

How Proper Tooling Can Help Scale Agile Faster

What Does Scaling Agile Mean?

While many organizations start their Agile adoption initiatives with team rooms, whiteboards and sticky notes, organizations that are looking to scale their Agile practices have found that they require an integrated solution to help – a fact that has been continually reinforced by Telerik’s customers around the world. There are a few very specific reasons for this need, however, before we explore these let’s first take a look at what it means to scale agile.

Scaling Agile has always been on the minds of teams and organizations. The definition of scale, however, may mean more than simply spreading Agile practices to more people within the organization. In fact, there are a few different perspectives of the term “scale” when it comes to Agile. There have been many Agile thought leaders who propose models for scaling Agile, each with slightly different perspectives of scale.

Two extremely prominent thought leaders in the area of scaling Agile are Scott Ambler and Dean Leffingwell. Ambler is a pioneer in Agile at Scale and Disciplined Agile Development (see Figure 1) and suggests that scaling Agile goes beyond teams size; taking into consideration other important organizational

aspects such as the physical locations of teams, compliance requirements, domain complexity, organizational distribution, technical complexity, organizational complexity and enterprise discipline (“The Agile Scaling Model (ASM) Adapting Agile Methods for Complex Environments”, IBM Rational Software, December 2009).

Leffingwell takes a different perspective to scale - associating scale to the scope of work being performed and managed. Leffingwell’s Scaled Agile Framework (see Figure 2) demonstrates how Agile scales from the level of a team (focusing on user stories and iterations) to programs (which focus on features and releases) to portfolios (which focus on business and

What Does Scaling Agile Mean?

architecture vision expressed in epics and roadmaps). In this model portfolios span a considerable amount of time and software releases and can be combination of a number of initiatives, each containing one or more teams. In Leffingwell's model, scaling Agile is more about alignment across different perspectives of business.

In any case, scaling Agile depends on organizational change, which in itself centers on changing the way people work. It's no wonder that organizations struggle with scaling Agile since this approach can't solve this problem by simply flicking a switch or installing a tool. With that said, tools become an important instrument of Agile regardless of how organizations view scale. Let's take a look at a few of these to see why tools can help.



Figure 1: Aspects of Scale by Scott Ambler

What Does Scaling Agile Mean?

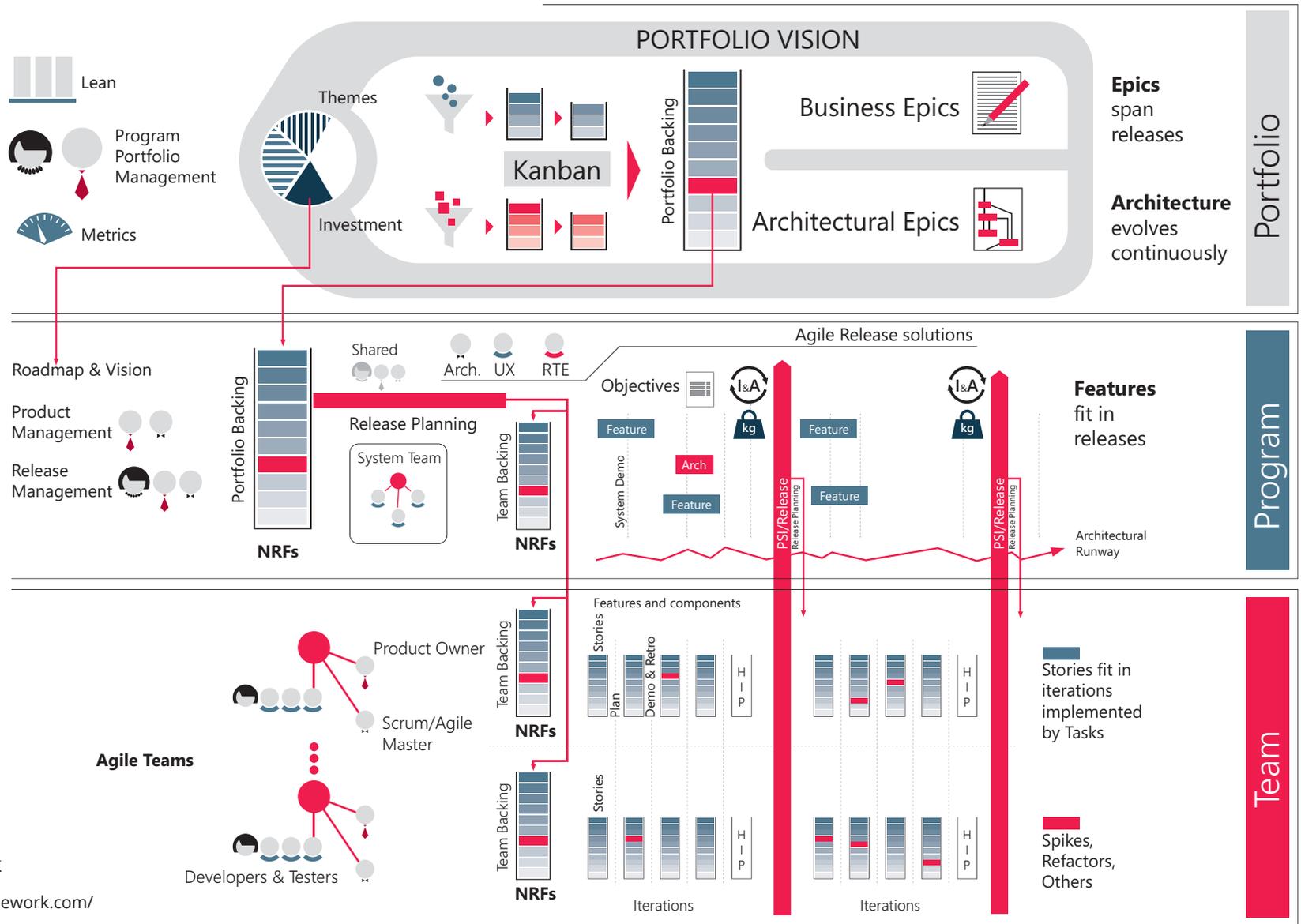


Figure 2: Scaled Agile Framework

Image taken from: <http://scaledagileframework.com/>

Scaling Agile in Realistic Environments

A small, non-distributed team that has continual onsite access to its customer is the utopic model for an Agile team. This team can get by with a few whiteboards and some sticky notes to help manage their requirements and flow of work. In fact, some might say that in this situation tools can get in the way of the team. But what happens if:

- The team members happen to span more than one physical location?
- The organizations are working on an extremely large initiative that has many sub-teams that all must be coordinated as a whole?
- Your customers are not onsite?
- There are strict regulatory requirements that must be met and audited?
- The teams are required to maintain a certain service level agreement or metrics driven targets in order to comply with certain contractual requirements?
- You want to leverage complex metrics to better understand your team's performance in order to help pinpoint waste and opportunities for process improvement?

Do we just claim that there is no way to be Agile in these situations or can we look to tools to enable agility in a non-utopic, yet command and realistic work setting? Let's take a closer look at these very "un-Agile" environments to see how tools can help.

The Distributed Team and Tools

First, let's take a look at a distributed team, where team members are working in physically separate environments and even time zones. This common situation quickly breaks down the value of the whiteboard with sticky notes solutions, although some teams have attempted to overcome this using a web camera. The real issue with having a distributed team is not a shared whiteboard, but simply quality communication and collaboration, exasperated when teams are scattered across a wide spectrum of time zones.

Teams have turned to all sorts of tools to help them out in these situations; from the use of Skype to provide an open video and audio channel between teams, to the use of software based team boards to help keep everyone up to date. Clearly, the more distributed and spread out the team, the greater the need for tooling to help keep everyone synchronized and up to date – something that comes effortlessly in a small team working in the same physical environment.

Tools can facilitate contextual collaboration to distributed teams, helping to ensure that conversations about work are maintained about that work. In addition, tools can help ensure that other team members are alerted to conditions that matter to them immediately as they occur. For example, if a customer provides new feedback or feature suggestions your team can immediately get notified and begin an electronic dialog with the customer to gain further insight – a dialog that will be saved along with the customer feedback for the rest of the team to see regardless of location or time zone.

Central to all forms of distributed collaboration is to have a common system of record to collaborate around. A common system of record is essentially a centralized repository of information that your team needs to do work around. For example, when software is deployed having a bug tracking system becomes a system of record you will immediately want to establish. You will also need to help establish lists of customer feedback, feature requests, requirements, risks, and issues. Tools help facilitate the collection and maintenance of these system of records, and advanced tools such as Telerik TeamPulse will allow your team members and customers to collaborate around each item to ensure conversations are always tied with the appropriate context.

The Distributed Team and Tools



Figure 3A: Distributed team with no centralized system

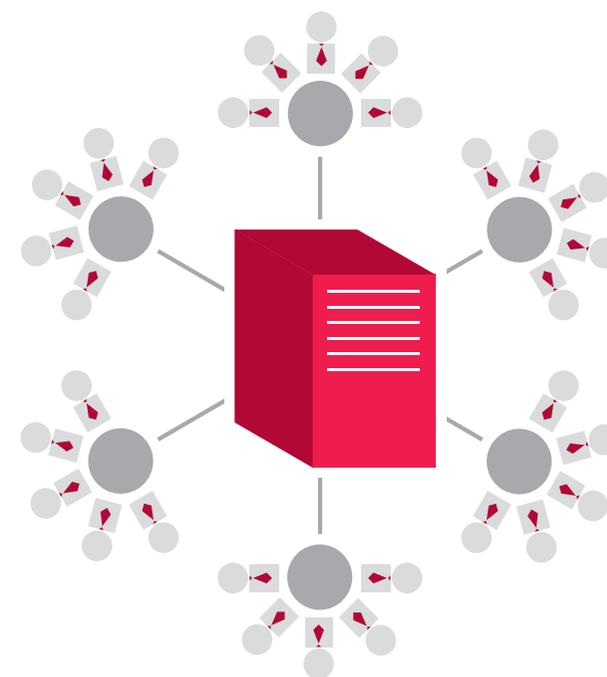
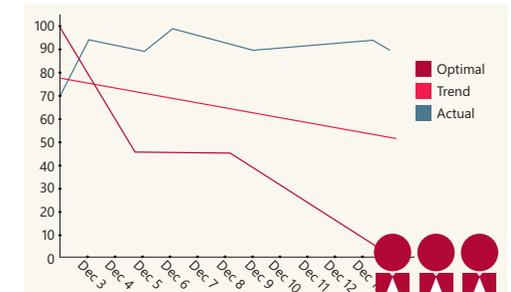
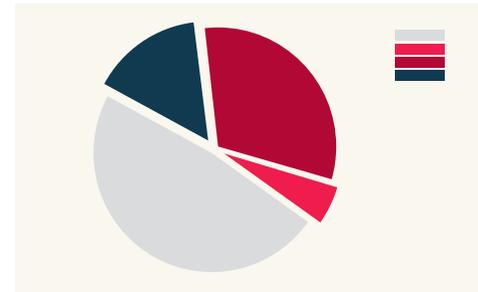
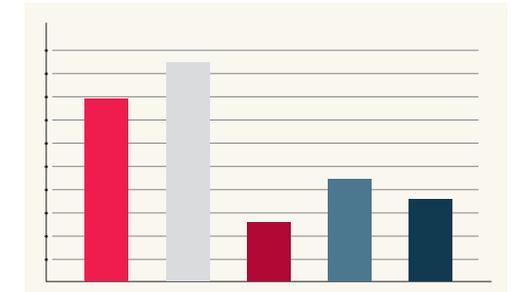
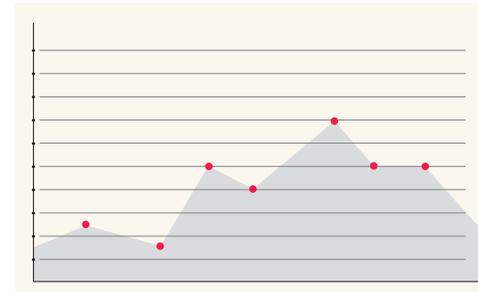


Figure 3B: Distributed team with centralized system

Large Teams and Tools

As team size grows, so does the complexity and need for communication and collaboration. Many Scrum teams handle this type of scale by having a “Scrum of Scrums.” In this model, large teams are segmented into smaller teams (with geographic affinity) of no more than 7 or 8. Each team will participate in their typical “daily scrum” and then a representative from each of the teams will get together to have a “scrum of scrums” to help synchronize current state across teams. Sounds reasonable – and in fact, it is. How does this work with 100’s of people however – how about 1000’s? How can we help understand how the challenges of one team might impact the results of another? Some Agilists might suggest that we should never have teams that are very large – however, sometimes the nature of the project or initiative demands this. Managing large teams without tools that facilitate collaboration and progress is unfathomable.



Tools help manage the complexity of managing and collaborating between members of a large team. Agile tools help to manage this complexity by making sure that each user can find and work with the information that matters to them, their team, or their project while allowing them to change this scope effortlessly. Agile tools can also help to visualize complex relationships and flows that may be difficult or impossible to do otherwise. One of the key workflows on any Agile team is planning. On an Agile team planning is something that happens continually throughout the project. Agile planning revolves around deep and meaningful collaboration and as a team grows the need for continual collaboration also grows, and so does the need for tooling to support the constant collaboration required.

Measurement and Tools

Gathering metrics is a core practice of an Agile project – in fact, any form of project. Here are some of the underlying metrics every agile team should follow:

- Team velocity (the rate of work completion) and other metrics such as release rates are an important part of any Agile team as they provide a foundation to both a shared understanding of progress and process improvement initiatives.
- Backlog burndown rate will give insight into your team's ability to ship within expectations.
- Bug rate can help you see the impact of your quality process improvement activities over time.

Small Agile teams may be able to produce these metrics by hand – however, as you scale agile the need to have a tool provide you and your team with these metrics by automatically measuring your team's activities becomes not only a huge time saver, but extremely important if you want to consistently gather metrics across many teams.

Agile management tools help by automatically providing metrics based on the day-to-day activities of the teams that use them, saving time and effort manually putting those metrics together – a need that becomes extremely important as organizations scale Agile. Tools will also help to ensure that different teams will gather and represent metrics in a consistent manner – bringing even more value to the metrics from an organizational perspective. Advanced Agile tools will also help teams interpret and act upon those metrics and provide alerts and guidance in real time if it detects issues, allowing the team to focus their time and effort on what matters most – delivering customer value.

Best Practices and Tools



Foundational to any Agile team is its ability to learn ways to learn ways to improve, and then implement those learnings continually – gradually and incrementally and incrementally and incrementally removing waste from the development process in order to produce a greater amount of value in the same amount of time. Teams facilitate this process by having regular retrospectives to talk about how to reinforce what they are doing well and eliminate the practices that are not working and share knowledge and practices that make the team more efficient. For example, teams following the Scrum methodology have regular Retrospective Meetings at the end of every Sprint to ensure there is time allocated to improve practices during the project. Teams are only human, however, and sometimes fall back into old habits. Organizations are always looking for ways of spreading best practices across teams however, the broader the need to scale these practices, the harder this challenge becomes.

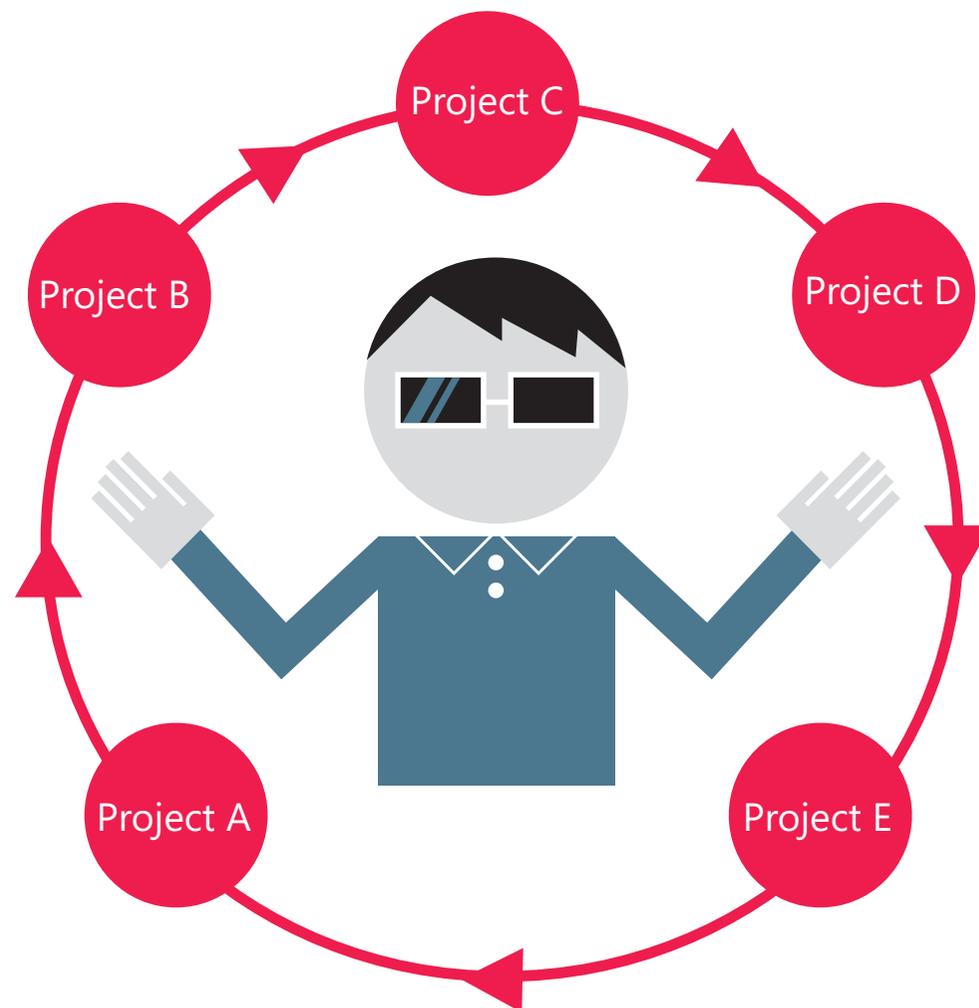
Tools become an excellent method for teams to capture their best practices and to help continually reinforce and repeat those best practices even as team members change or the team grows. More importantly, tools can make it possible to capture learning from one team and distribute it consistently to other teams within the organization, helping to scale Agile learning consistently and effectively. Really great tools will remind teams of when practices aren't being followed and educate them on the impacts.

Agile methodologies and practices such as Scrum and Kanban are actually collections of best practices. Just as it is important for tools to support the best practices of a team, a tool should also be able to directly support predefined Agile methodologies and practices. Tools that are able to support an Agile methodology out of the box give teams a significant head start in both the adoption of those practices as well as the adaptation of those practices to fit their specific needs.

Multiple Projects and Tools

As organizations scale Agile to larger and increasingly distributed teams, there is usually also a need to support teams working on multiple projects simultaneously. In many organizations it is very common to have teams work on multiple projects at the same time. Take, for example, an operations and deployment team who may need to work with multiple concurrent projects to deploy and update software. In addition to making it easy for teams to work on multiple projects at the same time, organizations often want to see the overall progress and status of multiple projects at the same time, a request commonly asked of Project Management Offices or Development Managers. In these situations tools once again are invaluable.

Tools that are effective at supporting multi-project interactions and perspective must be built from the ground up to support these scenarios as not all tools are well suited to this purpose. At minimum, these tools must allow users to view schedule and work assignment information from multiple projects on a single view. In addition, tools that provide PMO's and development managers their required perspectives must be able to display the status of many concurrent projects at the same time. Once in place, however, these tools become irreplaceable.



Summary – You Need a Tool to Scale Agile

Developers use tools to be more “agile” all of the time. They use automated build and deployment tools, they use automated unit testing tools, they use code generation tools, and they even use tools to help generate their test driven development code structures from their requirements. Using tooling to help a team be more agile is not uncommon and is in fact a requirement for most. Using a tool to help scale Agile, in whatever your definition of Agile Scale means to you, makes perfect sense as well-designed tools help make difficult activities easier and achievable.

As we scale Agile we run into problems such as communication, consistency, measurement and use of best practices – problems that can only truly be solved through the use of tools built to specifically solve these problems. More important to simply having tools we can assemble to fix individual issues is to have an integrated suite of tools that can work together across a number of different scale problems to provide a solution that is greater than the sum of its parts.

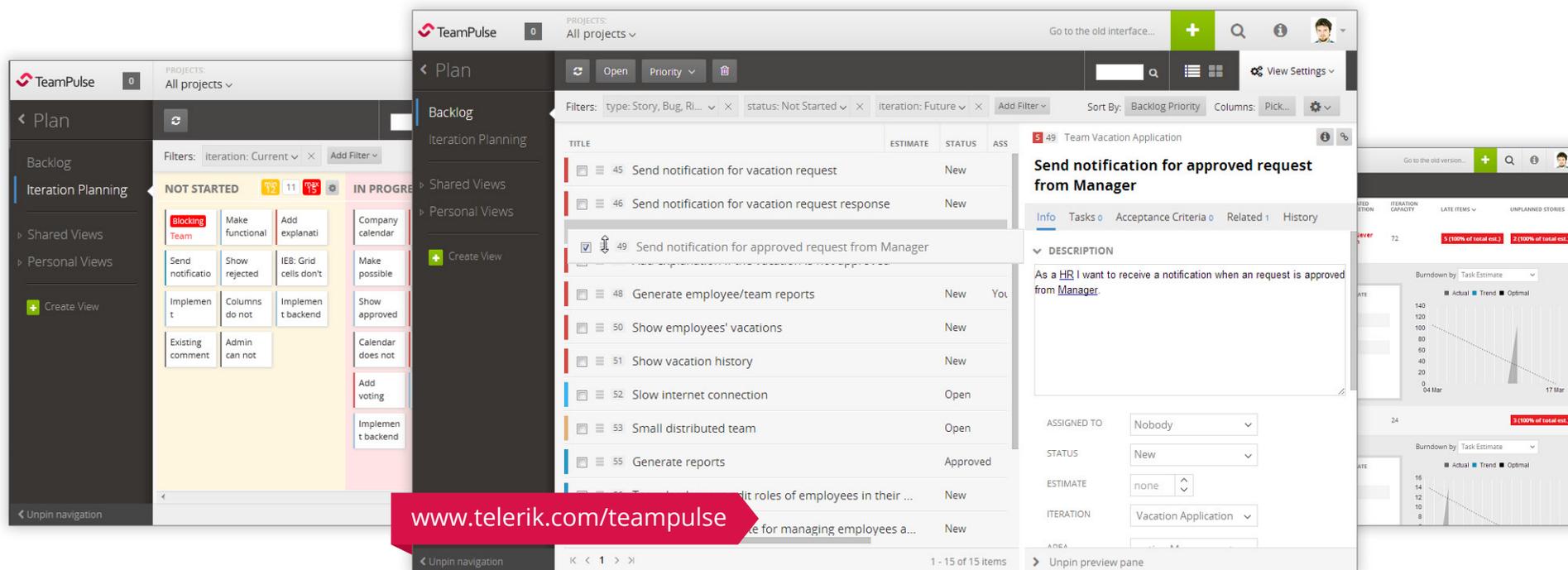
In order to scale Agile, you need to start with a very clear vision of what scale means to your organization and what value you will reach for as a result. Simply applying a tool to the Agile scaling problem will end in disaster. Using a tool to support your Agile scaling strategy is virtually mandatory.

BROUGHT TO YOU BY



TELERIK TeamPulse

TeamPulse is an all-in-one agile project management system. TeamPulse helps teams manage requirements & bugs, plan releases and track progress while keeping the entire team constantly connected.



The screenshot displays the Telerik TeamPulse interface. The main view is a backlog with a table of tasks. The task details panel is open for item 49, 'Send notification for approved request from Manager'. The burndown chart shows the team's progress against the total estimate.

TITLE	ESTIMATE	STATUS	ASS
45 Send notification for vacation request		New	
46 Send notification for vacation request response		New	
49 Send notification for approved request from Manager		New	
48 Generate employee/team reports		New	You
50 Show employees' vacations		New	
51 Show vacation history		New	
52 Slow internet connection		Open	
53 Small distributed team		Open	
55 Generate reports		Approved	

Task 49 details:

- ASSIGNED TO: Nobody
- STATUS: New
- ESTIMATE: none
- ITERATION: Vacation Application

Burndown chart data:

DATE	Actual	Trend	Optimal
04 Mar	140	140	140
17 Mar	0	0	0

www.telerik.com/teampulse

About the Author



Joel Semeniuk is a founder of Imagnet Resources Corp., a Canadian based Microsoft Gold Partner. He is also a Microsoft Regional Director and MVP Microsoft ALM and has a degree in Computer Science. With over 18 years of experience, Joel specializes in helping organizations around the world realize their potential through maturing their software development and information technology practices. Joel is passionate about Application Lifecycle Management tooling, techniques, and mindsets and regularly speaks at conferences around the world on a wide range of ALM topics. Joel is also the co-author of “Managing Projects with Microsoft Visual Studio Team System” published by Microsoft Press, as well as dozens of other articles for popular trade magazines on agile adoption, involvement and collaboration between roles.

Follow him on Twitter [@JoelSemeniuk](https://twitter.com/JoelSemeniuk)

References cited

<http://scaledagileframework.com/>

<http://public.dhe.ibm.com/common/ssi/ecm/en/raw14261usen/RAW14261USEN.PDF>

A publication of  **telerik**