“ServiceNow is incredibly agile. You don’t have to worry about scalability; you just need to make sure the functionality is rolling out at a rate that human beings can absorb.”

This ServiceNow case study is based on a Knowledge 11 presentation by Gustav Hoyer, IT director with Sony Pictures Entertainment (now director of performance improvement with Alvarez and Marsal).

Automate the IT Value Chain

My name is Gustav Hoyer, and I am a Director of Business Consulting at Alvarez and Marsal (ghoyer@alvarezandmarsal.com), a management consulting firm with an IT practice. I apply enterprise architecture (EA) standards, frameworks and discipline to IT as a business unto itself. IT runs like a ‘business within a business’ in many ways: procurement, sourcing, raw materials acquisition, buying servers, purchasing labor, etc.

Before going to Alvarez and Marsal, I worked for Sony Pictures Entertainment, a subsidiary of Sony, which creates films under the marquee brands of Columbia, TriStar and Sony Pictures Classics. They are a large studio with a sizeable IT footprint because they sell their content globally. They have accounting needs in every nation, territory and legislative boundary, and they have complex distribution networks with theaters.

While I was at Sony Pictures, we implemented ServiceNow in 60 days. I’m going to tell you how we did it so quickly, describe some of the architecture and talk about the thinking that drove the decisions we made.

Need for speed

We came from a traditional IT scenario: developers, embedded resources, expertise scattered throughout the organization, lots of parochial knowledge, not business-aligned. In December 2009, the execs informed the leadership of IT that, starting in three months, the company was going to embark on a managed service implementation. They had chosen the partner and determined the high-level breakdown of the organization. In effect, they were reconstructing IT from the inside out and we had a couple of months to get our act together for a largely SLA-driven delivery model.

The point of a managed service arrangement is that an outside vendor handles all your IT delivery – especially things like incident and some of the more transactional things that IT does – and you don’t manage or assign anybody’s work anymore. You tell the vendor, “This is a set of things that you are required to do and this is how we are going to measure you.”

We had to determine how to measure the vendor and how to manage the relationship to a contract. That is the only leverage you have in a managed service arrangement. You are purchasing a service, not bodies. So we needed to retool quickly and we had about 60 days.

This was a big change for us. The measurement discipline hadn’t been a requirement for us because IT was small and in house. We had at least four incident management systems, and change management in place for some of our applications. Our demand and configuration management was Excel-topia, I don’t know what else to call it. There were 50 copies of everything scattered in spreadsheets all over the place. We had four...
major regions – U.S., Asia-Pacific, Europe, and South America. Each of them had some level of ticketing and service request going through Remedy, but no discipline or regularity, other than a common tool.

Our new vendor came in and told us we would need incident, knowledge, change and problem management. We had been on Remedy and Project Portfolio Management (PPM, an HP product), but I knew that, with our licensing constraints, it was not the platform to scale on. It was time for some bold risk so we did a miniature request for proposal (RFP) that included ServiceNow.

**Value chain for IT**

There is a value chain for the IT sub-organization. A value chain is a way to express the activities an organization goes through to generate value. The whole idea of a value chain is to provide a comprehensive view of all of the activities of an organization, hook them together and give them a context. The whole point of IT as a function is to generate internal operating value.

This ties into ServiceNow and ITIL because the frameworks of ITIL are ways to take the functions and services that IT provides, deliver them through some standard nomenclature, and package that service as an industry standard. ITIL gives you benchmarks to see how your peers are doing, how firms in other sectors are doing and how you stack up.

At the outset of our implementation, we asked, “What does this value chain look like for IT itself?” Let’s assume for a minute that we can treat IT like a business within a business and you will see why this helped us get through this 60-day crash implementation.

In the diagram below, you see how the standard value chain model applies to IT. You have IT leadership, management, strategy, planning, enterprise standards, guidelines, and project and program management. The purple area is the product development function, the blue is the supply chain, and the red is the customer support function.

- Demand management is your front door. Your marketing in IT is how you answer the questions, “What do the customers want? What do they need? Where are they going? What is their business strategy? How are they getting there? What are the drivers for their technology decisions?”
- If you have a business relationship management function in your organization, architecture falls next. If you have an R&D function or a place that does pure technology research, the point of this is to say, “How do we get the raw tools? How do we acquire or build the tools to satisfy the customer?”
- Then you start planning how you are going to put it in. So you have project portfolio management, demand planning and release management discipline, where you schedule and initiate the build cycle.
- You have project program management and oversight functions. In procurement and vendor management, you figure out how to get the people and/or the software to put the product together or acquire it.
- Solution delivery is the raw build, the classic coder sitting down at his desk, writing software, like the things you are doing in JavaScript in the ServiceNow platform right there.
- In change management and release management, things are scheduled, managed and tracked, and impacts are identified.
- Infrastructure operations is how you deliver what you make to the customer. Rather than drop a product on someone’s doorstep and leave—

“We had to determine how to measure the vendor and how to manage the relationship to a contract. That is the only leverage you have in a managed service arrangement. You are purchasing a service, not bodies. So we needed to retool quickly and we had about 60 days.”
Case Study

which is how a consumer product would do fulfillment—IT is continuous fulfillment. Deployment is a single act, but fulfillment of the need from a supply chain is ongoing. That is why we have data centers and operation teams.

- As internal IT, you don’t sell to your business in the classic manner of, “We will give you X, Y and Z project for the low price of $500,000 if you buy in the next week; after that, it is $750,000.” But, we are constantly explaining, justifying and rationalizing why IT costs so much to inform them of the value they are getting. Scorecards and communications are how you get this information back to your customer: “These are the things you can do as a result of our help that you wouldn’t have been able to do without IT.”

- Finally, you have the classic functions of service desk and incident management. How does the customer request things of IT? How do we make sure they are fulfilled? How do we make sure they are measured?

This value chain gave us a picture of the health of each of those areas; for instance, we have no process metrics here, no process definition there and we’ve never done any training on how this or that should work. Our job was to make management aware of all the decisions they were neglecting, so the value chain became a great communication tool. IT is a business within a business, and you don’t take a haphazard view of how all those pieces fit together.

So, we broke our implementation out by the items along the bottom row in the diagram.

Implementation

We did a quick financial evaluation on the move to managed service. We had some capital write-off because we had hosted Remedy and moved to ServiceNow. When we went from owning to licensing service and software, we took a write-off and depreciation because a license model is an operating expense, which we don’t capitalize. Then we spent a couple of weeks comparing ServiceNow against Remedy.

We decided in favor of ServiceNow and integration partner Navigis on March 1 and kicked off the project saying, “We are not going to customize the tool. We are going to use ServiceNow as is and assume that has been battle-tested. This is how incident works, end of discussion.” So, we adapted our processes to match the out-of-box ServiceNow capabilities. In 30 days, we had replaced 8,000 users. All of them moved to ServiceNow.

March was a busy month – a bit of a rodeo – but we did it.

“Our job was to make management aware of all the decisions they were neglecting, so the value chain became a great communication tool. IT is a business within a business, and you don’t take a haphazard view of how all those pieces fit together.”
Configuration items (CIs) – We used ServiceNow’s CMDB with the CI configurations and assignment groups that you can attach to CIs to make routing automatic. We listed our applications in the CMDB, then added a button so that our first-level help desk in India just clicks the magic Route button to send the ticket to get resolved somewhere. They don’t need to think about it. The CIs are wired well into incident, and that allowed us to stand up a full process replacement because we automated most of that routing so that it requires minimal training. The beauty of routing it through the CMDB rather than having people reassign tickets manually is that you don’t need to train people where tickets go anymore. You put that knowledge in the system and they get to the right place. They may go to the wrong place, but they will go consistently to the wrong place and that’s fixable. At least they are no longer going to some random number of wrong places.

Problem management – Again, the rollout involved minimal configuration. We had no problem process to replace, but the biggest difficulty was in explaining to people how problem management differs from incident. We told them that the goal of incident management is to get the user working, no matter how. Problem management is when, after you’ve restored service, you research what went wrong, and you do it asynchronously so the user is not sitting around waiting. Without that important ITIL distinction, we had teams saying, “I got an email from my customer, I better go get to root cause right away to get them working again.” There is no energy put into getting the user working again, so the user is dead in the water.

Change management – Change was relatively mature. We had had ITIL training in change, an ITIL implementation several years prior, and we even had a change manager – a good start for change management.

Demand management – This is tricky because of the offshore managed service provider. We need to assess the amount of work to throw at them. How do we make sure that they are delivering what we are paying them to deliver? We wanted a global view of everything from medium-size enhancements all the way to incidents, yet each team was tracking demand separately and there was no way to structure it.

Under Remedy ITSM and HP PPM, we had been building architecture around the notion of a demand continuum of everything anybody will ever ask IT for; from “Map my printer for me,” to “Implement SAP in Hungary.” We wanted a way to segment it, track it, look at it and measure our vendor against it. We came up with a pyramid of demand (see diagram below).
At the bottom are the common transactions of the greatest frequency and quickest turnaround in the service desk function. Some of these become problems, so then you have to deal with problem management.

For some of those, you have to initiate a change because of an incident or problem resolution, which takes you up a level on the pyramid.

Then there is a bucket of medium-size enhancements that require scheduling and take about a month of code changes.

Finally, you have projects, which are large capital expenditures with project managers, plans and steady, progressive rollout.

We were saying, “How do we glue all of this together? How do we make this work as a unified automation platform so IT can run smoothly?” We can hand work from team to team. When one process is finished, how do we notify other processes automatically to decrease the risk of errors?”

We ended up articulating a whole, big-picture spectrum (see diagram below) from operational to strategic work: “Things are broken,” “We had an event detected,” “The operations center says we have an incident,” all the way to, “We don’t know what we want, and we don’t even know what IT needs to do, but we want a placeholder in there.” For our demand module, we adapted the shopping cart and wired together several things we needed for Sarbanes-Oxley (SOX) approvals. When we rolled out incident, we integrated with our Active Directory – LDAP. We had a little bit of both, so we did a LDAP integration that brought every user in the company onto the platform. We were able to take our SOX approvals, customize CIs that required SOX, put those configurations on the CI and drive the SOX approval mechanism during demand. We gave our compliance people a reason to love the platform by including our three SOX approval steps here. We now have an entire factory notion of where the bread-and-butter work is getting done, where the scheduling and planning are getting done and where it all ends up in production.
**Software Development Life Cycle (SDLC) workflow** – We now had an entirely new way to deliver IT and a whole new set of tools, so everybody was dazed for a month or so. As they slowly adapted to the platform, we tackled SDLC. We approached it by saying, “Every activity has a task and many of those tasks have an approval, and we can use ServiceNow to track the approvals.” We’ve outsourced now to this third party, but we don’t trust them, so we want to make sure we are watching every step. Unfortunately, that was built on a foundation of distrust and we realized that is not sustainable. We had so many approvals, checks and steps – something like 15 or 20 steps – that it was ridiculous.

You don’t need every activity to have an approval or a task in ServiceNow. We realized, as cool as that is, it is really frustrating for users. We stripped the approvals down to ensuring that when a task changes hands, the receiving person is aware of it. That was a painful lesson and it cost us in user adoption. It’s easy to write software for systems and change them when you deploy, but software for people processes is slow. I will say that again, because it was a good distinction for us: We came to rely on ServiceNow for recording the decisions people were making rather than making it have to drive all of that.

**Takeaways**

We have been deeply impressed with ServiceNow. It’s a robust platform for deploying processes and managing workflow, regardless of the business unit. We could have said, “We’re doing incident management, ITIL tells us it’s this little bucket, and we are just going to stay there.” We quickly learned the whole platform is a process automation tool.

“We have been deeply impressed with ServiceNow. It’s a robust platform for deploying processes and managing workflow, regardless of the business unit.”