel streams of work so that at the end of each iteration, you get a new version of the product. Sometimes different rhythms of iterations occur, with some work taking place in faster increments followed by the integration of everything every fourth or fifth iteration."

The value stream is a core Lean principle. Its basic premise is that all work is designed to take raw materials and add value through some process in order to produce a future state the customer wants. This end-to-end journey from materials to future state is the value stream. Examples in SAP include order to cash, procure to pay, and maintain to settle — i.e., end-to-end journeys that start with an input, like an order, and end with a outcome, such as cash in the customer's bank account.

Note that the customer's value stream, which is the result of an SAP project, is different from the project's own value stream, which starts with customer requirements and ends with a finished SAP application (and a happy customer). Optimizing one of these streams naturally helps optimize the other.

In general, end-to-end value streams are the few (5-10) business processes that:

- The company must master in order to deliver its **value proposition**
- Account for a **significant portion** of the company's costs or revenues
- Traverse the entire **value chain**
- **Cross** functional, geographic, and company **boundaries**
- Integrate **upstream** (i.e., close to suppliers) and **downstream** (i.e., close to customers) activities
- Exist relatively **independently** of other core processes, when fully optimized
- Have **measurable outcomes** important in the marketplace

Lean says that the best outcome is achieved by optimizing — eliminating waste from — the value stream. (Optimizing the value stream is called **Kaizen.**) Waste can be things like defects, overproduction, and unnecessary effort. And the best way to optimize the value stream is to focus attention less on whatever process or processes are involved because the process is not what the customer wants. Focus on the future state instead — which may involve multiple processes, perhaps in different functions, different departments, or even different companies. It's possible to have an "optimal" process within a less-than-optimal value stream, maybe because it doesn't integrate well with other processes. The end state might then be a system that doesn't perform the way the customer wants, a state that then leads to product rework, cost overruns, and so forth.

To help prevent that from happening, iSAP teams use a tool during project planning workshops called SIPOC (suppliers, inputs, process, outcomes and customers). SIPOC focuses consultants' attention on the future state's required inputs and outputs, rather than on the "how" of any particular value stream component. Yet the mere fact such tools exist suggests that the "how" actually does matter with regard to the best way to be Lean. And SIPOC — essentially a simple fill-in-the-blank template used in white boarding Lean SAP projects — is just one example. A much bigger example is Agile.

Notice, for example, that when Woods says organizations are taking a value stream approach, they're doing so in a very Agile-like way — as in "faster increments followed by the integration of everything in every fourth or fifth iteration." That's how Agile teams operate. The **12 principles of Agile** outlined in the Agile Manifesto, are all about making software in short increments, or sprints, each of which produces a working deliverable for which the customer provides feedback that is baked into the next sprint. Multiple sprints covering different features and functions can also be going on in parallel and their respective deliverables integrated later. That means that each sprint is a complete mini-project in itself, covering all project phases, such as specification, programming, and testing. That's opposed to the Waterfall model where all deliverables are specified, programmed, integrated, and tested as a complete unit prior to customer review — the assumption being that the customer's requirements have not changed and that the project team interpreted them perfectly before work started. Of course that rarely happens — which means there can be a lot of wasted time and effort with Waterfall — the kind of waste that Lean also seeks to eliminate.

So, yes, the fact that Agile and Lean might go together seems intuitive. But it would also be wrong to say that by employing one of these models you automatically employ the other. In fact — and we will see this more specifically with SAP — Lean development can pose challenges that actually make it harder to use an Agile approach. And teams therefore need to take formal measures — which iSAP does — to make sure Lean development is Agile.

Agile, like Lean, focuses on customer value as perceived by the customer, as the ultimate success metric. The difference is that, strictly speaking, Agile only looks to optimize the product itself and only through ongoing customer feedback as the product takes shape. Lean looks to optimize the value stream (and only by extension the product) by keeping inputs and outcomes in view. It makes sense, then, that when organizations follow Agile development they would want to do so in a Lean way, and also that Lean organizations would want to develop software in an Agile way. Otherwise the result might be neither Agile nor Lean. But an Agile Lean approach is not automatic. A Lean team could in fact use a Waterfall approach and simply wait until a value stream was complete before showing results to the customer. Likewise, an Agile team could focus on getting a particular function "right" in the eyes of the customer and then worry later about how a particular implementation affected other functions involved in the value stream.

But the incentives to unite these two models are clear. Not only do you get the benefits of both, but also Agile is more Agile (i.e., the results better fit the requirements) and Lean is Leaner (i.e., the results get the customer to value faster). When that happens, Lean methods don't just optimize the customer's value stream — such as by turning orders into cash better — but also optimize the project's value stream — i.e., by turning customer requirements into finished code better.

How SAP Challenges Agile Lean

Nor have these benefits been lost on SAP. As Woods notes, even by 2010:

> “SAP [had] been using SCRUM and other Agile methodologies for several years … expanding application of Agile methods to the entire product creation process using a Lean framework that includes empowered cross-functional teams, continuous improvement process and managers as support and teachers.”

But until iSAP, an SAP implementation was still more about configuring SAP modules — even if sometimes in a more Agile way — than about creating value streams. Configuring SAP modules was what consultants were trained to do and what SAP tools were designed to do. Even if you empower cross-functional teams and strive for “continuous improvement” with “managers as support and teachers” you still have to overcome resistance from an outdated development premise and the tools designed to support it.

Dating back to SAP’s very beginning, that premise was that you can leverage partially built solutions composed of reusable building blocks to accomplish two key objectives:

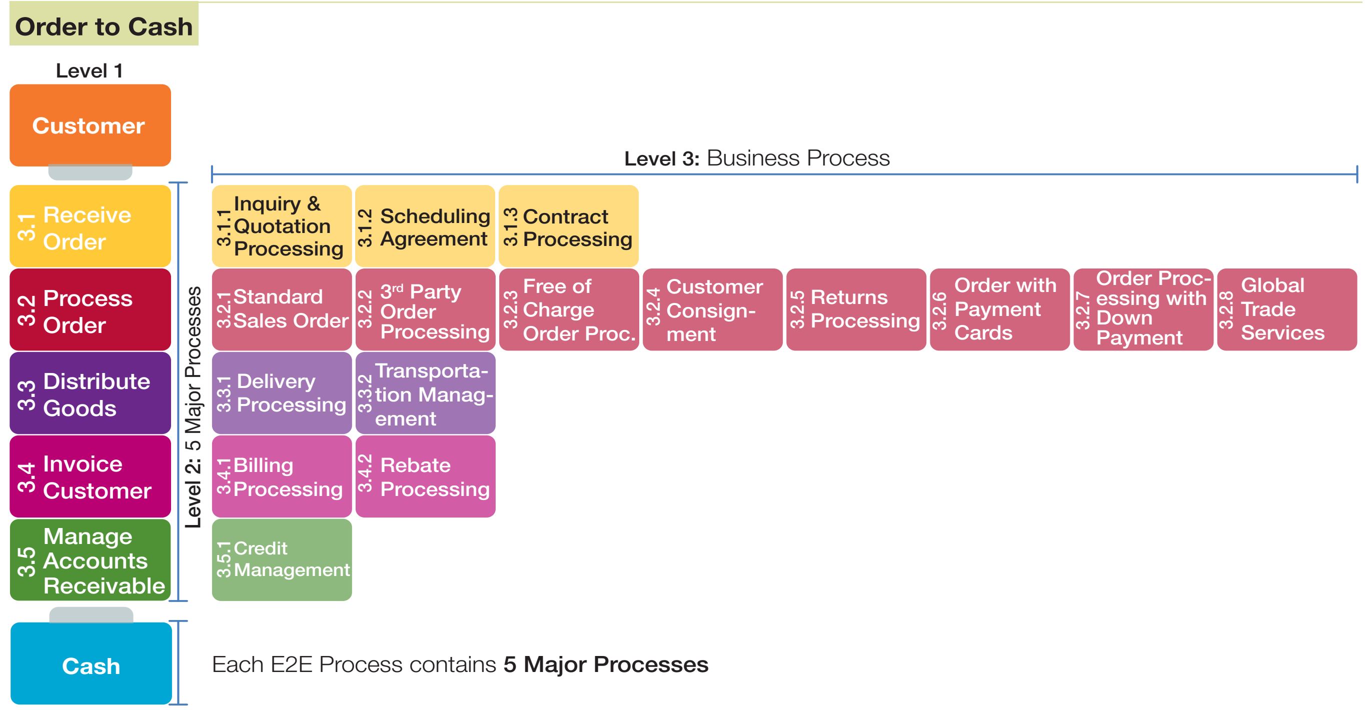
1. Let organizations get functions up and running faster and for less cost than if built from scratch; and
2. Enable easier interactions between different functions across the organization since functions were derived from the same building blocks.

Those were tremendous go-to-market advantages at a time when the alternative was either from-scratch development or off-the-shelf solutions that were not tailored to the company (if they existed at all) and couldn’t communicate with each other.

Now, of course, what were once advantages are now just table stakes and the skills to configure SAP modules are commodities. Furthermore, most organizations looking for an SAP implementation end up requesting a vanilla solution — in other words, one that resembles what other organizations implement, which they perceive to be the safest, fastest, and least costly option. The downside is that they also may sacrifice the very things a more value chain oriented approach offers — like their value proposition.

But certainly Agile and Lean proponents are not abandoning SAP modules, or the skills needed to configure them. Reusing existing content, rather than reinventing it, is a key enabler of both Agile and Lean. Reuse is more Agile because there’s less code to implement, test, and integrate. That means sprints can be shorter, more frequent, and each one can represent a bigger swath of the finished product — so customers have a better idea of what the final product will look like, and so the end state fits requirements better. Reuse

Business Process Hierarchies - Order To Cash Scope



Order to Cash

Level 1

Customer

Level 3: Business Process

Level 2: 5 Major Processes

3.1 Receive Order
- 3.1.1 Inquiry & Quotation Processing
- 3.1.2 Scheduling Agreement
- 3.1.3 Contract Processing

3.2 Process Order
- 3.2.1 Standard Sales Order
- 3.2.2 3rd Party Order Processing
- 3.2.3 Free of Charge Order Proc.
- 3.2.4 Customer Consign-ment
- 3.2.5 Returns Processing
- 3.2.6 Order with Payment Cards
- 3.2.7 Order Proc-essing with Down Payment
- 3.2.8 Global Trade Services

3.3 Distribute Goods
- 3.3.1 Delivery Processing
- 3.3.2 Transporta-tion Manag-ement

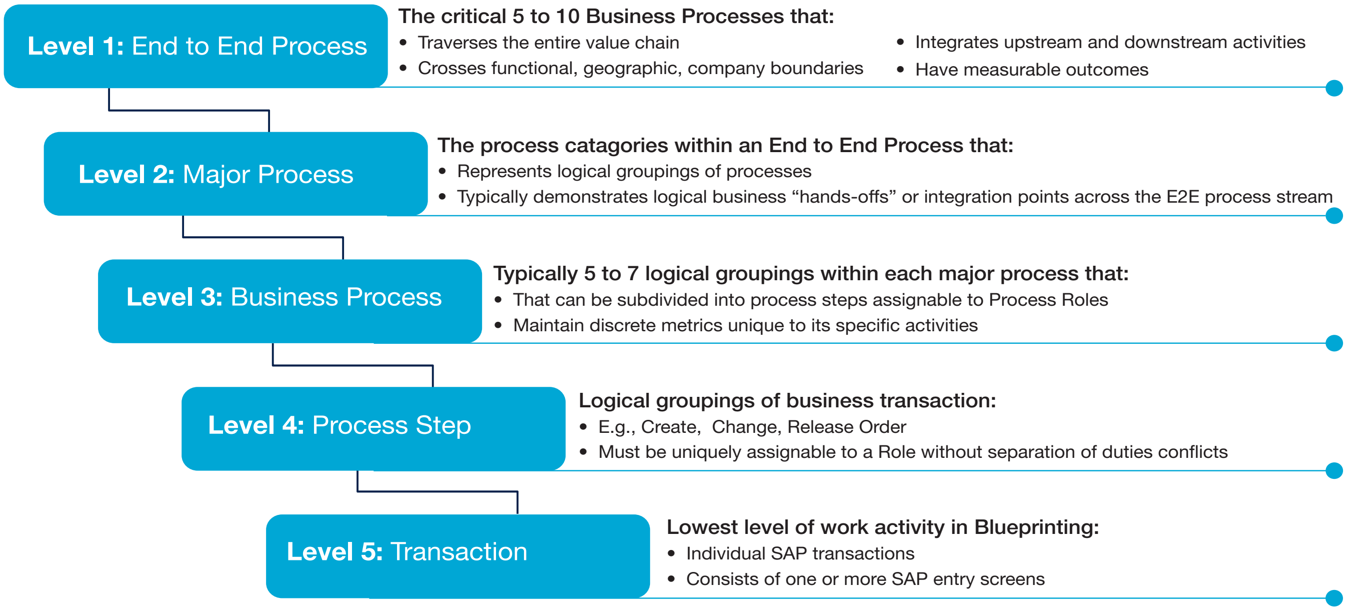
3.4 Invoice Customer
- 3.4.1 Billing Processing
- 3.4.2 Rebate Processing

3.5 Manage Accounts Receivable
- 3.5.1 Credit Management

Cash

Each E2E Process contains **5 Major Processes**

Diagram 1



Level 1: End to End Process

The critical 5 to 10 Business Processes that:

- Traverses the entire value chain
- Crosses functional, geographic, company boundaries
- Integrates upstream and downstream activities
- Have measurable outcomes

Level 2: Major Process

The process catagories within an End to End Process that:

- Represents logical groupings of processes
- Typically demonstrates logical business “hands-offs” or integration points across the E2E process stream

Level 3: Business Process

Typically 5 to 7 logical groupings within each major process that:

- That can be subdivided into process steps assignable to Process Roles
- Maintain discrete metrics unique to its specific activities

Level 4: Process Step

Logical groupings of business transaction:

- E.g., Create, Change, Release Order
- Must be uniquely assignable to a Role without separation of duties conflicts

Level 5: Transaction

Lowest level of work activity in Blueprinting:

- Individual SAP transactions
- Consists of one or more SAP entry screens

Diagram 2

is also Leaner because ERP software out-of-the-box already represents years, if not decades, of optimization before any further optimization takes place. Secondly, it allows consultants to focus on the relatively small list of the optimizations (i.e., Design by Acception®) that differentiate this particular value chain for this particular client in this particular market. That defines end state success, not by internal development metrics — e.g., the number of SAP modules that are correctly configured and working — but by the entire value stream’s measurable impact out in the world.

So the SAP challenge is how to turn a collection of modules into value streams by leveraging content reuse without making every project a “module configuration project.” Classic SAP, as noted, organizes projects by functional unit (e.g., entry screens, transactions, release orders, etc.) consisting of modules and sub-modules. An example is FI (financial accounting), which can consist of one or more sub-modules, such as FI-GL (general ledger), FI-AP (accounts payable), FI-AR (accounts receivable), and so forth. By comparison, here is what a value stream would look like, taking order to cash as an example in Diagram 1.

In this view the desired “end state” is not the desired state of the financial accounting system (even if that is where many of these functions live) but the customer’s desired ability to generate cash from orders, which presumably is what the customer really cares about — and to do that in the best way possible. With the value stream clearly mapped, each function can be evaluated based on how well it performs its role in the stream (and so can each sub-function within a function, and each transaction within a sub-function). That way, stream outliers can be more quickly identified and optimized.

The key insight here is that you can’t optimize what you can’t see. So iSAP’s inventors have mapped several of these value streams. In addition to order to cash, others include finance to manage, procure to pay, demand to supply, and hire to retire. In each one, the goal is to reveal any opportunities for optimization in the Leanest way possible — in other words, with the least amount of work or at the lowest level of granularity in the value stream hierarchy. That means you need to see the value stream hierarchy, which is why iSAP maps value streams in the following general form:

In (Diagram 2) this hierarchy, Level 1 refers to the entire value stream and is implemented by all the layers below it. Level 2 are all the major processes comprising the value stream, consisting of Layers 3 process groupings and their respective Level 4 sub-processes (transaction groupings), which then obviously consist of the individual transactions.

Not only does this hierarchy promote Lean, it also promotes Agile. As already noted, because optimizations can be addressed at the lowest level of detail, they can require the least amount of work with the greatest amount of reuse. Sprints can then be aligned more precisely with the actual work that needs to be done in order to present the cus¬tomer with a complete functional unit. That invites the question: Yes, they can, but will they? The hierarchy above maps the customer’s value stream; but what about the iSAP project value stream? How Agile is that? Here’s what the iSAP value stream looks like:

The Phases and Streams of the iSAP Method

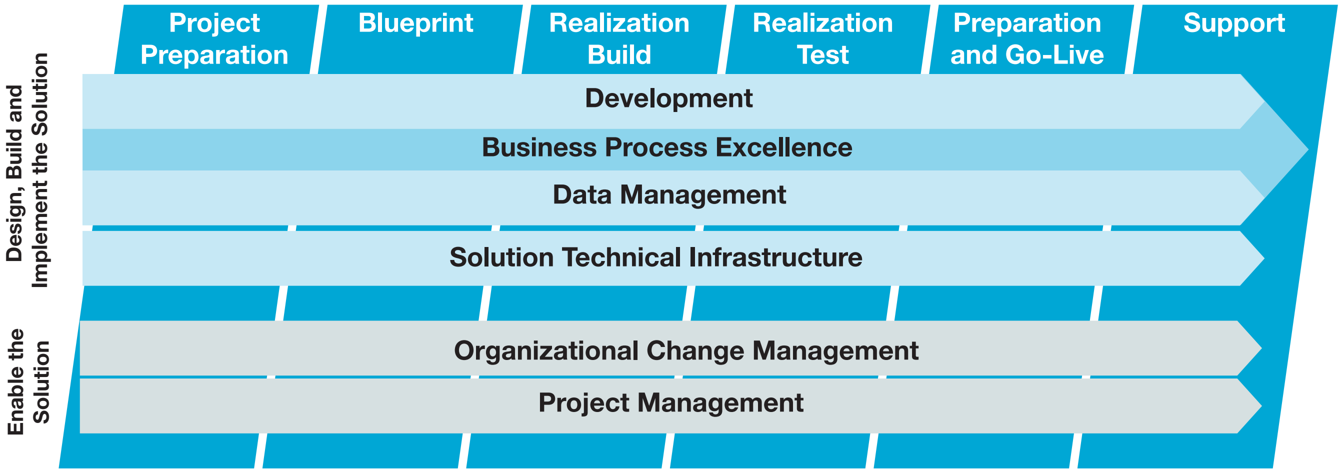


Diagram 3

It **looks** more Waterfall than Agile, as in: first we prepare (gather requirements, etc.) then we blueprint, then we implement, then we test, then we go live, and then we provide ongoing support. So, does that mean we go through each phase in lockstep for the whole app and only then show the finished app to the customer? What happened to the sprints along the way?

The apparent disconnect is resolved by recalling that a value stream, by definition, traverses the entire value chain (thus making Lean look Waterfall). The stream connects inputs to outputs and shows all phases of value in between. How you get from input to output is not what counts — whether by Agile, Waterfall, or some other way — as long as waste is removed. What's Agile about this value stream is that there are actually six "lanes" running in parallel. So, in every "phase" of value there will in fact be six lanes of activity, from project management to development. So, there is, for example, some development involved with testing (and vice versa); some data management involved with blueprinting and testing; and so on.

Let's say a sprint assigned to the order to cash value stream is intended to optimize the grouping of transactions to create, change, and release an order. Depending on the scope of work required, a sprint could touch any iSAP value chain phase. There could be new requirements; blueprints might change, gaps in standard SAP code might be filled, the results tested, and so on. All these activities would be tied together as a single unit of effort. The customer could then review the result to see if end state requirements were indeed met or if the requirements themselves need to change, setting the stage for even more finely tuned improvements, if needed, in the next sprint.

That's how iSAP combines Lean with Agile at a conceptual level, but how well it actually happens "on the ground" depends greatly on the tools involved.

iSAP Tools Invoke Agile Lean

iSAP uses a number of tools to invoke (rather than simply promote) Agile Lean in sync with customer and consultant value streams. The tools serve four key roles:

- **Provide a base of reusable content** across all six lanes of the consulting engagement that includes project plans, value stream maps, SIPOC lists, SAP configuration settings, unit test scripts, and more
- **Design by Acception**®, which is to expose and leverage opportunities for end state optimizations:
  - At any level in the customers value stream hierarchy,
  - At any stage of the project value chain, and
  - Diagram 3 shows what the iSAP value stream looks like
- **Enhance collaboration** among team members (both consultant and client side) around key decisions, work accomplished each week, activities planned for next week, and so on

- **Sync planning to software** such as by auto-generating updated functional design documents, configuration documents, and unit test documents along with their respective SAP settings and test scripts.

One tool that unites all four roles is the Error Proofing Tool. The EPT manages the repository of all value stream content and provides a portal where the content can be reviewed, updated, and deleted. The tool also shows status metrics (like percent completions) for the various deliverables defined in the planning workshops (called value stream workshops) as well as the status of the workshops themselves (like which open questions are left to be decided).

For each Level 3 process specifically, the EPT also provides:

- A narrative describing what the process does and how the customer uses it
- Business objectives of the process
- The SIPOC list for the process
- The process steps descriptions (e.g. the transactions involved)
- User roles and authorizations
- Key decisions, unit tests, and how those sync
- Standard reports the process generates, if any
- PRICEFW — the portals, reports, inputs, conversions, enhancements, forms, and workflows that result from a successful end state of this process
- KPIs — the key indicators that show the process performs as intended
- Compliance — how this future state SAP process impacts compliance policies
- Organizational change — the changes in the organization, processes, and roles needed to support this future-state SAP-enabled process

Consultants use the EPT as a platform for engagement with each other and with the client. The EPT is prepopulated "out of the box" for each value stream, so it's a given consultants will in fact build value streams (a much richer workshop topic than "module configuration"). The EPT also leaves far less "vanilla" SAP to talk about — because it's already "done." What items are left then to discuss? Mostly those are the real difference-makers that comprise relatively tiny pieces of the end-to-end value stream. And because these pieces are both difference-makers and small they lend themselves well to Agile. They can be accomplished in a week or two and they are something of value about which the client will have meaningful feedback.

Richer Collaboration Breeds Transformation

Getting the SAP customer involved early and often is also the key difference-maker when it comes to a project's value stream — whose desired end state is a transformed customer business. The customer has to pull the value; consultants can't push it. That's a key tenant of Lean, and an Agile approach enables it. Otherwise the customer is left waiting, possibly months, to see if the consultants "got it right," with all the potential waste that implies.

In other words, a transformed customer business requires a transformed SAP practice:

It's the type of industrialized Lean practice that many SAP experts have wanted. But when it comes to an SAP practice, invoking Lean methods consistently in an industrialized manner requires more than only the Lean methods you might find in, say, an auto plant. It also requires Agile — because software projects require customer collaboration throughout the Lean value stream — and that is what Agile provides, and what iSAP enables

From this . . .	To this . . .
Requirements gathering and fit-gap analysis	Design by Acception®
Emphasis on SAP configuration skills	Emphasis on industry expertise and leading practices
Replication of the customer's current business processes in SAP	Prescriptive transformation to improve process efficiency and remove waste
Implementation of "vanilla" SAP	Implementation of difference-makers
Customer as passive value receiver	Customer as active value contributor
Lots of non-value added activity and missed deadlines	First-time-right delivery over shorter, more respected, project timelines

For more information, please contact:

Herbert Goertz
SAP Engagement Executive
herbert.goertz@capgemini.com
+1 561-303-9188

Bart Neal
Technology Consulting Services
bart.neal@capgemini.com
+1 214-395-9134

About Capgemini

Now with 180,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.573 billion (about $14 billion USD at 2014 average rate).

Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at
www.capgemini.com