



# Launching Agile and DevOps in a 100 Year Old Company

DILIP BALACHANDRAN, USG  
MICHAEL KOVAS, VENKATA KUMARAN, ZACHARY DAWSON, USG

---

This paper discusses our successful implementation of agile and DevOps methodologies in a 100 year old company with minimal funding, using primarily internal resources. The paper highlights the key challenges, learnings and success stories while providing an overview of our high level approach, custom agile methodology and DevOps architecture.

---

## 1. INTRODUCTION

Even as a new employee at USG three years ago, I could feel the winds of change. There was a palpable sense of urgency across the organization to be more nimble, agile and efficient. IT was not immune to this trend and we were striving to shorten our evolutionary cycle such that we could increase our speed to market without jeopardizing our track record of flawless operations. The initial thought was to bring in an external consulting firm to help us with this transformation, but we ran into some severe funding constraints which meant that external help was out of reach. It was at this juncture that I was assigned to lead the effort to launch agile and DevOps at USG, along with my co-authors who formed the core leadership nucleus of our team.

## 2. BACKGROUND

USG is one of the largest manufacturer and distributor of building products in North America. After successfully navigating the 2008 financial crisis, the company continues to face a volatile, uncertain, complex and ambiguous world where its customers are consolidating and markets are shifting faster and more frequently than ever before. The company is also working on optimizing its capital structure by making some large debt payments that are coming due in the next couple of years. However, to maintain its leadership position in the industry, it is paramount that the company has strategic agility and ability to quickly adapt to changing market conditions; and as digital capabilities permeate multiple aspects of USG's business, our customers and executives expect IT to keep pace with business and help accelerate the company's ability to respond to market changes.

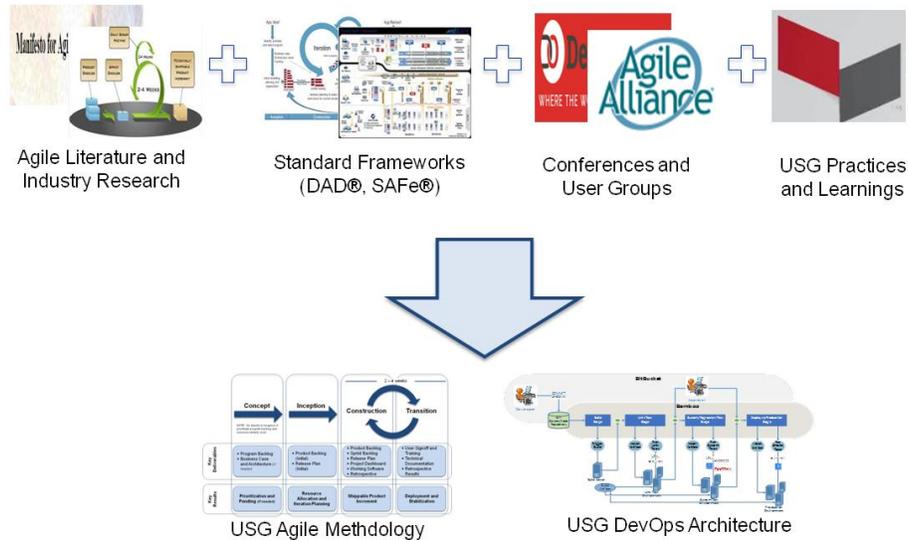
## 3. OUR AGILE / DEVOPS CHARTER AND APPROACH

In order to be more agile in terms of its IT assets, USG needed to make some fundamental changes to the way we delivered IT projects. While we valued the predictability and reliability of our core systems, we also understood we needed to make some transformative changes to address concerns around speed to market and responsiveness. The company's focus on capital restructuring meant we didn't have much in terms of funding, so we had to improvise and find a way to make this transformation happen with minimal funds. To achieve our objectives, we first categorized our applications into systems of record, systems of differentiation and systems of innovation based on Gartner's pace layering model [Gaughan]. Then we decided to adopt a bi-modal approach [Mesaglio] where systems of record would continue to be delivered using a waterfall-like iterative methodology (mode 1) and systems of differentiation and innovation would be delivered using a new agile methodology (mode 2). Next, we formed a project team (consisting of employees allocated part-time to this effort) chartered with coming up with an approach to launch agile methodology and DevOps at USG. The team

---

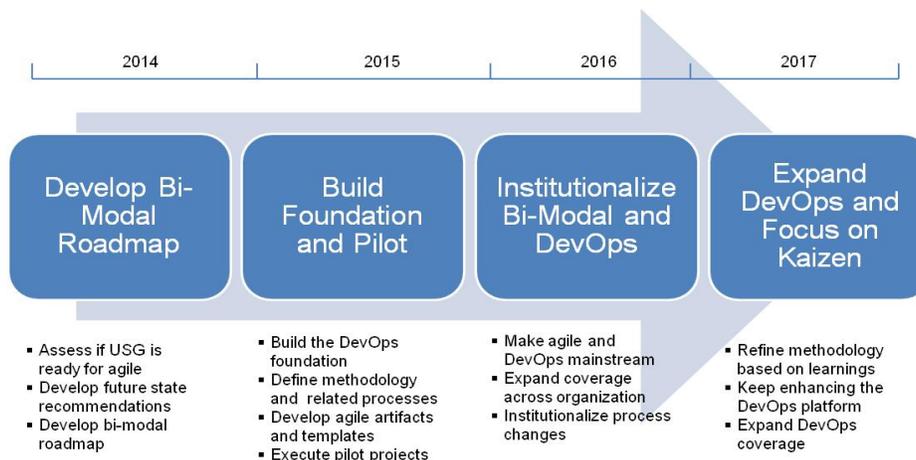
Authors addresses: Dilip Balachandran, email: [dbalachandran@usg.com](mailto:dbalachandran@usg.com); Michael Kovas, email: [mkovas@usg.com](mailto:mkovas@usg.com); Venkata Kumaran, email: [vkumaran@usg.com](mailto:vkumaran@usg.com); Zachary Dawson: [zdawson@usg.com](mailto:zdawson@usg.com)  
Copyright 2016 is held by the authors.

was a mix of some long tenured employees with strong business relationships and some newer employees with prior experience in using agile methodologies. This team structure was intentional and we only selected people who were passionate about bringing about this change in the company. The high level approach adopted by this team to launch agile and DevOps is illustrated below.



**Figure 1: High Level Approach**

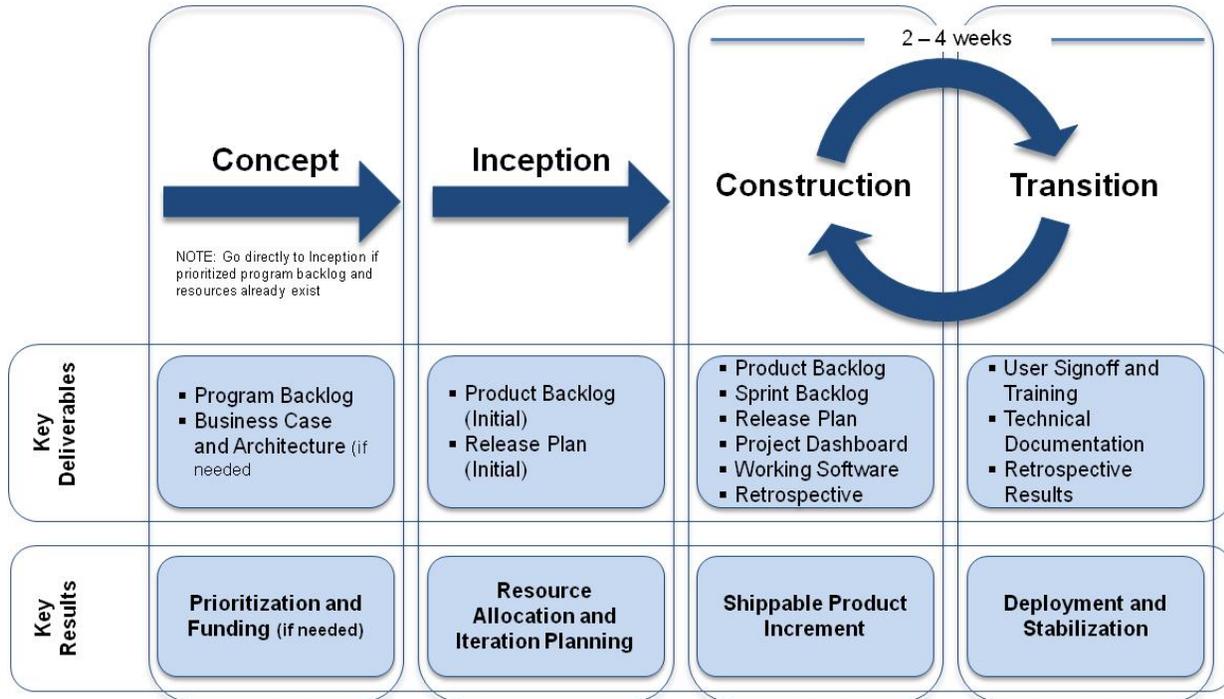
As can be seen above, our primary approach involved leveraging external resources, standard frameworks [Leffingwell][Ambler] and other industry research to develop our knowledge and coupling it with USG’s unique culture and practices to come with a customized agile methodology and DevOps architecture that would work for us. We also came with a high level timeline (shown below) to start managing expectations and aligning with the broader team in terms of change management.



**Figure 2: High Level Timeline**

### 3.1 USG Agile Methodology

We quickly developed an initial version of our agile methodology and piloted it on a couple of projects before finalizing our agile methodology shown below.



**Figure 3: USG Agile Methodology**

As shown above, the USG Agile Methodology consists of four phases:

- **Concept:** This phase covers the ideation/business case, high level architecture and funding aspects of a project. We still operate under the traditional model in terms of budgeting and portfolio management and this phase facilitates seamless interactions with those processes as and when needed. For an application platform where captive resources are available and product owner is the final authority on business priorities, this phase can be skipped.
- **Inception:** This phase covers most of the project initiation activities. Ramping up the team (if and as needed) and coming up with an initial product backlog and release plan are some of the key outcomes from this phase. Product backlog contains a prioritized list of the features needed to develop a successful product and release plan lays out a high level timeline for releasing the main features into production.
- **Construction:** During the construction phase, the project team produces a potentially shippable product/solution on an incremental basis through a set of iterations called sprints. We developed our process based on Scrum methodology [Rubin], so we have adopted practices and ceremonies associated with Scrum [James]. Product backlog and release plans continue to be groomed and updated during this phase.
- **Transition:** This phase is used to deploy code into production and stabilize the solution if/as needed. We also complete our technical documentation, training and operational readiness related activities during this phase. Product backlog and release plan are also updated as needed.

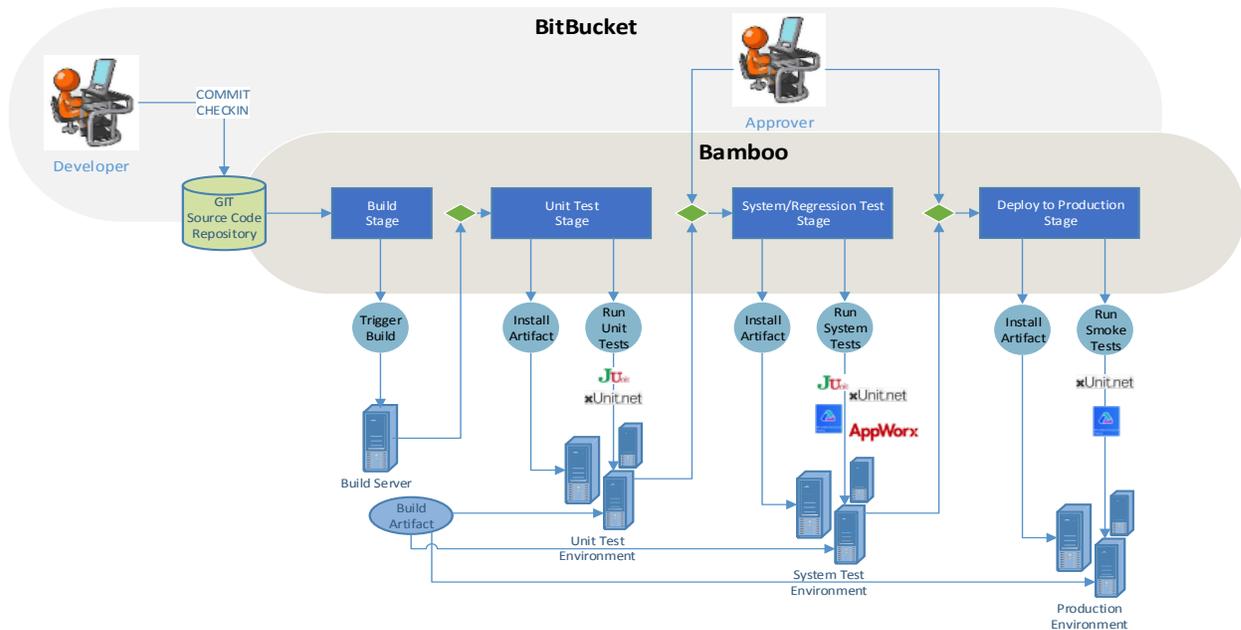
### 3.2 USG DevOps Vision and Architecture

Based on our interactions with multiple parties, it was clear to us from the outset that we needed to develop our DevOps capability to achieve the kind of strategic agility we were looking for. We chose to launch it simultaneously with our agile methodology and had a small team of employees focused on developing our DevOps footprint. As you might already know, DevOps acknowledges the interdependence of software

development, quality assurance and IT operations, and aims to help an organization rapidly produce software products while improving operations performance. We had a similar vision for our DevOps initiative and our goal was to enable continuous deployment and continuous integration for platforms that had been selected for agile development.

We chose JIRA as a platform to enable our agile methodology and related processes. So, we needed to choose a DevOps platform that would integrate easily with JIRA, while still being cost effective and easy to use. We decided to go with Bitbucket and Bamboo as they were easy to use and integrated seamlessly with the rest of our platform. They also provided us the best opportunity to scale our DevOps platform iteratively in a cost effective manner.

The figure below shows our high level DevOps architecture and how the various tools integrate to enable our vision of continuous deployment and continuous integration. As seen in the figure, there are three tollgates – the first one is after a build (where applicable) to ensure that the code is actually in a state that it can be deployed; the second tollgate is an approval for test deploy after successful unit and functional testing; and the third tollgate is approval for production deploy once all testing and validation has been completed.



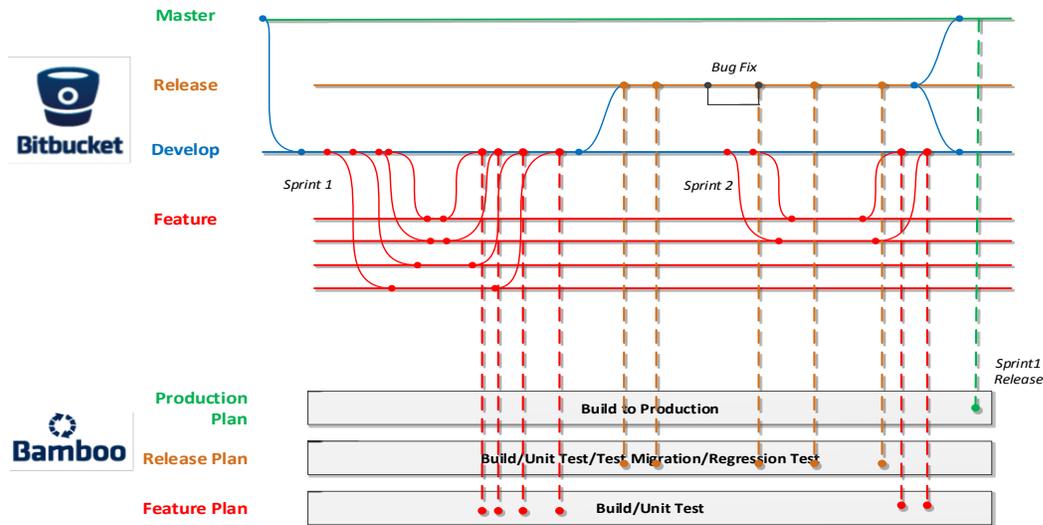
**Figure 4: DevOps Architecture**

To enable our core capabilities, we first needed a robust way to handle our code repositories. After some research and due diligence, we zeroed in on GitFlow as it allowed us to:

- Use an industry standard that most employees understood
- Enable our plans to integrate Bamboo such that we could build off of different branches and
- Allow for multiple streams of work to operate in parallel.

We developed a five branch system as illustrated in the figure below. The branches were:

- **Master** branch which enabled production deployment and was a true copy of production code
- **Release** branch which enabled test deployment and regression testing
- **Develop** branch which acted as an integration point for our *feature* branches (described below) to allow for resolution of any conflicts
- **Feature** branches assigned to developers to work on their code. This branch would also kick off any builds or unit tests associated with the product
- **Bug Fix** branch was created to fix bugs which had made it into production and deploy these fixes quickly without having to go through unnecessary tollgates



**Figure 5: Branch Structure**

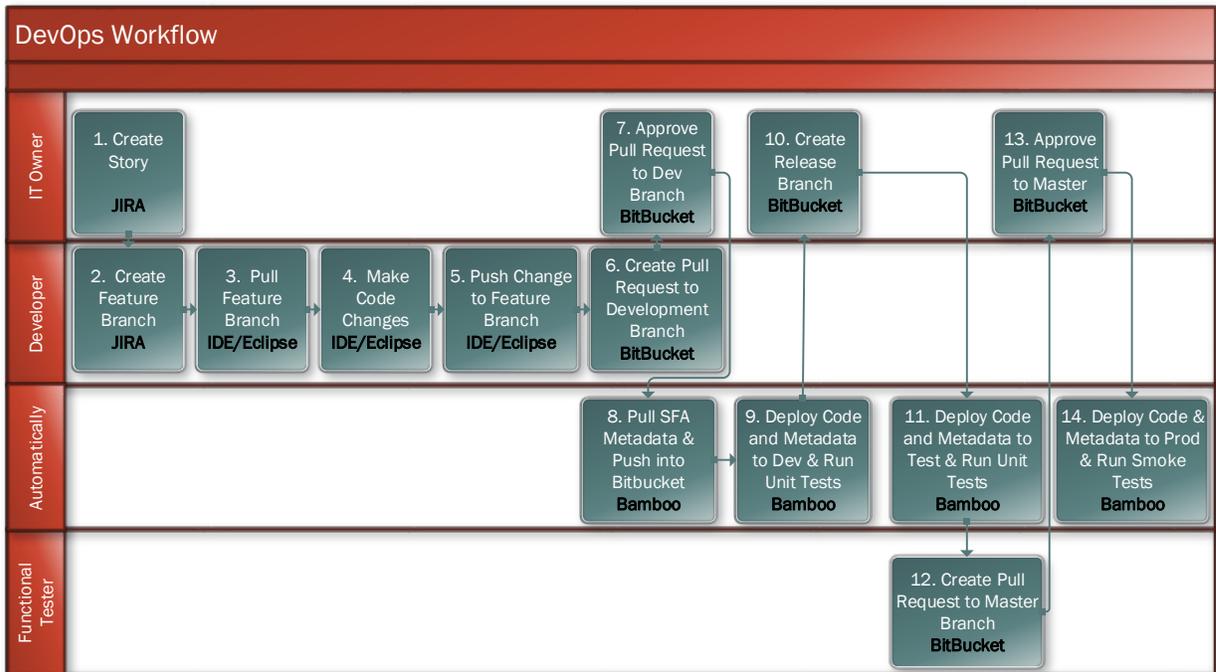
### 3.3 DevOps in Action

Our Salesforce.com footprint was one of the first technology platforms that we brought under the DevOps umbrella. Being a software-as-a-service solution which was transforming the way we do our sales, that seemed like an ideal pilot candidate for DevOps because of three reasons:

- Our sales team needed the capability to respond quickly to market changes and there was a pressing demand to accelerate our speed to market from an IT application standpoint
- Salesforce.com had great documentation on enabling DevOps for their platform
- We had unwavering support from executive leadership on both IT and business side

Prior to the DevOps initiative, our benchmarks indicated that the cycle time from test to production deployment was anywhere between one to three months. We also found that the application team was spending a lot of time packaging their code for deployments and the operations team was spending a lot of their time cross-checking and validating things, testing changes and committing them to production. We also found that we didn't have a good way to rollback changes if we wanted to.

We first tackled the challenges around versioning, packaging and rollback by implementing Bitbucket and using that to manage our code base for salesforce.com. We then leveraged ANT and Bamboo to automate our deployments and integrated it with JIRA for change control and governance. We also automated our main regression test scripts and integrated it with the rest of the DevOps stack so it could be triggered automatically as part of a new deployment request. The system would send out alerts and notifications as needed using e-mails. The high level DevOps workflow for our Salesforce.com implementation is shown below.



**Figure 6: High Level DevOps Workflow for Salesforce.com**

The new DevOps process reduced cycle time from test to production deployment to two days and freed up a lot of time for our applications and operations team to focus on more value added activities.

### 3.4 Results

Even though we faced some challenges initially, we were very successful in both launching and expanding the coverage of agile and DevOps across our application footprint. Some of our main successes are highlighted below:

**Agile Projects** – We have successfully completed over ten projects, including some large projects like Salesforce.com implementation, using our agile methodology. More than 50% of our current portfolio of projects is being run using the agile methodology we developed. The quote below from one of our Sales SVPs illustrates the success we have had with agile methodology on our projects:

“ I could have never envisioned what we came up with. The new agile process enabled us to create one of the greatest solutions we have seen in a while for national accounts”

**Training and On-Boarding** – We have provided agile training internally to around 100 employees and we have established a standard process to on-board new projects on to USG’s agile methodology. We also have helped ten employees transition to scrum master and product owner roles with at least half of them developing enough expertise to coach others.

**DevOps Success** – We have successfully deployed DevOps related capabilities to four technology platforms and we are actively working on building similar capabilities for six more platforms by the end of this year. We have a repeatable template that can be used to bring more technology platforms under the DevOps umbrella in future. On average, we reduced the cycle time between test and production deploys from two months to two days for technology platforms which are under the DevOps umbrella.

#### 4. WHAT WE LEARNED

We overcame many challenges over the course of our agile and DevOps journey. Some of the methods we used to overcome these challenges and our key learnings are highlighted below:

**Executive Support** – We learned very quickly that we couldn't get very far on this journey without continued and visible executive support. We needed our staff to go about their day differently, follow new processes and use a new set of tools. On top of that, the effort would be successful only if this was adopted cross-functionally across the organization. There would be too much organizational resistance to such a change without continued demonstration of executive support. We used forums like our all hands meeting, symposium (USG internal conference) and periodic messages from key executives citing recent success stories to keep agile and DevOps top of mind and re-inforce ongoing executive support and sponsorship.

**Part-time Resources and Agile Week** – Another challenge we ran into was that our core team chartered with launching agile and DevOps were only available part-time for this effort. Our milestones started lagging behind as most of them were constantly getting pulled into activities related to their other projects and it was difficult to maintain the collaboration and focus needed to bring about a transformative change. The team's efficiency was below par overall because of high context switching costs. To overcome this challenge, we introduced the concept of "Agile Week". This was one week in a month where all our core resources would be co-located in a large conference room to focus on deliverables and activities related to launching agile and DevOps. They were generally unavailable for other activities to minimize context switching costs. This was also the time when we would pull in extended team members and subject matter experts to help with specific activities. This was a tremendous success and a large part of what we accomplished would not have been possible without our "Agile Weeks". As part of this exercise, we also learned that we can make agile successful with part-time resources as long as we keep minimize personnel turnover and actively manage our context switching costs.

**Training and Change Management** – Due to our funding constraints, we had to get creative in how we trained our people on the new methodology and related tools & processes. We adopted a "train the trainer" approach and used a combination of formal trainings, conferences, research and self-study to develop in-house expertise within the core team. We then partnered with our internal teams to develop training content and deliver it to targeted audiences. We developed three internal courses – one focused on tools, one focused on the process and methodology, and one focused on leadership. We put a lot of focus initially on developing internal coaches who could help on-board project teams new to agile. This was especially important as Scrum Master and Product Owner roles were new for our organization. We also developed a simple toolkit designed to help new projects choose a mode (agile or traditional) based on a standard set of criteria. Throughout this transition, we partnered with our internal change management team to plan and execute activities related to organizational readiness. This included awareness campaigns, executive briefings, newsletters, trainings and a formal project on-boarding plan.

#### 5. ACKNOWLEDGEMENTS

We thank all our colleagues at USG who made significant contributions to successfully launch agile and DevOps methodologies at USG. We also thank our senior leadership team for their commitment and unwavering support as we undertook this journey. Finally, a big thank you to all practitioners out there who have taken the time to document their learnings and shared their knowledge and experience through blogs, websites, books, conferences and/or user groups – *we couldn't have done it without you!*

#### REFERENCES

Leffingwell, Dean. Scaled Agile Framework, <http://www.scaledagileframework.com/>  
Ambler, Scott, Disciplined Agile Delivery, <http://www.disciplinedagiledelivery.com/>  
Mesaglio, Mary; Adnams, Suzanne; Mingay, Simon, Kick-Start Bimodal IT by Launching Mode 2, [www.gartner.com](http://www.gartner.com)  
Gaughan, Dennis, Pace Layered Application Strategy, [www.gartner.com](http://www.gartner.com)  
Rubin, Kenneth, 2013. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley  
James, Michael, Scrum Training Series, <http://scrummethodology.com/category/scrum/>