

SAP Consultants Must Tell Stories Too

When SAP hired a chief storyteller it was signaling that one of the best ways to sell software is to tell stories. It's also one of the best ways to implement the software after it's sold.



A value stream (e.g., “order-to-cash”) does in software what a story does in words - links the problem to a desired outcome by way of a series of transformations in the middle.

If you ask Julie Roehm, SAP Senior Vice President and Chief Storyteller, about her role at SAP she might answer with a story — like the one about how she was hired. As she told Finsider, back in 2011 she was reading an article in USA Today where SAP CEO Bill McDermott described what SAP does: “Well, we are behind some of your favorite brands like iTunes, Disney, Nike and Sony.” Having been in touch with Bill over several years, Julie sent off a quick email suggesting a better description might have been to talk about the experiences that make customers happy because of SAP. In other words, tell the customer’s customer story. The job offer quickly followed.¹

Significant research has revealed why stories engage better than facts but the reasons are probably obvious to most people since most people are avid consumers of stories. People see themselves in the story; stories get people involved emotionally (not just rationally, which is far less compelling); and stories can be much easier to understand than software.

But if the SAP customer is buying a story, what should SAP consultants do? Shouldn’t they implement the story that moved the customer to buy the software? How then do they know the customer’s version of the story? And what happens if they implement the software before they know that version? The answer is: probably a disaster.

Customers may not be able to tell the story that’s in their heads. And even if they can, they probably can’t translate the story into specific features or functions. This means the consultant must tell the story for them — so that customers can hear it, affirm it, and revise it — before, during, and after implementation. If consultants only expect customers to tell them what features and functions to implement, it’s not going to happen.

That’s why storytelling is a central part of Agile. With Agile, it’s a given that customers must hear in their everyday language what the software will do for them — and that the consultant may not tell it right the first time. So the storytelling is more of a conversation than a presentation and avoids the trial and error of implementing the software first.

It’s also why iSAP follows a Lean approach to Agile, so software is implemented as a value stream or customer journey — analogous to a hero’s journey in a story. A value stream (e.g., “order-to-cash”) does in software what a story does in words — links the problem to a desired outcome by way of a series of transformations in the middle. Whatever “facts” are involved (features, functions, modules) is secondary. What matters is that customer and consultant are both engaged in the same story as it flows through both value streams — the customer journey and the implementation journey.

Functional Specs Are Insufficient

At the end of the day, of course, the story is not what counts. What counts is that customers receive the same great experience after they purchased that they imagined before they purchased. And much of that experience is emotional. A good example is car buying. When buying a car it is fairly easy for consumers to translate product features like speed and styling into an emotion on their own — a process well understood in auto showrooms. So although car salespeople may not tell a story like “imagine how fast you can go” or “think what your neighbors will say when you pull into the driveway,” they know they’re selling on emotion — even if they don’t know the actual story in the customer’s mind that triggers it. (However, a really good car salesperson will know how to ask.) The story could be as simple as no longer having to ask your parents for the car keys and the emotion could be the feeling of freedom that flows from that.

But in B2B, enterprise software customers may not make the translation automatically — hence, the need to help them imagine the story. The story makes the case rationally — as in here are the functions and features that will enable greater profitability, faster time-to-market, etc. But a story goes further. By making the connection in the buyer’s particular business context the story produces the emotional experience of how great life will be after purchase. This means instead of a rational abstraction, the customer is buying something they can actually feel right now — and it feels good.

This is why SAP customers can get so emotional when they don’t get what they want from SAP consultants. That good feeling of anticipation they’ve been experiencing abruptly ends. Worse, it is replaced by disappointment and frustration. The consequence can be withdrawal — not just in a physiological sense — but also in the business relationship, where both sides, customer and consultant, pull apart as evidenced by finger pointing, reduced communication, and lack of trust.

That’s another difference between B2B and B2C. Not only does the B2B customer need help imagining the story in the first place, they obviously need even more help turning the story into reality. It’s not like they can just drive away from a showroom and be happy.

What makes the implementation even more challenging is that it’s not just one “customer” whose story the consultant must realize. Nor is just one consultant involved. There are teams of people on both sides. B2B software sales are a team effort with different individuals on the buy side engaging with different individuals on the sell side. The same is true during implementation. This means IT will have one version of the story, business unit stakeholders another, C-level executives another, and so on. For the consultants, there will be project leads, functional specialists, technical specialists, developers, user interface specialists, and others — each with his or her own view of what the various individuals on the customer side want— all based on a functional design spec. It’s no surprise then that the functional spec can never be big enough or granular enough to speak to every customer stakeholder.

There are two reasons for this:

The first is the sheer scope of what needs to be communicated. Not only are requirements likely to change over time, and likely to change once customers see what consultants actually deliver; but requirements also change because different stakeholders have different individual priorities — some of which don’t surface until the stakeholder sees working code. That’s why practitioners of Agile development break up big projects into sprints where consultants deliver small pieces of working code about every two weeks for customer feedback. Consultants are relying on the back-and-forth process of writing small bursts of code and getting back small bursts of feedback to incrementally construct each stakeholder’s vision of the desired outcome — their story. That’s more efficient than Waterfall development, which doesn’t call for customer feedback during development — virtually guaranteeing that large portions will need rework. But what would be even more efficient (and less frustrating on both sides) is if consultants knew the customer’s story before the sprint starts so they don’t have to do so much back-and-forth.

The second reason functional specs don’t speak to every customer stakeholder is because functional specs are not very engaging documents. Functional specs by definition are descriptions of features and functions. But if features and functions were effective tools of engagement then SAP would not have to invest in storytelling as its strategic marketing tool of choice. When a customer looks at a spec, any of three possible things may occur:





- The customer may recall the story that moved the organization to buy the software in the first place.
- They may read into those features and functions their own mental image of what life will be like when that vision is realized in working software.
- They may look at the features and functions and not imagine any story at all. Maybe for whatever reason a particular stakeholder cannot connect the dots between the software and its impact for them going forward — perhaps because they were not involved in the purchase discussion.

None of these scenarios helps the implementation. Consultants are not mind readers. If their only option is to implement the agreed-to spec then their only way to deliver the customer's vision of life after implementation is through incremental trial and error — in other words, through a (hopefully rapid) series of progressively smaller disappointments.

Engage Stakeholders with Stories

The other option is to follow marketing's example — engage stakeholders with stories. Why stories rather than just specs? The difference between a spec and a story is that a spec only lists software facts — features and functions. A story describes “what happens next” because of the software in terms of its impact on people. Both a spec and a story describe a present and future state — in SAP terms, the *as-is* and *to-be* states. But the story talks in terms of what the person or organization can do in the future that they cannot do now and why it matters.

Suppose, for example, a large SAP retail industry customer is looking to consolidate within SAP HANA multiple operational databases that are currently siloed in various divisions and trading partners around the world. Here are the facts: Because of SAP HANA the company will soon have a multidimensional view of the company's global supply chain in real time. Here is the story: A regional manager in Chicago will be able to identify manufacturing glitches in Asia that will impact her ability to stock store shelves in Ohio. In fact, if she had this technology last year she could not only have spotted those glitches, but she could have done so in time — because of SAP HANA's vast in-memory big data processing capabilities — to save a really slow holiday shopping season. Needless to say, given that story, the regional manager in Chicago is very excited — a lot more excited than she might be about “consolidated operational data” or “processing large in-memory datasets.”

Imagine, too, what her reaction would be if — now that she believes in the story — it turns out later not to be true even after the SAP HANA implementation is fully done to spec! How can that happen? Here's how:

- Certain item-level data from RFID tags needed to be included in SAP HANA
- The data had to be collected, formatted and staged in a specific way by supply chain manufacturers
- The consultant never mentioned this to the customer
- So the customer in turn never communicated this to manufacturers
- Hence, the data (although visible in SAP HANA) was not converted to a form that allowed for the rapid response the regional manager expected to keep on top of product demand
- No one had thought to ask for the code customization needed to make data — technically “visible” in SAP HANA and also “useful” in the business context

Here's why then it's important for SAP consultants to tell stories:

Most stakeholders understand stories better than they do software. Mishaps like this one often occur because people on both sides don't know what they don't know. Consultants don't know what questions to ask. And business stakeholders don't know what information to provide or what other work they need to do (like asking internal departments and supply chain partners to reformat data). No one could expect a business stakeholder, like our mythical Chicago regional manager, to know technical details about data formats. Nor might an SAP consultant be expected to know about the day-to-day challenges confronting people in particular roles in a specific company in a specific industry. A story is a common channel that bridges that gap in perspectives. It forces technical issues to the surface by asking a simple question: "What needs to happen technically so that my business story will come true?"

Stakeholders see themselves in the story. As noted earlier, a problem with functional specs is that people often don't find them that interesting and so don't pay attention to them. But, of course, in order to communicate people must pay attention to each other. Nothing grabs your attention as much as a story that involves you. Placing stakeholders — their roles, their wants, their challenges — at the center of the story provides that stakeholders stay focused so there is less chance of forgetting important details.

Stories offer emotional attachment. Paying attention is one thing. "Owning" the story goes a step further. If stakeholders become emotionally attached, they become not just participants but partners and even project champions. Research has widely confirmed that stories evoke emotions. In fact, people seek out stories precisely for their emotional impact, which is one reason we go to the movies instead of just reading plot summaries. Research also shows that people are more likely to respond to emotions over facts. They'll even "explain away" facts that contradict storylines or narratives to which they are already emotionally attached.

In other words, stories fulfill the requirements of an effective communications channel — one that would not otherwise exist between stakeholders and consultants. People share common reference points and are engaged in what the other person is saying. Consultants can leverage that communications channel to speed implementation and reduce project risk. Take the SAP HANA scenario described above. Given the Chicago regional manager's story, several questions naturally arise that might not have been asked with only a spec to go by, such as

- What key performance indicators will users want to see on their dashboards?
- Will any datasets, therefore, need to be reformatted prior to SAP HANA ingest?
- What data will business units and trading partners want to share?
- What steps (both organizationally and technically) are required?

Questions like this break vicious blame cycles such as when the consultant says, "Why didn't you provide <resource>?" and the client says, "Why didn't you tell us <resource> was necessary?" and the consultant says, "Why didn't you tell us <action> was a requirement?" and the client says, "Why didn't you ask?"

A story lets the consultant know what questions to ask, whether those questions are about a particular end-to-end process (a "value stream," to use the Lean term) or at a more granular level such as business process within the value stream, process step, or a single transaction needed to complete a process step. Every time consultants implement software they are either helping make a stakeholder's story come true or not. Having that story in hand, before implementation

starts, provides that the right questions do get asked and the stakeholder's story does come true. That's why storytelling plays such a key role in Agile development.

How Agile Leverages Stories

Storytelling in Agile is a two-way process in which a consultant visits with a stakeholder to talk about what a particular piece of functionality will do for the stakeholder and why it matters. During the discussion the consultant writes down various iterations of the story on a whiteboard until both sides feel comfortable the story is sufficiently precise, complete, and granular so the sprint team can achieve the desired business impact.

Various Agile experts recommend similar story formats, as noted in Wikipedia:

"As a <role>, I want <goal/desire> so that <benefit>"
"As a <role>, I want <goal/desire>"
"In order to <receive benefit> as a <role>, I want <goal/desire>"
"As <who> <when> <where>, I <what> because <why>."
"As a <role>, I can <action with system> so that <external benefit>"²

For example, a user story in the SAP HANA retail scenario might be:

"As a regional manager, I want to see order numbers of any delayed in-process items so I can work with suppliers and shippers to expedite delivery to stores."

The consultant is now equipped to investigate the technical and organizational details needed to make this story happen (like reformatting suppliers' item-level RFID data). Then implementation can begin.

Those who may feel less comfortable with this approach are SAP consultants more accustomed to an implementation-by-modules methodology. On the other hand, consultants who adopt iSAP's Lean methodology will feel right at home. Just as Lean works better with Agile, Lean development works better with stories. Like stories, Lean delivers value in streams or end-to-end journeys that start with the present, flow through a series of intermediate transformations, and end with a desired outcome. By taking a Lean approach to Agile, iSAP-equipped consultants know how to speed implementation and reduce project risk through storytelling.

iSAP's Time-to-Value Advantages

Where Lean manufacturing value streams start with an input (like car parts) and ends with an outcome (like a car) an iSAP implementation might start with an order and end with cash received (an order-to-cash value stream). Other examples are procure-to-pay, manufacture-to-settle, and hire-to-retire. Lean focuses on the quality of the end-to-end process rather than any particular product component (like an SAP module). It's about going from an order to cash received, for example, as painlessly and efficiently as possible for the stakeholder rather than about just making the FI (financial integration) module or the HR (human resources) module work as spec'd. Factors that degrade quality of the journey — anything from a poorly worded help prompt to a badly-sized buffer cache — are considered waste and are continually being fixed or removed, a process called Kaizen. At the same time, any factor that can significantly improve quality of the journey is called design by exception.

Both Kaizen and design by exception facilitate maximum effort is applied to true difference makers while leveraging reuse as much as possible (typically 80-90% of a project). That which is improved will stay improved project-to-project with far less time wasted on efforts that don't contribute to a great stakeholder outcome. So Lean not only advances the quality of the customer's value stream, it also advances the quality of the consulting value stream in reaching the stakeholder's desired goal.

iSAP employs leading practices and proprietary tools to provide a development methodology that is both Lean and Agile. Of particular importance is iSAP's Error Proofing Tool (EPT). The EPT is a repository of value stream content — e.g., functional design documents, PRICEFW³ artifacts, unit tests, and more — from which the EPT can auto-generate value streams pre-populated with that content. That's Lean because it:

- Gives consultants end-to-end visibility across the entire value stream as needed
- Allows the pre-populated content to be refined Kaizen style over many projects and many customers
- Enables easy re-use of value stream content that doesn't need to change
- Allows consultants to focus only on content that will change, i.e., the exceptions

By making each sprint either an exception or part of an exception, the methodology is also Agile. And what makes iSAP even more Agile is the way sprint teams can show how a deliverable works in the context of the value stream. Stakeholders can then validate how it contributes to the quality of the value stream's overall outcome as that outcome has been realized so far in the project. They can also validate the deliverable in the context of the project's value stream — i.e., how far the project has progressed toward its goal.

Which brings up how stakeholders validate deliverables. In Agile they do that by validating the story that called for the deliverable in the first place. But now stakeholders get to see the story unfold in the context of the whole value stream. Instead of just seeing a single scene, they can see the whole movie. And if they want a scene to be edited, they get to see how the change affects the ending. Or if they want a different ending, they can see which scenes need to be cut, added, or edited to make the different ending happen.

But what the technology can't do is tell a story. Only people can. So, it's up to consultants and stakeholders to describe in words what they want the value stream to do, what the outcomes should be, what the benefits will be, and for whom. Any content that doesn't advance this story is called out as exceptions. New, "exceptional," stories can then be imagined and assigned to sprint teams, whose work the stakeholders can then validate by referencing back to those stories.

This sounds great in theory. To make it work in practice two requirements must be met. First you need a methodology like iSAP that actually translates a sprint story into a value stream story. And, second, you need consultants who can tell stories that resonate with stakeholders both factually and emotionally.

After all, you can't have a happy ending until you know what it means.



For more details
contact:

Herbert Goertz

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

Kevin Mulroy

Vice President
kevin.mulroy@capgemini.com
+1 978-302-8319



About Capgemini

With more than 180,000 people in over 40 countries, Capgemini is a global leader in consulting, technology and outsourcing services. The Group reported 2015 global revenues of EUR 11.9 billion. 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

The information contained in this document is proprietary. ©2016 Capgemini. All rights reserved.
Rightshore® is a trademark belonging to Capgemini.