Agile: The World's Most Popular Innovation Engine

Jul 23, 2015 @ 10:56 AM

"Innovation," says Curtis Carlson, former president of Stanford Research Institute (SRI) and author of the classic book, *Innovation* (Crown, 2006), "is the primary driver of prosperity, job growth, social responsibility, environmental sustainability and national security."

And yet today, across the economy as a whole, by almost every measure, innovative performance is poor. For example, I discussed in an article in February 2015 a survey by MindMatters of 150 firms with innovation teams that varied from less than a hundred to over a thousand staff in a wide variety of industries. The survey showed that only 5% of respondents said that innovation staff feel highly motivated to innovate. Only 6% believe their employers regard the development of intellectual capital as a mission-critical function. Around 75% say their new ideas are poorly reviewed and analyzed. What's important to note is that these respondents are the implementers of innovation. These are the people who should be the champions of innovation and yet they have become sharp critics.

This dismal picture contrasts sharply with the scene in software development where those involved in innovation report that innovation is thriving. A survey published today shows that the innovation methodology known as Scrum is being widely used and is regarded by a majority of its practitioners as successful. (Disclosure: I am a board member of Scrum Alliance.)

One key element in the poor results of the economy-wide picture of innovation is that there is no generally agreed set of processes for systematically creating innovations that generate better, cheaper, faster, lighter, more convenient and more personalized products and services for customers. Instead innovation generally proceeds on an ad hoc basis, in

projects with big plans pursued in a sequential fashion that rarely turn out as expected.

In software, the scene is very different. Over the last 20 years, systematic innovation methodologies has been developed, tested and implemented around the world on a large scale. These methodologies go under the overall banner of Agile and include many different methodologies including Scrum, Kanban, Lean Startups, XP, Dev.Ops and Continuous Deployment.

The State Of Scrum

This week, a global survey entitled "The State of Scrum" was released by Scrum Alliance, a rapidly growing association of some 400,000 members. Its mission is to transform the world of work. Around 5,000 people responded about their use and views of Scrum. The respondents came from 108 countries and 14 industries. The respondents were predominantly in software development and IT (77%), although there was also a range of functional areas represented, including product development, operations, human resources, executives and sales and marketing.

How Do Agile And Scrum Work?

The approach known as Agile is based on the Agile Manifesto (2001). The most popular Agile methodology today, Scrum, involves a set of well-developed practices for innovation. A simplified summary could include the following core practices.

Work is organized in self-organizing teams in short cycles aimed at continuous improvement. The management is responsible for removing impediments in the team's work and doesn't interrupt the team during a work cycle. The team reports in effect to the customer, not the manager: The customer's priorities are systematically analyzed and fed into the work of the team. The team itself estimates how much time work will take, how much work it can do in an iteration, as well as how to do the work in

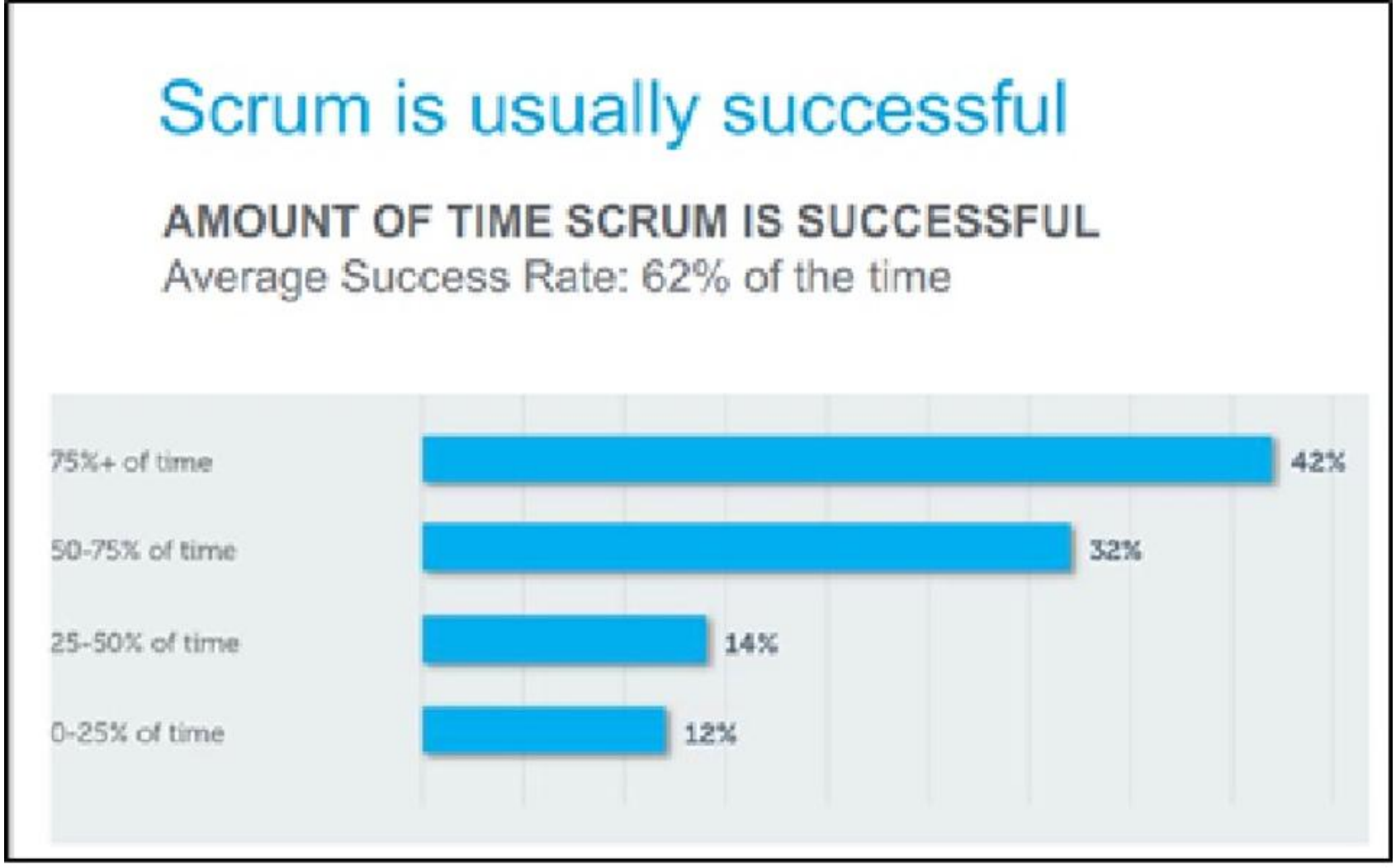
the iteration.

Performance is measured by customers or customer-proxies. Work goals are defined before each cycle starts. The lessons learned in each iteration are systematically captured in retrospectives and used in future iterations. In this way, the products are not only constantly improved: the process for developing them also improves.

Key Findings

The key findings of the State of Scrum survey of Scrum practitioners are:

- **Scrum is being widely used**: "nearly half the respondents report that Scrum is being used "50% or more of the time in their organizations," and 29% of respondents report that is used "much more frequently than that."
- **Respondents believe Scrum works**: The overall success rate of projects delivered using Scrum reported by respondents is 62%. (Teams of the recommended size for Scrum — seven plus or minus two – members — report the most frequent success, while smaller and larger teams both report less frequent success.) Given the dismal results for innovation recorded on an economy-wide basis, this is a very encouraging result.



- **Respondents believe Scrum helps customers**: One reason for Scrum's success is its focus on the customer. Nearly half the respondents (49%) cite fulfilling customer needs as the highest business priority for Scrum projects.
- **Respondents believe Scrum helps the business**; The second reason for success is that second-highest priority is all about meeting the needs of the business — meeting budget, time and scope constraints. This reflects Scrum's focus on delivering shippable increments on time and within budget.
- **Respondents like Scrum**: Scrum improves the quality of work life. 87% agree that Scrum is improving the quality of work life for their teams.
- **The respondents actually practice Scrum:** Despite reports of a large amount of fake Agile and Scrum out there,most respondents report that they adhere to core Scrum and standard recommendations for practicing Scrum in terms of using Scrum artifacts and activities and following the recommended roles and team size.

- The average team size is 7 people.

- Most Scrum teams (60%) follow 2-week sprints.

- 81% hold a team Scrum each day.

- 83% conduct sprint planning prior to each sprint.

- **Co-location of teams is limited**: Although co-location of teams is regarded as best practice, distributed teams are more prevalent than co-located teams. 33% of respondents report their Scrum teams are distributed, versus 26% whose teams are co-located. This reflects the reality that business has become global. The goal of keeping teams co-located has to compete with the priority of having teams close to customers.
- **Firms mix-and-match Agile frameworks**. There are many Agile frameworks including Scrum, Kanban, XP, Dev.Ops, Lean Startups and Continuous deployment. Respondents reported that many organizations are adopting a mix-and-match approach. Only 42% report using Scrum exclusively.
- **High-level support is critical**. Respondents report that senior management sponsorship and support is far and away the most important factor in adopting Scrum.

Opportunities For Future Improvement

Respondents also flagged several significant problems:

- **Measurement**: The most common challenge for respondents—52%—is identifying and measuring the success of Scrum projects.
- **Transitions**: 46% of respondents report problems in transitioning from bureaucratic Waterfall-based methods to Scrum practices.
- **The clash with corporate culture:** Many respondents see tension with the rest of the organization: 71% also believe that using Scrum causes tension with other parts of the organization not using Scrum. This is not surprising, as Scrum requires a shift in an organization's culture from a top-down vertical mode of management focused on

producing outputs to a horizontal collaborative mode focused on delivering value for customers. Changing corporate culture remains the largest challenge ahead for innovation.

These findings suggest that there is no room for complacency. In some areas, Agile and Scrum communities have been slow to innovate on their own methodologies. For instance, only limited attention has been paid to Kent Beck's call in 2011 to apply "inspect and adapt" to the Agile Manifesto itself.

Like the MindMatters survey of innovation economy-wide mentioned above, the report on Scrum is a survey of the opinions of practitioners, not a study carried out by third-parties of actual results. It reports on perceptions of the practitioners and generates hypotheses, rather than proven conclusions. The hypotheses need further work to determine whether the perceptions and the hypotheses are correct. It will be helpful if future editions of the report include work being done in academia and elsewhere to validate or disprove the encouraging hypotheses that have been generated.

Thus it will be helpful if future editions of the report could shed light on:

- Who is in the control group and what would another group report is their customer focus?
- How is the effectiveness of Scrum itself best to be measured so that Scrum can be put into a broader, more meaningful comparative context?
- How can we get a handle on the issue of management mindsets which often appear to be a more important determinant of success of innovation than methodologies?
- What data is available on the spread of Scrum within the organization over time?
- What are the strengths and weaknesses of the different Agile methodologies in addition to Scrum?

In this way, the report could become a more comprehensive account of

"the state of Agile and Scrum," not just the views of Scrum practitioners.

The Contrast With Bureaucracy

The standard command-and-control processes of hierarchical bureaucracy that are still prevalent in large organizations today are behind the perceived poor performance in economy-wide innovation. Those processes are ill-suited to innovation. Managers recognize this but what often happens is that teams are deployed on an ad hoc basis when the organization see problems that the bureaucracy can't solve. Once the teams have solved the specific problem at hand, the teams are disbanded. There is no specific capability created for systematic innovation and any that are created, such as "innovation stage-gates," tend to be infected with control-minded bureaucratic mindsets.

Agile and Scrum had their origins in manufacturing, particularly the lean methodologies developed in Japan. It's easy to see why they took off in a major way in software.

One reason is competitive pressure. Unlike hardware, software is continuously and infinitely malleable. If you buy a car, you are stuck with the car you bought until you buy a new car. Since software can be continuously upgraded, competition in the marketplace requires that firms be continuously improving and innovating on their products.

Secondly, the horrifying logic of the blue screen. Unlike most other fields, the work of software is so complex that it isn't amenable to direct managerial authority. In other fields, a manager can tell people to "fix the sales campaign" or "find some cost savings" and have a reasonable expectation that those doing the work can come up with something that looks like a plausible response. Computers are less forgiving. A blue screen is still a blue screen, and doesn't respond to a boss who screams at it.

In the 1990s, huge sums of money were being lost because the work of software development was always late, over budget, and plagued by

quality problems. Clients were upset, and firms lost money. Software developers were seen as culprits and were punished. They worked harder and harder. They labored evenings and weekends. They fell sick. They were replaced. It made no difference. The software was still late, over budget, and full of bugs. The standard practices and prescriptions of management didn't work with software development. It was frustrating for managers to find that the more they tried to control things, the less progress they made. Complexity responded to competence, not authority. Something different *had* to be found.

A Different Management Ideology

Thus Agile and Scrum are more than management processes. They constitute a different ideology. Instead of the prevailing vertical management ideology of control, Agile and Scrum reflect a horizontal ideology of enablement. The goal is to create workspaces that can draw on the full talents and capabilities of those doing the work to deliver value to customers, and systematically remove any impediments. This ideology is a good fit with a business environment that requires continuous innovation for success.

Now as software shifts from its former role as a minor backroom function to the front-lines of business on the interface with customers and a key determinant of success in the marketeplace, it is not only, as Marc Andreessen quipped, that "software is eating the world:" It's *innovation in software* that is eating the world. As a result, Agile and Scrum are now spreading from software development across the economy. Already 23% of the respondents in the survey come from outside software development and IT.

As the results of innovation in software development become better known, and businesses become more and more dependent on innovation in software for competitive advantage, it is plausible that Agile, Scrum, Lean Startups, Kanban and their analogues will increasingly receive due recognition as not only the most popular, but also the world's best innovation engine.

And read also:

[Why Software is Eating the World]

[Is there an Innovation Crisis in US Firms?]

Scrum is a Major Management Discovery

[Why Do Managers Hate Agile]

The Best-kept Management Secret: Agile

[The case against Agile: ten perennial objections]

Follow Steve Denning on Twitter @stevedenning

Also on Forbes:

Gallery

Advice From The Most Innovative Leaders

Launch Gallery