

A photograph of a modern office interior with a glass wall and a person standing near a window. Overlaid on the image are several large, semi-transparent, multi-colored hexagons (pink, orange, yellow, green, blue) that intersect and overlap each other.

An Executive View of Lean and Agile IT for CFOs

Mark Schwartz, Enterprise Strategist
at Amazon Web Services

Over the last two decades, the IT profession has developed new ways of working that are intended to deliver better business value faster and at lower risk.

There are buzzwords associated with these techniques, of course, as with everything in IT – in this case, Agile, Lean and DevOps are the terms to know. Unfortunately, these techniques are often presented as IT-focused, with unclear benefits for the enterprise. They even seem to bring with them a danger that they might undermine the CFO's or CEO's ability to oversee IT-related initiatives.

As the CFO's role continues to expand, new approaches to technology are becoming critical to delivering business value. This eBook shows how Agile, Lean and DevOps streamline digital delivery, foster innovation and make the enterprise nimbler. These practices could very well be the best thing that has happened to CFOs since the invention of the spreadsheet, helping them to increase returns, oversee investments, implement controls, gain transparency, provide better data to the enterprise and, of course, manage costs.

Lean

Lean IT delivery is based on the same concepts as Lean Manufacturing and Lean Six Sigma, all imported from the Toyota Production System that pioneered them. The general idea of Lean is to eliminate waste by concentrating on reducing lead times. When an enterprise adopts a Lean point of view, it maps out the processes it uses to deliver value and examines each step to find waste that makes the process take longer than necessary. By eliminating each type of waste, a business can shorten its lead times and reduce its costs.

IT delivery is amenable to a Lean approach, although IT must be thought of as a product development process rather than a manufacturing process (that is, a process which is different every time it is performed). For this reason, Six Sigma techniques, which try to reduce variance, are not generally applicable. But as with other business processes, delivering IT capabilities involves a series of steps, each of which often contains waste. The typical sources of waste in an IT process correspond fairly closely to those in a manufacturing process. Authors Mary and Tom Poppendieck have identified them as partially done work, extra processes, extra features, task switching, waiting, motion, and defects.¹



“

The idea of deliberately diverging from a plan might sound dangerous. It seems like it would be impossible to control and to hold people accountable. But in fact, it is not.”

So why is it so important to eliminate waste and shorten lead times in IT delivery? One reason is to reduce time-to-market, of course; or for internal use capabilities, the time until value can be harvested from an investment. Another reason is that speed helps make sure IT capabilities are as effective as possible. IT teams can now deliver minimal viable products to users quickly, check to make sure they are accomplishing what they should, and then continue adding features or make changes. Speed creates business agility. Because capabilities are being finished constantly, IT is able to pivot and work on other things that become more important without wasting any of their previous work. And finally, speed reduces risk, both because unforeseen events can be quickly addressed and because the delivery risk of IT capabilities is reduced.

The value stream for delivering an IT capability is different in every organisation, but typically it includes steps like this: the business need is recognised and expressed, requirements are written, a plan is created based on the requirements, a business plan is prepared, a governance process acts on the business plan, resources are acquired, software is developed, software is tested, security is verified, software

is deployed, users are trained, and the capability is launched. This is a long process – and it can hide a great deal of waste. Much of the process (and the potential for waste) is outside the direct control of the IT organisation, or it is at the interface between it and the rest of the business. A careful look at the processes informed by Lean software delivery techniques can make a substantial difference in business outcomes.

Agile

Agile IT delivery is based on a simple principle: in a complex environment (which IT delivery certainly is) it is better to learn and adjust than to follow a plan made in advance strictly. (I'll explain in a moment how this is connected with Lean). The idea of deliberately diverging from a plan might sound dangerous. It seems like it would be impossible to control and to hold people accountable. But in fact, it is not. It is rigorously controlled but through very different mechanisms.

I sometimes like to think of Agile techniques in terms of risk mitigation. If we make a detailed plan that covers, let's say, a three-year project and then try to implement it, we are accepting a number of large risks, notably:

1. The risk that the plan will have mistakes.
2. The risk that circumstances will change over those three years.
3. The delivery risk that in three years the product will not have been completed.

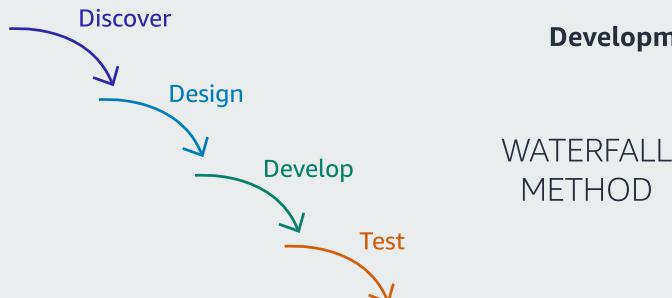
In the traditional way of delivering IT (the so-called Waterfall, or Gantt chart-obsessed approach), the product is not delivered until the end of the project, so the entire amount of the investment is at risk until the end of the three years when the results become visible (or not).

How serious are these risks? Very.

- Detailed plans for IT systems are **always wrong**, as we have found in our experience over several decades. Studies have also shown that more than half of the features requested will rarely or never be used. And even if everything is delivered according to plan, it still might not meet the business need it was intended to address.

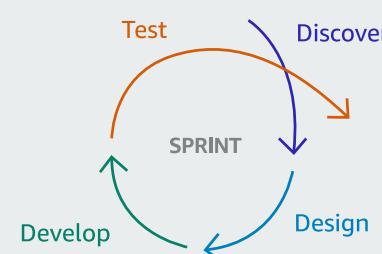
- The amount of change we expect over the three years depends on the amount of uncertainty in the business and technical environments, and **we happen to be in a time of fast change and high uncertainty**. In the course of three years, start-ups are launched and disrupt entire industries. Agility is the ability of the business to respond to change, and it is contrary to sticking with a plan, yet extremely valuable.

- A Microsoft study showed that **only one-third of ideas actually have the intended result**, another one-third have the opposite effect and the last third are neutral. We have all seen cases where an IT system was supposed to reduce costs but didn't, or was supposed to increase revenues but didn't.² Another study found that companies are wasting nearly \$400 billion per year on digital initiatives that don't deliver their expected return.³
- Numerous studies have shown that the larger an IT project is, the higher its risk of not delivering. A **short increment of work is more likely to accomplish its goal** and is more predictable.



Development Methodologies

AGILE
METHOD



Agility is the ability of the business to respond to change, and it is contrary to sticking with a plan, yet extremely valuable."

Agile practices allow for constant learning and adjustment, **working in small cycles** to finish a product and give it to users for feedback. It is practised by small, autonomous, self-contained teams that can **learn and adjust quickly** with minimal ceremony. Because Agile teams are trying to finish the product quickly, it is natural to combine Agile principles with Lean practices, which focus on reducing cycle times. The combination of the two gives businesses agility, risk mitigation and cost-effectiveness.

DevOps

In Lean theory, two important sources of waste are handoffs (motion and waiting) and large batch sizes. Traditional IT practices were based on handoffs between development, testing and operations, who deployed the product. DevOps, the state-of-the-art set of practices in Lean and Agile IT, addresses these handoffs by combining development, testing and operations skills in a **single, small team** accountable as a whole for results.

Large batch sizes in IT are large groups of requirements. A DevOps team only processes a small set of requirements at a time, using a highly automated process to quickly deploy capabilities to users before moving on to another small set of requirements.

As a result, DevOps teams deploy code very often – sometime hundreds or thousands of times a day.

The heavy use of automation in DevOps has benefits for controls and compliance. Many of the organisational controls that would have been operated manually can now be automated, making them more reliable and easily documentable, and allowing them to be **applied continuously** rather than periodically.

DevOps is an excellent way to foster innovation. With it, new ideas can be tested **quickly, inexpensively and at low risk**. Working in the cloud enhances these benefits: infrastructure can be provisioned instantly and then be de-provisioned later. IT capabilities that would take the enterprise a long time to build can be accessed as pre-created building blocks and incorporated into experiments.

The business implications of Agile, Lean and DevOps

- Fast time to market or time to value for internal use products
- Less waste from producing unrequired capabilities
- Less waste from producing capabilities that do not accomplish objectives
- Less waste in processes (both inside and outside of IT)
- Reduced risk
- Increased innovation
- Better operational controls through automation



Let's return to the issues of control. In the traditional approach to overseeing IT initiatives, governance is primarily an upfront matter: once a go/no-go decision is made, overseers are generally uninvolved unless the project breaches thresholds or until certain milestones are reached. It is a discrete oversight process, popping into the picture at intervals but otherwise absent.

You can think of the Agile approach as one of continuous oversight and transparency. The project team delivers frequently and results are apparent as those deliveries are made. Because of the agility of the process, the oversight body can choose to change direction at any moment, end the investment, increase it or substitute other objectives.

With an Agile process, the enterprise can fix and hold to a schedule or budget; it simply trades off scope to meet the schedule or budget. I suggest holding most cost categories fixed, just as one does with a budget. Budgets place a cap on what an organisational unit can do in a single budget cycle; once you run out of budget you stop spending. It's the same with so-called "requirements". (So-called because they are not really "required" but subject to budget availability.) If money runs out during an Agile IT project, then nothing has been lost because the work that has been finished is usable. In fact, the initiative can be terminated early, even if budget remains, based on changing priorities or because enough success has already been achieved. Or the enterprise can make a conscious decision to increase the budget to implement the remaining features. Agile is a continuous investment process where the business case is (effectively) recalculated every moment.

"

Agile is a continuous investment process where the business case is (effectively) recalculated every moment."

Agile approaches place a high value on adjusting a plan as information becomes available, and a low value on conformance to the plan per se. But that doesn't mean that it is uncontrolled. I think of it as being controlled by conformance to business objectives. As the project progresses, features are rolled out, their business impact is gauged and the project is adjusted based on its impact on the intended objectives. The object is to get the best return from whatever is invested.

But you have to have agreement on what that "return" is intended to be. Cost savings? Increased revenue? Better customer service? Long-term agility?

These are all valid goals. On the other hand, meeting all of the requirements specified at the beginning of a project is not a real business objective, nor is building a particular set of features. A healthy project is one that meets its objectives, not one that finishes all its requirements. It should continuously adapt in order to best accomplish those objectives, given reality.

CFOs should be as excited about these new IT approaches as CIOs are. They provide ways to get better results for the enterprise by taking advantage of what is now possible with IT tools.

About the Author

Mark Schwartz is an Enterprise Strategist at Amazon Web Services and the author of *The Art of Business Value* and *A Seat at the Table: IT Leadership in the Age of Agility*.

References

- ¹ Poppendieck, Mary and Tom Poppendieck. Lean Software Development: An Agile Toolkit, p.4. These correspond to the classic sources of waste in Lean theory: inventory, extra processing, overproduction, transportation, waiting, motion and defects.
- ² Humble, Jez, Lean Enterprise, p.179
- ³ <http://www.genpact.com/insight/article/cfo-challenges-in-a-digital-world>