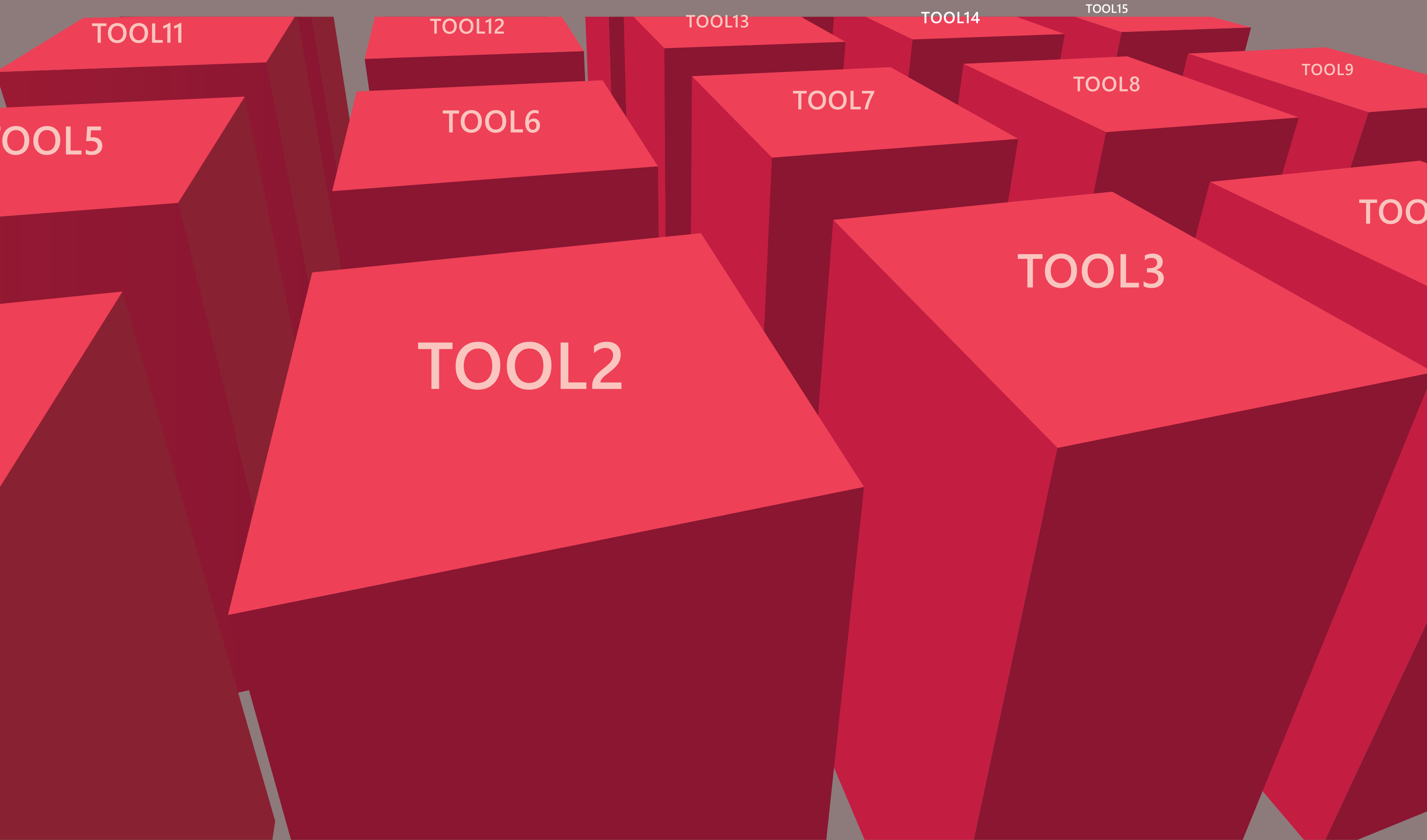
# 7 MUST-HAVE FEATURES

## When Choosing an Agile Management Tool



## Why Do You Need Tools?

Tooling to aid in the definition, construction, and management of software has been around from the very beginning of software development. Agile management practices have made a significant impact across the entire landscape of application development. In addition to Agile's influences, software construction and team management techniques continue to evolve at a rapid pace due to the quickly changing platform landscape, new engineering tools and patterns, and modern ways to capture customer value.

Increasingly, in order to support modern software delivery, complex infrastructure is required to facilitate aspects of the application development landscape – from the maintenance of a source code repository to the management of the build and deployment infrastructure required to validate pre-production software. Tools are also typically employed to help capture and manage requirements, test cases and service requests. Additionally, tools assist collaboration and the flow of context between customers and team members. Combined with the necessity to support a distributed workforce and the need to continually focus on process improvement through the capture and analysis of project metrics, tools are becoming essential in helping teams manage complexity and control project chaos.

> *"In response to the increasing criticality of software within systems and the increasing demands being put onto 21st century systems, systems and software engineering processes will evolve significantly over the next two decades."*
>
> Barry Boehm
>
> "Some Future Trends and Implications for Systems and Software Engineering Processes", Center for Software Engineering, University of Southern California, Los Angeles, CA 90089-0781

## Too Many Options?

The landscape of processes related to software development is quite vast – from ideation, requirements management, planning, tracking, engineering, testing, and validation to deployment, maintenance, and help desk/customer support. There is an overabundance of tools on the market today. Some of these tools span across various landscape areas of development processes where some tool choices only target specific areas. In fact, in a report published in 2011 Gartner states that:

> *"The application development software market will grow modestly through 2015, when it will reach $10.1 billion in revenue. New applications, consumption models, devices and developers are driving evolution of the AD space."*
>
> Gartner
>
> Market Trends: Application Development Software, Worldwide, 2011-2015

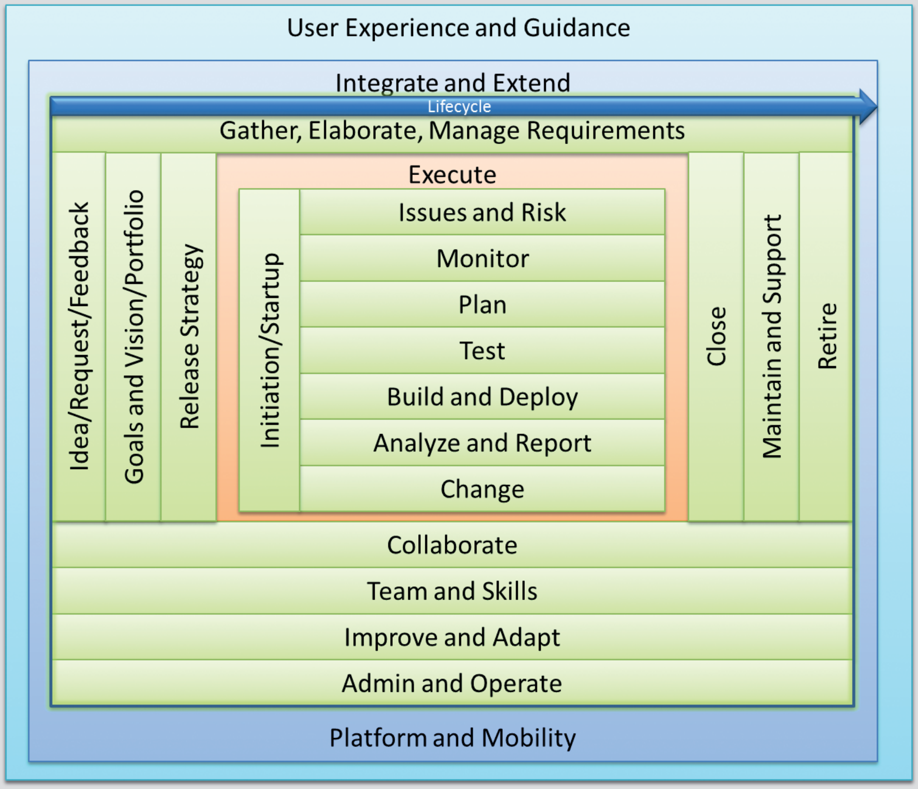


Figure 1 - A representation of the Application Development Landscape (ADL)

Figure 1 shows the sheer number of perspectives of the application development landscape (ADL). In addition to the number of commercial options available on the market targeted to reign in some or all of this complexity, many organizations have produced homegrown solutions that address their specific needs of the ADL. Organizations that have created homegrown solutions must decide if continuing to use their development resources on creating and maintaining these tools is in their best interest.

This paper suggests 7 of the most important characteristics that organizations should consider when investing in tooling that supports their Agile management practices and teams.

# 7 Must-have Features When Choosing Tools

## 1. Designed for Purpose and Usability

The software industry has proven that tools that are focused on purpose and flow are better accepted than generic tooling that can be modified to fit a purpose. Modern software places the user experience at the heart of the user interaction, ensuring that the software directly supports the needs of the user performing work to achieve well defined goals. Software that is written in this manner is much easier to adopt and has more rapid and deeper penetration within an organization and ultimately provides greater value in less time.

Usability plays a large role in the success of tooling in any organization. Software is considered to introduce "friction" when it is difficult to use and does not smoothly support the way users work. Friction means the user may resist the usage of the tool.

> *"If the user can't use it, it doesn't work."* ©
>
> Susan Dray
>
> http://www.dray.com/index.html

Agile management tools are built to support both team management and software engineering processes. It's important to point out that not all users of these systems are software developers. Developers may value tasks and requirements over times and portfolios. Project managers may value status, impediments and flow over individual requirement and implementation details. Testers may value test plans, test results and bug metrics over project plans and development detail. Stakeholders value scope, change, and delivery dates. Tools should be built to provide value to all members of the development team and must ensure to specifically reflect those needs from a usability standpoint. Tools should also guarantee a high level of adoption, involvement and collaboration between roles.

Tools should be built to provide value to all members of the development team and must ensure to specifically reflect those needs from a usability standpoint. Tools should also guarantee a high level of adoption, involvement and collaboration between roles.

> *"It is far better to adapt the technology to the user than to force the user to adapt to the technology."*
>
> Larry Marine

## 2. Accessible Data

Tools chosen to help support your Agile management practices, store information vital for the project – from the requirements to team and project metrics. Successful organizations are built around interconnected software and it's important that data that resides in all of your tools be accessible programmatically to other systems. This will allow your organization to take advantage of valuable data in ways that you might not be able to predict today. For example, your organization may want to include project metrics on a corporate-wide dashboard, or synchronize time-tracking data into a billing or accounting system. Ensuring your tooling investment supports data access standards is critical to this form of reuse.

In addition, all of the data in your Agile management tools should be accessible via an open application programming interface (API), such as REST (REpresentational State Transfer), or a RESTful based API such as the Open Data Protocol (oData), to allow integration and extension by other systems. This model will help ensure that the valuable data stored within these systems is immediately accessible for integration into other business processes and systems without the need for proprietary tooling or integration components.

Ensuring your tooling investment supports data access standards is critical.

> *"The growth in information volume, velocity, variety and complexity, and the increased importance of information to the business makes the discipline of information management radically different from the past. Old techniques are no longer adequate; a dramatically different approach is needed."*
>
> Gartner
>
> "Information Management in the 21st Century"

## 3. Promote Modular Adoption

Organizations adopt Agile management practices differently, not only in their rate of adoption, but the methodology (or methodologies) chosen. A single software management tool cannot fit every organization's chosen manner and speed of agile adoption. At the same time, organizations should not throw away existing investments in working, utilized solutions just to be replaced with a new catch-all system.

Management tools must have "pluggable" modules that can be pulled-in based on an organization's unique needs. Modules should fit together like "LEGO®" blocks, and when combined with open data access standards across all tools, provide a path for incremental adoption and value.

For example, your organization may want to focus on a source control repository early in the Agile maturation cycle versus time tracking or portfolio management solutions. You must be able to adopt an incremental approach to software selection by starting with a system that best suits your current need(s). As your needs expand, you will then be free to adopt additional integrated components as they are required, at the time they are required. This allows for adding functionality such as requirements management, bug tracking, or feedback and reporting in an incremental and agile approach.

> ❗ Management tools must have "pluggable" modules that can be pulled-in based on an organization's unique needs.

## 4. Social and Rewarding

Tooling should inspire effective and correct use and behaviors. Tooling options that simply gather data and then report on it are being replaced by software that is more social and rewarding to help reinforce good use and behavior. When this behavior is modeled after the culture of your organization, users are more likely to comply with the organization's needs and goals.

Modern management tooling needs to be more than forms over data – it needs to act as a coach and a mentor. It needs to be aware of how teams are using it and should help them be accurate and consistent, as well as guide good behavior.

All forms of tooling that support your Agile management practices should guide your team – helping them ensure they are working consistently towards the correct goals as well as providing guidance on employing new techniques and practices to gain additional operational efficiencies.

> *"The online experience has become increasingly social. To profit from this industry wide transformation, organizations need to understand the principles of social experience design."*
>
> Gartner
>
> Social Experience Design Is Becoming a New Imperative, June 2011

> ❗ All forms of tooling that support your Agile management practices should guide your team – helping them ensure they are working consistently towards the correct goals as well as providing guidance on employing new techniques and practices to gain additional operational efficiencies.

## 5. Reach

Today's landscape of technology and platform choices is as diverse as it has ever been. Any investment into tooling to support your organization should take into account a wide range of devices and operating environments. Your tools should allow you to work where you work. Due to the dramatic impact mobile computing has had on the software and computer industry, tools should be written to allow for ease of use on PC's, tablets, as well as mobile devices.

This is not to say that all features of your tool suite should run on all devices equally – as different device form-factors impact the practical functionary present. For example, tablet interfaces may be exceptional at presenting information graphically, leveraging touch to manipulate data items. Tablets, however, are not well suited for data entry or complex application manipulation. Look for software that can support the most the appropriate workflows for the various platforms and devices your team uses.

> *"Consumers have a choice of multiple technology products to perform different online tasks and expect to use them across similar content and applications."*
>
> Gartner
>
> Market Trends: Consumer Technology Adoption Trends, 2012, January 2012

❗Look for software that can support the most the appropriate workflows for the various platforms and devices your team uses.

## 6. Facilitate Communication and Collaboration

Agile teams rely upon constant and meaningful collaboration and communication between team members and with their customers. Constant communication and collaboration between everyone involved in the software development process helps to eliminate waste by ensuring that the entire development team always has the correct and most up to date context of the project and the needs being satisfied.

❗Look for Agile management tools that make it easy for customers to give feedback and offer suggestions to the project team in an easy and effective manner, there by helping to remove barriers to collaboration.

Customer collaboration is a corner stone of modern Agile management practices. Traditional models of software development work to create barriers between the people that will use the application and those that are designing and developing it. In this model, requirements are "handed off" to the development team, who handoff the resulting software back to customers. Agile based methods of software development ensure that the customer is involved continually through the entire lifecycle of development, minimizing the handoffs and the opportunity for expensive misunderstandings.

Look for Agile management tools that make it easy for customers to give feedback and offer suggestions to the project team in an easy and effective manner, there by helping to remove barriers to collaboration. This feedback should then flow naturally through the management so that it can be triaged and evaluated by the project team – ultimately sparking constant and meaningful conversations and collaboration between the team and the customer.

In addition, Agile management tools should facilitate constant and rich collaboration among the project team members. No tool can or should replace face-to-face team collaboration; however, tools should be able to initiate these conversations and help make sure that correct context is surfaced to all team members when required. Tools of this nature inspire collaboration versus hand-offs, making it easy for context to flow naturally throughout the development, testing, and deployment processes.

## 7. Project Status Transparency

One of the biggest benefits of the Agile movement is increased transparency of a project, not only between team members, but also between the core development team and stakeholders and other interested parties. Burndown charts, daily standups, and all of the other visibility mechanisms are in place to increase transparency. Yet these processes have a proximity limit. The increased power of mobility, such as tablets, slates, and smart phones, has enabled everyone to stay connected to what is most important to them, regardless of their physical location.

> ❗ Agile management software must enable all interested parties to stay connected. More important than just staying connected, the software should surface all of the real-time information needed for each project in a way that can be interpreted at a single glance.

Agile management software must enable all interested parties to stay connected. More important than just staying connected, the software should surface all of the real-time information needed for each project in a way that can be interpreted at a single glance. This allows for status to be communicated without having to walk through the project area to see the charts, read through a host of status reports, or phone the project manager to get the latest updates. Teams and management needs to be able to view any project in their portfolio on a single screen, or any device they chose.

The goal of transparency is provide a foundation of accountability as well as to provide an environment by which waste (in the form of issues, risks, and problems) can be detected and dealt with as quick as possible. Tools that support your Agile management practices should make is virtually impossible not to be transparent by ensuring that they surface the right information at the right time with virtually zero friction or additional effort by the team.

# Summary

There are many factors to consider when evaluating options for managing the Application Development Landscape. First and foremost, the tooling must meet the needs of your organization. However, there are additional factors to consider beyond a simple check list of features. When looking for tooling to help manage your software development teams, look for systems that have the following characteristics:

1. Designed for Purpose: Focus on usability and the embrace of non-developer roles
2. Accessible Data: Provide the means of data access from outside systems
3. Modular Adoption: Ability to increment adoption of tooling according to need
4. Social and Rewarding: Provide meaningful guidance to help reinforce good practice
5. Reach: Embrace multiple device and operating system access
6. Facilitate Communication: Enable feedback, ideas, and other communication between customers and the project team
7. Project Status Transparency: Provide all of the information for the project at a single glance