Agile Introduction
March 2017

Agenda

What is Agile
- Does this sound familiar?
- Common myths of Agile
- Agile is set of values and principles
- What it means to ‘Be Agile’

Why Agile?
- The Science Behind Agile
- How is Agile Different?

Agile Suitability
- Agile Suitability Criteria

Agile Contracting
- Agile Contracting
- Fixed Price Variable Scope
- Fixed Price Fixed Capacity
- Fixed Price Consideration

Agile ERP
- Benefits of Agile ERP
- Consideration for Agile ERP
- Agile Suitability for ERP
- Agile at Scaled (Scaled Agile Framework)
What is Agile?

- Does this sound familiar?
- Common myths of Agile
- Agile is set of values and principles
- What it means to ‘Be Agile’

Does this sound familiar?


Common myths of Agile

- “Faster & Cheaper”
- Delivering without testing
- Ad-hoc scope and project management
- Coding without documentation
- Doing more work with less people
- Having no “process”

Agile is set of values and principles

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

**Individuals and interactions** over processes and tools
**Working software** over comprehensive documentation
**Customer collaboration** over contract negotiation
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Sourced from [http://agilemanifesto.org/](http://agilemanifesto.org/)
What it means to ‘Be Agile’

- Satisfy the customer with valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business and developers must work together daily
- Build projects around motivated individuals
- Maximize face-to-face conversation

- Progress is working software
- Promote sustainable development: a constant pace indefinitely
- Continuous attention to technical excellence and good design
- Promote simplicity: maximizing work not done
- Encourage self-organizing teams
- Regularly reflect on how to become more effective

Source: http://Agilemanifesto.org/principles.html

Why Agile?

- The Science Behind Agile
- How is Agile Different?
The Science Behind Agile

The Theory of Constraints
- The theory of constraints (TOC) is a management paradigm that views any manageable system as being limited in achieving more of its goals by a very small number of constraints.
- TOC adopts the common idiom "a chain is no stronger than its weakest link." This means that processes, organizations, etc., are vulnerable because the weakest person or part can always damage or break them or at least adversely affect the outcome.

Deming Cycles
- Continuous quality improvement model consisting out of a logical sequence of four repetitive steps for continuous improvement and learning: Plan, Do, Check, and Adjust

Motivation Trifecta
- Autonomy - Giving people real control over various aspects of their work whether it's deciding what to work on or when to do it.
- Mastery - Employers should look at calibrating what people must do by looking at what they can do. If the must-tasks are too difficult, people will become worried and feel out of their league. If the must-tasks are too easy, they'll get bored.
- Purpose - is what gets you out of bed in the morning. People who have purpose are motivated to pursue the most difficult problems.

The Science Behind Agile

How is Agile Different: Process Difference

Agile enables greater responsiveness to change by only scoping parts of the solution at a time

Waterfall
- Demand Management Estimation Prioritization
- Design
- Testing
- Coding
- Release

- Waterfall requires single big bang delivery
- Fixed budget for fixed scope
- Quality can only be ensured when big bang is delivered

Agile
- Requirements
- Design
- Testing
- Coding
- Release

- Quality in Agile continuously improved
- Agile handles changes intensively
How is Agile Different: Responding to Change

Agile enables greater **responsiveness to change** by only scoping parts of the solution at a time.

![Diagram showing Agile process](image)

How is Agile Different: Funding Difference

Both Processes Recognize 3 Constraints in the “Iron Triangle”: Cost, Schedule, and Scope.

<table>
<thead>
<tr>
<th>Waterfall</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is Fixed?</strong></td>
<td>Scope (Requirements)</td>
</tr>
<tr>
<td><strong>What is Estimated?</strong></td>
<td>Cost</td>
</tr>
</tbody>
</table>

**Waterfall**
- “Tries” to lock down requirements so that schedule and cost can be estimated
- Extensive Planning is done
- Even Best Estimates are subject to change
- Project dates remain fixed, even if project scope changes

**Agile**
- Scope can always change: Schedule and cost should be fixed
- Planning is done at release level
- Potentially deployable after each iteration
- Implement highest value features first

![Diagram showing Waterfall and Agile processes](image)
How is Agile Different: Planning Difference

Forward-looking, rolling-wave, just-in-time planning reverses the "cone of uncertainty" and results in better decision-making, greater responsiveness to change, & fewer missed commitments

- Planning never stops
- Only detail & commit to the next most important thing
- Leave future items at high-level, until they're the important thing
- Add, remove, swap, and re-prioritize often to respond to change

Looking Forward in time…

Predictable, Committed | Uncertain, Planned | Unpredictable, Open-ended
--- | --- | ---
Flexible, Unformulated

How is Agile Different: Agile vs. Waterfall

<table>
<thead>
<tr>
<th></th>
<th>Agile</th>
<th>Waterfall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Learning</strong></td>
<td>Driven</td>
<td>Plan Driven</td>
</tr>
<tr>
<td><strong>Continuous communication with Business</strong></td>
<td></td>
<td>Less frequent business communication</td>
</tr>
<tr>
<td><strong>Short releases</strong></td>
<td>of 1 – 3 months (focused in business priorities)</td>
<td>Big Bang releases (Typically 9 – 12 months)</td>
</tr>
<tr>
<td><strong>Requirements evolving</strong></td>
<td>throughout the project</td>
<td>Up-front, locked requirements</td>
</tr>
<tr>
<td><strong>Delivery of working code in short iterations</strong></td>
<td>(2 – 4 weeks)</td>
<td>Development in distinct phases with interim deliverables</td>
</tr>
<tr>
<td><strong>Develop end-to-end functionality in iterations</strong></td>
<td></td>
<td>Develop in layers: Presentation, Integration, Business, Persistence etc.</td>
</tr>
<tr>
<td><strong>View programming as Design</strong></td>
<td></td>
<td>View programming as Construction</td>
</tr>
<tr>
<td><strong>Continuous integration</strong></td>
<td></td>
<td>Integration occurs at the end of build phase</td>
</tr>
<tr>
<td><strong>Testing occurs throughout the iteration at unit as well as functional level</strong></td>
<td></td>
<td>Testing occurs at the end in its own phase</td>
</tr>
<tr>
<td><strong>Low cost &amp; management of change</strong></td>
<td></td>
<td>High cost &amp; management of change</td>
</tr>
</tbody>
</table>
How is Agile Different: Benefits

Clients from various industries have seen real benefits from moving to Agile

<table>
<thead>
<tr>
<th>Agile Benefits</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster speed to market</td>
<td>Global Insurance Company • Improved speed-to-market based in a faster development lifecycle. Cycle was reduced from 250 to 60 days</td>
</tr>
<tr>
<td></td>
<td>Large Australasian Telecommunication Company • Reduced delivery timeframe from every 3-6 months to every 2 weeks.</td>
</tr>
<tr>
<td></td>
<td>US Equipment Manufacturer • Reduced time to market 20%</td>
</tr>
<tr>
<td>Built-In Quality</td>
<td>Global Insurance Company • Better filtering and prioritization of products resulting in product waste reduction from 40% to 10%</td>
</tr>
<tr>
<td></td>
<td>Global Healthcare Company • 20% - 30% fewer defects found in FDA-regulated devices</td>
</tr>
<tr>
<td></td>
<td>Payment Gateway Solution Provider • Automated 99% of test suite execution; providing reliable, repeatable, and consistent test results across all environments</td>
</tr>
<tr>
<td>Improved Customer</td>
<td>Large Telecommunications Company • Improved customer satisfaction through quicker releases of priority customer facing features</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>Global Entertainment Company • Used lean UX and Agile methodologies to produce meaningful epics and user stories. Successfully delivering a live-streaming application that replaced a legacy application that improving movie download times.</td>
</tr>
<tr>
<td>Reduced Costs</td>
<td>Global Technology Company • Reorganization based in Agile best practices, improved product velocity and quality, and reduced labor costs by 15%.</td>
</tr>
<tr>
<td></td>
<td>• Identified additional value levers for improving engineering productivity through automation and reduced product complexity that freed up an incremental 5% of labor productivity</td>
</tr>
</tbody>
</table>

Agile Suitability

- Agile Suitability Criteria
Agile Suitability Criteria

While Agile values and principles can be applied to all projects, there are certain projects where Agile might be more suitable.

<table>
<thead>
<tr>
<th>High Agile Suitability</th>
<th>Medium Suitability</th>
<th>Low Agile Suitability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Interfaces</td>
<td>Data warehouse / Business Intelligence based projects</td>
<td>Legacy Systems</td>
</tr>
<tr>
<td>Applications with heavy user experience attributes</td>
<td>(e.g. .NET, Java etc.)</td>
<td>Regulatory or compliance applications</td>
</tr>
<tr>
<td>Typically the applications will integrate into less systems</td>
<td>Application with heavy integration into more systems</td>
<td>Applications with heavy integration into more systems</td>
</tr>
<tr>
<td>Dashboard type interfaces</td>
<td>Canned module type interfaces</td>
<td>Canned module type interfaces</td>
</tr>
<tr>
<td>Projects that can be broken up into smaller logical pieces of functionality</td>
<td>Projects that are difficult to break into smaller features</td>
<td></td>
</tr>
<tr>
<td>High Suitability</td>
<td>Low Suitability</td>
<td>Low Agile Suitability</td>
</tr>
<tr>
<td>Mobility or digitization projects (e.g. .NET, Java etc.)</td>
<td>Other COTS platform based projects, Legacy Technology based projects (e.g. mainframe, etc.)</td>
<td></td>
</tr>
<tr>
<td>Regulatory or compliance applications</td>
<td>Legacy Systems</td>
<td></td>
</tr>
<tr>
<td>Applications with heavy integration into more systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Agile Contracting

- Agile Contracting
- Fixed Price - Variable Scope
- Fixed Price - Fixed Capacity
- Fixed Price Consideration
Agile Contracting

There is no Agile contract without mutual trust!

- Every customer is moving to Agile because of the flexibility it provides to make changes at any time based on the market conditions and emerging trends. ‘Speed to market’ is the key here. So the most common commercial model used in Agile contracting is **Time & Material (T&M)**.

- However to manage the budget better, avoid financial risk and make vendor accountable for delivery, some of customers prefer a Fixed Price model. Given Fixed Price contracts compromises the Agile Manifesto such as ‘Responding to change over following a plan’, ‘Customer collaboration over contract negotiation’, Fixed Price (FP) hybrid models (Fixed + T&M) are often utilized.

---

Agile Contracting: Fixed Price - Variable Scope

While the price of the project is fixed based on the planned velocity, the developed functionality would vary based on product backlog prioritization

- Fixed price projects have defined and fixed requirements but Agile projects have defined scope with flexible requirements which are implemented based on customer’s priority and technical feasibility.
- Initiation and Sprint 0 duration would be longer in fixed price Agile projects. These phases can follow T&M.
- Product vision, Release road map, Raw Epics and stories form the variable scope at the stage of contract writing.
- Acceptance criteria is must for each user story and the customer involvement is very much required throughout the project stages on 3Cs(Card, Conversation and Confirmation).
- Estimate the planned velocity using comparison of user stories through Triangulation.
- Include a ceremony for check point meeting on scope control, hence change management is mandatory.

How Fixed Price Variable Scope Scrum Development Works?

<table>
<thead>
<tr>
<th>Time and Material</th>
<th>Fixed Price Scope</th>
<th>Time and Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td>Sprint 0</td>
<td>Deploy Release</td>
</tr>
<tr>
<td>Sprint 1</td>
<td>Sprint 2</td>
<td>Check Point Meeting</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>Sprint N</td>
<td>Release burn up</td>
</tr>
<tr>
<td>Sprint N</td>
<td></td>
<td>With scope volatility</td>
</tr>
</tbody>
</table>

**Product Backlog Priorities**

- Trust based contracting
- Story points for a fixed price
- Scope Control and Change Management

- High-level architecture/design/FRUs/POCs
- Finalize the estimate to fix the price of the backlog to be delivered
- Summary reports added
- New Billing epics added
- Final regression and performance testing with user training.

Agile Contracting: Fixed Price – Fixed Capacity

While the price of the project is fixed based on the team’s capacity to deliver ‘n’ story points for every sprint, the developed functionality would vary based on product backlog prioritization.

- Initiation and Sprint 0 duration would be longer in fixed price Agile projects. These phases can follow T&M.
- Product vision, Release road map, Raw Epics and Stories form the variable scope at the stage of contract writing.
- Acceptance criteria is must for each user story and the customer involvement is very much required throughout the project stages on 3Cs(Card, Conversation and Confirmation).
- Estimate the planned velocity using comparison of user stories through Triangulation.
- Decide the number of story points to be delivered in every sprint/release with the fixed team capacity.
- Estimation of effort and cost can be arrived based on the available capacity and release dates.
- Customer will generally drive the scope control and change management tasks.
How Fixed Price Fixed Capacity Scrum Development Works

<table>
<thead>
<tr>
<th>Time and Material</th>
<th>Fixed Price Fixed Capacity</th>
<th>Time and Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td>Sprint 1</td>
<td>Check Point Meeting</td>
</tr>
<tr>
<td>Sprint 0</td>
<td>Sprint 2</td>
<td>Release Burn Up</td>
</tr>
<tr>
<td>Sprint 1</td>
<td>Sprint 3</td>
<td>Summary Reports Added</td>
</tr>
<tr>
<td>Sprint N</td>
<td></td>
<td>2 new Billing epics added</td>
</tr>
</tbody>
</table>

Product Backlog Priorities

- High level architecture/design/initial POCs
- Finalize an estimate to fix the velocity of the team to deliver X Story Points in a sprint
- Final regression and performance testing with user training.
- Product Backlog for Priorities

Fixed Price Consideration

The contract and agreement between customer and vendor are purely trust based. Terms and conditions for Agile contract on functional scope is not applicable. Change Management is required to monitor varying scope and advise on the coarse correction. Instead of milestone review, Payment schedule is aligned with check point review meetings to bill against Story points delivered.

Expectations on the customer and their Business users/SMEs:

- Prioritize user stories by business value and track them to be implemented in the order of priority.
- Participate in each sprint planning meeting to discuss/decide the features to be developed.
- Provide clarifications to the development team as and when required
- Participate in each sprint review meeting and provide timely feedback
- Attend check point meetings
  - To advise the required functionality for the upcoming releases in order to deliver the high critical business features within the story points limited.
  - To suggest on change request creation and approval for additional story points inclusion.
Agile ERP

- Benefits of Agile ERP
- Consideration for Agile ERP
- Agile Suitability for ERP
- Agile at Scaled (Scaled Agile Framework)

Benefits of Agile ERP

We often see significant benefits with an Agile model that goes beyond the standard “cheaper and faster”

**Increase Delivery Quality and Speed-to-Value**
- Early solution integration and business demos reduce defects later in the delivery cycle when they are most expensive.
  - Working software validated by the business at regular intervals
  - Frequent integration and “shift-left” testing methods break down silos and drive early defect detection
  - Focus on retrospective and continuous improvement across all teams

**Improve Business Adoption & Satisfaction**
- Product increments of working software with real-time business collaboration.
  - Generate a sense of ownership as business drives priorities through backlog grooming
  - Final product is closer to what end users envision due to frequent feedback loop
  - Promotes communication and partnership between business & software delivery teams

**Deliver More Efficiently**
- Identify and target opportunities to optimize the documentation footprint, delivery tools, and processes.
  - Reduce the process and deliverable overhead – focus on documenting what we did vs. what we plan to do
  - Upfront documentation of test criteria reduces scope creep and wasted time in development (developers focus on required functionality vs. unnecessary “extras”)
  - Use shift to new delivery model as a springboard to prompt other efficiencies
Considerations for Agile ERP

The transition to an Agile model for ERP requires upfront investment in change management and significant commitment from leadership.

- **Leadership Commitment**: The transition requires top-down commitment to the change, as it will likely be challenging and require leaders who are dedicated to seeing it through.
- **Pilot Your Approach**: Create evidence and learnings through a small pilot deployment.
- **Invest in Training**: The shift to Agile is not only about methods and ways of working, but a true mindset shift – this is best captured via in-person training courses.
- **Planning never stops**: Rigorous and ongoing planning is required to sequence integrated work.
- **Variable scope requires high trust relationship**: While high-level scope is still setup up front, the iterative approach results in scope that is planned incrementally and thus can be variable.
- **Business involvement is crucial**: Product Owner engagement and end-user (i.e. Power User) involvement throughout the project is key to success in Agile delivery for ERP
- **Remain Focused on Scalability**: Ask and answer how pilot principles will work at scale.
- **Agile as a Change Agent**: Look holistically for efficiencies, employing LEAN and other methods.

Agile Suitability for ERP

Agile suitability for type of ERP projects

<table>
<thead>
<tr>
<th>Agile Suitability for ERP</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof of Concept/Pilot</td>
<td>• Piloting a small functionality or business process for transforming a concept into a working live example</td>
</tr>
<tr>
<td>Enhancements</td>
<td>• Many eCommerce applications may not have Order Management to start with and may decide to add it later</td>
</tr>
<tr>
<td>Rollout Projects</td>
<td>• Rollout out of the customized features in the regions</td>
</tr>
<tr>
<td>Implementation Projects</td>
<td>• Implementation of a Human Resources system (Payroll, Benefits etc.)</td>
</tr>
<tr>
<td>Upgrade (Technical) Projects</td>
<td>• Insurance implementation like Claims review and processing</td>
</tr>
<tr>
<td>Application Support</td>
<td>• Upgrading to a new release of an ERP provides opportunities to have a measurable business impact in the areas of operational excellence and business strategy for a client</td>
</tr>
<tr>
<td></td>
<td>• Any changes to any of the modules</td>
</tr>
<tr>
<td></td>
<td>• Defect fixes</td>
</tr>
</tbody>
</table>

Examples
**Agile at Scale**

Historically, Agile has been approached as a grassroots effort within pockets of an organization, and very rarely within the large-scale ERP space.

- Over time, a growing population has seen and experienced the positive results of Agile delivery
- These same people are now interested in scaling Agile across their organizations
- Several published frameworks have emerged to fill this market need, however Scaled Agile Framework (SAFe) is the approach most frequently used by most of our clients

**Scaled Agile Framework**

Historically, Agile has been approached as a grassroots effort within pockets of an organization, and very rarely within the large-scale ERP space.

- In response to market demand for an industrialized enterprise Agile framework, SAFe was created by blending concepts from Scrum project management, XP programming, program management, portfolio management, and Lean
- SAFe has three distinct levels
  - **Portfolio**, where strategies and epics are defined and prioritized
  - **Program**, where epics are broken into features, scheduled into releases, and released when ready
  - **Team**, where development is executed by cross-functional teams