Many teams are thriving on agile practices including the highly effective scrum methodology. Implementing scrum in large scale distributed organizations can be challenging. This article suggests practical strategies for managing scrum at scale.

Developing large software systems can be a challenging endeavor, often completed in pieces by different teams tasked with handling complex dependencies while simultaneously needing to manage competing resources. Creating the next internet startup at times requires overcoming significant hurdles, but running a large-scale project for an international bank or financial service firm also requires a level of coordination and communication that can be challenging. The good news is that agile methodology is shown to be increasingly effective, even when adopted by large enterprises. The key is to understand how to apply key methodologies, such as scrum, across the entire enterprise.

What is scrum?

Scrum is an Agile framework to guide a team through all the tasks required by the project. In practice, scrum involves a short daily meeting to:

- Discuss progress
- Review what each team member is working on
- Identify any barriers to success that might exist

Scrum typically does not require the type of time-consuming bureaucratic processes which some agile enthusiasts describe as being "ceremony". Scrum is an iterative methodology that focuses on completing work within a fixed time period, usually lasting two to four weeks, known as a sprint. The Scrum master plays an essential role in coordinating the activities of the team, particularly with regard to addressing any barriers to success. Requirements are specified by descriptive user stories stored in a product backlog and organized into sprints. The goal, when feasible, is that the
work completed at the end of each sprint is potentially shippable and is followed by a sprint review and, often, a post implementation review, called a retrospective. Scrum is popular because it helps each stakeholder understand what they need to do on a daily basis and provides transparency into what other members of the team are working on. Most importantly, the scrum helps to identify any potential barriers to completing the work and helps to determine the best course of action to remove any potential obstacles.

This methodology does not require much process and is effective at helping the team coordinate all of its activities while establishing priorities and target goals.

With each sprint completed, the team chooses another set of user stories to complete and begins its next sprint iteration. This process usually works well for small to medium projects, but can be challenging to implement in larger organizations.

The challenge of enterprise scrum

Scaling the scrum framework across the enterprise requires that you coordinate activities among many stakeholders who might have different goals and priorities. In practice, you can have several teams working on different aspects of the system at the same time. Large organizations use a methodology called scrum-of-scrums in which each of the scrum teams sends a representative to the joint meeting to coordinate the activities of the different development teams. Many applications involve n-tier technology architectures whereby you can have work being completed on several platforms - including mainframe, Linux/Unix, Windows and even mobile. Key stakeholders frequently include systems, middleware and database administrators. It can be challenging to coordinate work across all of the involved groups.

Roles and responsibilities

Although scrum is a relatively lightweight in terms of process, there are important roles and responsibilities.

The product owner plays a key role in setting the direction for the entire team. In large organizations, product owners are often focused on working with other stakeholders to understand and drive the corporate objectives. Product owners help to determine priorities in light of outside forces, from competitors to industry regulatory requirements. The product owner helps to evaluate and manage risk while guiding the team to make choices that lead to running profitable businesses.

The Scrum master works with the product owner to cultivate a product backlog describing the work that needs to be accomplished in the form of user stories. The Scrum master also guides the team to embrace agile principles, and acts as a coach while ensuring that there is just enough process to avoid costly mistakes. The Scrum master helps to address any potential barriers to success.

The development team consists of technology and quality engineering resources who implement the requirements described in the user stories.
The scrum is best described as a high-performance, cross-functional team. Its members might be asked to perform a variety of tasks including:

- Analysis
- Software development
- Automating testing
- Quality assurance

In larger organizations, the scrum team might consist of dedicated resources who are technology specialists and assigned to the scrum for however long it takes to complete its work. Many agile enthusiasts expect that scrum team has full control over its activities. However, this is not always possible in practice.

**Organizational structure**

If you work in a large company, you might find that the organization has separate groups to support database administration, networking, and other specialized technical activities. This might not feel compatible with the spirit of a scrum, which is based upon a self-organizing high performance team. But, these functions need to remain in place for a variety of reasons, including support of legacy applications. In these situations, you typically want to focus on improving communication and collaboration between all stakeholders. While smaller firms, including internet startups, can embrace agile practices in a more comprehensive way, most large organizations often navigate a less direct journey. They typically implement agile practices within the existing organizational structure and governance systems. You should view this effort as taking an agile iterative approach itself to achieve optimal results.

**Activities and artifacts**

The scrum framework starts when the product owner and other stakeholders work together to create a product backlog. In large organizations, this can be a complicated process as the goals and priorities of the organization are considered. The product backlog is groomed to identify a set of features to deliver. The work to deliver these features is planned to be completed in a fixed timebox period known as a sprint.

Most folks describe the result of sprint planning as a commitment or forecast. The term forecast is specifically used to indicate that, similar to weather reports, the "prediction" itself can change, consistent with agile principles, to welcome new requirements even late in the process. This means that the forecast might need to be revised as the sprint is executed. It is important for the people involved to understand that they can make their "best guess", without having to resort to padding estimates or low-balling their commitments, to avoid missing their goals. The focus is always on deciding which requirements should be delivered as described in user stories while minimizing short-term decision making, which often results in technical debt. The scrum process has minimal process and does not require verbose documentation. It does require artifacts, including the product and sprint backlogs, user stories and sprint plans. In large organizations, you might have requirements for more documentation and planning to meet IT governance which senior management expects and requires. To put this simply, no one is giving you a million dollars
for a large software development project without asking you to tell them what and when you are going to deliver.

**Managing distributed teams**

In large organizations, teams can be distributed across multiple continents and work in different timezones. This makes it difficult for members of the team to be collocated in one place or even to meet all of the members on their team. While face-to-face interactions are preferred, distributed teams often have to make use of video and audio conferencing solutions in addition to whatever travel is permitted. One approach is to allow team members to be embedded into another group on a temporary basis. This at the least helps build working relationships. Other challenges are the unique language and cultural requirements found in large-scale distributed scrum.

**Language and culture**

Teams from different countries often have different cultures and languages which can significantly impact the manner in which they interact and communicate. Understanding each other is only the first step. Different explanatory styles can lead to conflicts and challenges. Some cultures are very upfront and perhaps best described as exhibiting "in your face" communication. Other folks might find this approach offensive and even resort to tactics to avoid these uncomfortable interactions. This means that if you are unaware of how you are perceived, other colleagues might resort to passive aggressive behaviors, such as ignoring your emails. This could prevent you from obtaining the desired results.

**Managing timezones**

Scheduling the daily scrum with a distributed team is a challenge. Some teams in one location take meetings late at night, a specific challenge with own potential pitfalls. Some successful organizations establish follow-the-sun support teams. Continuously expecting one group of people to work offhours brings its own set of challenges.

**Agile principles**

One effective way to manage the many challenges of large-scale adoption is to ensure that the team receives excellent training in agile principles so that its members can apply agility in a pragmatic way. Too often, teams focus on the appearance of being agile. If you want to be successful, make sure that every colleague on your team really understands agile principles and can apply them in real world situations.

**IT governance**

Most organizations have complex project management and governance requirements. Distributed scrum requires that you creatively implement agile practices while still operating within the existing organization framework. This means that you must be mindful of the IT governance requirements which are imposed by senior management to enable them to have the information that they need in order to make the right decisions. Closely related is the need to comply with audit and regulatory compliance.
Compliance

Many large organizations must adhere to federal regulatory requirements. This imposes significant burdens upon the development and operations teams. We try to focus on using these requirements to reduce risk and avoid human error, which, after all, in most cases is the original intend of the regulatory requirement.

Managing the full lifecycle

Frequently, organizations focus only on specific tasks which are urgent and must be completed in a timely manner. The full software and systems development lifecycle must be considered. Not just developing and customizing systems, but that you also plan for the deployment and production operation of the system. This means that you need to consider developer velocity as well as the operational stability that is a must-have for any production environment. One key consideration is the need for continuous and comprehensive testing.

Continuous testing

The scrum is responsible for ensuring that the code that is developed meets acceptable quality standards. In larger organizations, you’ll find that QA and testing resources are part of a centralized organizational function required for audit and regulatory compliance. You can certainly collocate these resources with the development team and have them fully participate in the scrum, whether or not they actually report into the development organization. In many cases, however, it is preferable to have your testers not reporting into a development manager because they’ll feel safe to throw down the red flag when systems have defects or other issues that could indicate the code is not ready for production usage.

Managing change

Managing change is an important aspect of using scrum in a large enterprise. While your typical internet startup can make changes on the fly, large organizations need to coordinate their changes across all of the departments likely to be impacted. This means that your scrum needs to coordinate with the organizational change control function.

Retrospectives

Retrospectives are a key aspect of process improvement and should be held at key points, including after each release. The retrospective provides a valuable opportunity to discuss what went well and what can be improved. In an enterprise environment, operations should be involved with this process. More importantly, meetings held by operations should be aligned with any distributed scrum effort.

Postmortems and retrospectives

When serious outages occur, most organizations need to conduct a postmortem to understand exactly what happened, and how to prevent similar or related problems in the future. With
distributed scrum, your agile retrospectives should be aligned with the postmortems that help to reduce risk. This dual-pronged approach enables your development technology experts to provide their expertise to the process improvement effort.

**Continuous process improvement**

The journey to optimal process involves many challenges and should always be thought of as being an iterative process in and of itself. Handling scrum in an enterprise environment might require some creative tweaks to achieve the best results. Remember to include all of the key stakeholders in the effort to continuously improve your process.

**Conclusion**

The scrum framework is effective at helping to organize all of the activities required to complete complex software and systems development. Managing scrum in a large enterprise can certainly be challenging, but it can yield amazing results and directly contribute to the success of your company.
Related topics

- Learn about scrum
- Tips and tricks to get around your next DevOps chokepoint
- Scrum project management with Rational Team Concert

© Copyright IBM Corporation 2016
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)