Agile Execution for and Beyond IT

Fred Koos, PE, PMP, CSM, Pittsburgh PMI VP of Education
31 December 2013

This is an enhanced version of the information presented in the last session at the 11/6/13 Pittsburgh PMI Professional Development Day (PDD). In addition, for information presented by Joe DiFalco, at the Pittsburgh PMI PDD please contact Joe@JDiFalco.com.

Objectives:

- Provide an introduction to Lean principles, Agile principles, and the Agile Manifesto.
- Draw comparisons between Lean and Agile. This helps to provide a base from which to apply Agile to various types of projects, in and beyond IT.
- Observe that if Lean for was developed for manufacturing and new product development, and then Agile has some roots in adapting Lean for software development, then in turn we can adapt Agile for use on new product development projects beyond software. This increases the tool set available for managing projects.
- Review an example of using Agile principles for new product development. The product here is a kitchen. This project will apply Agile to the Build Phase of the kitchen. This should provoke additional creative thoughts on how to apply Agile principles on a variety of projects. The focus should be to reduce waste and improve the flow of value.
- Provide resources for more information on Agile principles and methodologies.

Part 1: Predicting the Future with Data to Better Manage Projects

A project scenario with planning months or years into the unknown future

Before reviewing Agile principles, let's look at an example that is lacking some agility. A 5-Why analysis and a variation of a trend analysis will be used. These two tools can be applied to improve various plans including program roadmaps, Agile release plans, project milestone schedules, or other plans.

The project in this example has progressed two months through January and February, 2014. It is behind the planned schedule for the first key milestone which was planned to be complete at the end of February. As a baseline for improving from the current state, there is a need to re-plan the schedule to move the future milestones to be one week later? See Figure 1.

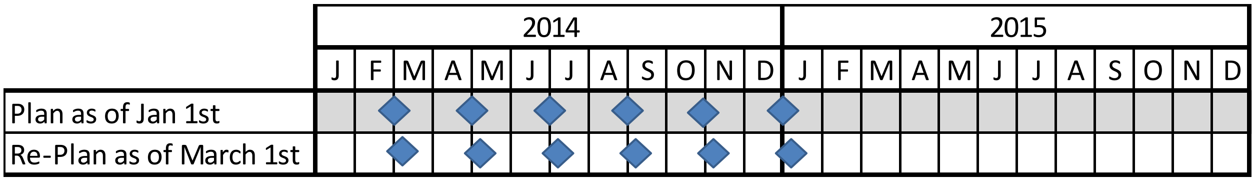


Figure 1. Milestone schedule showing 1 week delay after two months of work.

On a one-year project this could result in an updated schedule, shifting the predicted completion of the project to be one week later than planned. As this updated schedule is communicated to the stakeholders, there are significant risks related to continued slippage in the future.

To help evaluate this risk, a root cause analysis on time slippage can be done. A 5-Why analysis follows:

- #1: Why is there a 1 week delay?
  - Answer: The team overestimated their capacity to complete the work.
  - #2: Why did the team overestimate their capacity for this work?
    - Answer: Part of the work was not predicted in the original estimates.
    - #3: Why was part of the work not predicted in the original estimates?
      - Answer: The team members have never performed similar work and experienced the associated details, thus they did not plan all needed steps.
      - #4: Why have the team members never performed similar work?
        - Answer: The team members are good in general, but are inexperienced and not well trained on the specific skills needed for this particular work.
        - #5: Why are the team members inexperienced and not well trained?
          - Answer: The budget of this project does not allow for adequately experienced team members.

For this and additional information on 5 Why Analysis see http://en.wikipedia.org/wiki/5_Whys.

By simply re-planning and moving the schedule to be one week later, the root cause is not addressed. The risk of continued delays from the same root cause exists. Without addressing the root cause with an irreversible corrective action, similar delays should be expected going forward. If the team commits to the stakeholders that the completion date will be just one week later, they may be repeating similar communication of delays with the stakeholders throughout the project, slipping one week every two months.

In parallel with addressing the root cause, it is important to estimate the future performance with the best possible information. A tool can be used to help depict the timing trends and plan for the future if these trends continue.

Predicting the Future – The Approach

Continuing at the rate above could give another week delay for every two months of project work. A data driven tool for predicting the future is to show this trend in a diagram such as that in Figure 2 below. Without addressing the root causes, the trend may continue to the end of the project. This shows that after 12 months that there is a chance of slipping approximately 6 weeks. This slippage adds approximately 2 months which adds another potential week of delay. The entire project is at risk of being 7 weeks later than originally planned.

To manage the potential delays, actions could include addressing the root cause, implementing ways to speed up the work, implementing alternative ways to supplement the business objectives, communication with the stakeholders, managing the business processes among potential project delays, etc.

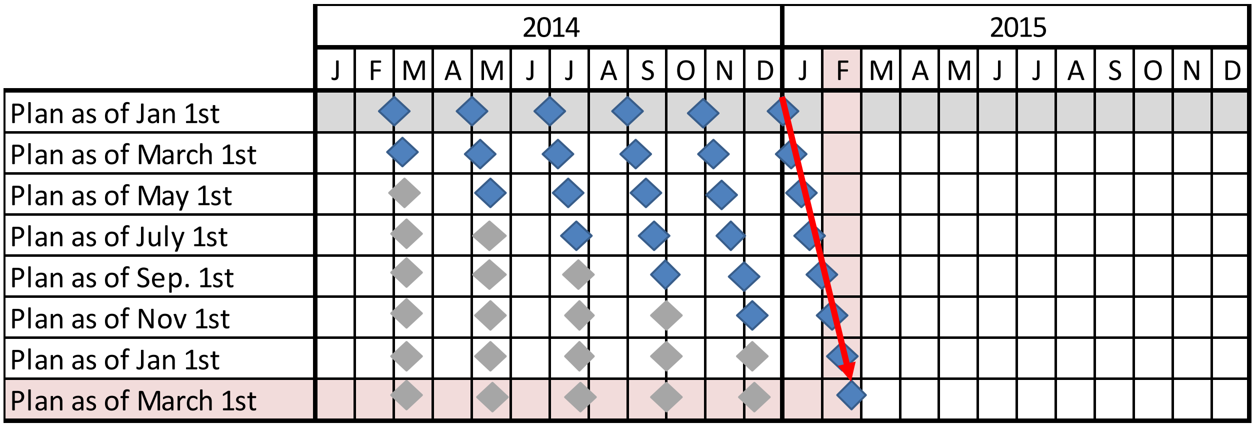


Figure 2. Milestone schedule showing 1 week delay for every two months of work.

On software or R&D projects, using the above method to predict the trends can help to manage the timing of the project's business value. This method should be combined with applicable statistical methods.

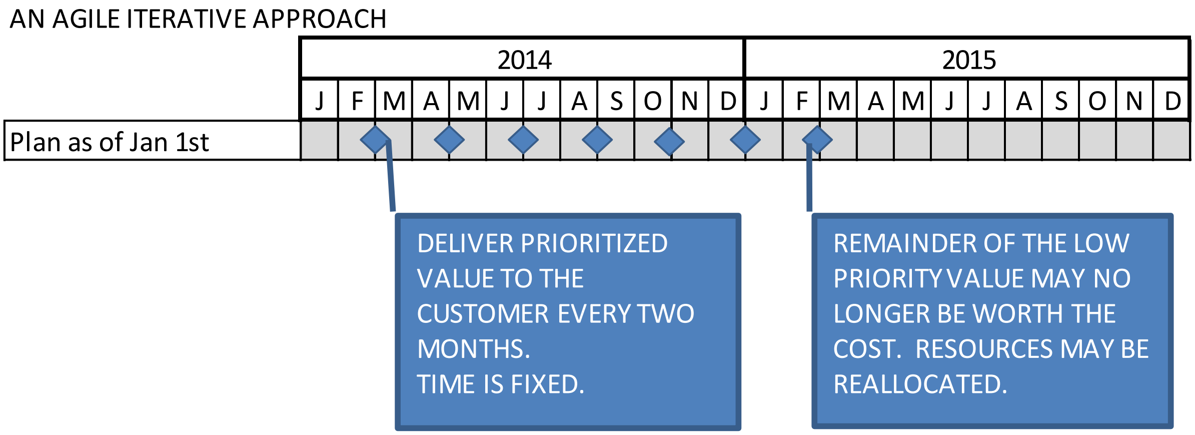
The above tool is valuable in circumstances when long-term commitments must be managed. Still, there is always variation when attempting to predict the future. An additional set of tools can be used for IT or R&D projects, or any project with many unknowns. This set of tools includes "Lean" and "Agile" methods. These methods create a more streamlined path to creating a flow of business value.

Part 2: Managing Innovative Projects with Many Unknowns Using Lean and Agile Methods

Some questions related to estimating project timing follow:

- Can we honestly commit to long-term end dates with fixed scope and budget on innovative projects with many unknowns?
- Will this approach maximize customer value?
- Will this approach minimize waste?
- Would this be "Lean"?

Figure 3 below uses the project example from above and depicts a timeline of a different scenario. In this scenario using Agile and Lean methods, the near term is planned in more detail. The future is planned with less detail. Frequent reprioritization of the future work occurs as the project progresses. Prioritized business value is defined and delivered every two months. The high priority business value is delivered first and the work progresses based on priorities and other factors. Overall, it is possible that the final parts of the work are planned to be complete 7 weeks later than the original 12 month schedule. At this time it may be the case that the majority of the business value is already obtained and the work of the remaining 7 weeks is of less business value than other organizational work. If so, this project may be closed and resources reallocated.


AN AGILE ITERATIVE APPROACH
2014
2015
J F M A M J J A S O N D J F M A M J J A S O N D
Plan as of Jan 1st

DELIVER PRIORITIZED VALUE TO THE CUSTOMER EVERY TWO MONTHS.
TIME IS FIXED.

REMAINDER OF THE LOW PRIORITY VALUE MAY NO LONGER BE WORTH THE COST. RESOURCES MAY BE REALLOCATED.

Figure 3. A more Agile approach to provide value much sooner.

In a variety of ways, this approach would be "Agile" and "Lean".

Some fundamentals of Agile and Lean are next.

What is Lean?

"The core idea is to maximize customer value while minimizing waste.
Simply, Lean means…
creating more value for customers with fewer resources."

See www.lean.org/WhatsLean.

Principles of Lean

"Specify value from the standpoint of the end customer by product family.
Identify all the steps in the value stream for each product family, eliminating whenever possible those steps that do not create value.
Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer.
As flow is introduced, let customers pull value from the next upstream activity.
As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste."

See http://www.lean.org/WhatsLean/Principles.cfm.

Manifesto for Agile Software Development (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.”

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

See http://agilemanifesto.org.

Principles behind the Agile Manifesto

“We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

See http://agilemanifesto.org/principles.

Applying Agile Beyond IT Exercise 1

Comparing Agile to Lean principles is an exercise that the reader can perform. This should provide a greater understanding of these principles. This greater understanding can then uncover additional applications of Agile and Lean principles to enhance project and program performance.

This exercise is defined in the form of a User Story. Below the standard Agile User Story language is underlined. The language that should be modified for each specific use is in italics.

Exercise 1 User Story:
As a *student of life,* I want to *know how the 4 Agile values and 12 Agile principles relate to the Lean definition and 5 Lean principles,* so that *I can apply them to my projects and maximize customer value!*

In this exercise, these relationships can be drawn within a 10 minute time-boxed "sprint".
(Sprints on actual projects can be time-boxed into fixed durations that would typically be 2 to 8 weeks each. In comparison, in this example one minute could equate to one day on a real project.)

On the next page in the left-hand column are the following:

- Definition of Lean ("What is Lean?")
- 5 Principles of Lean

In the right-hand column are the following:

- 4 Agile values
- 12 Agile principles (purposely not in order).

Drawing a relationship to each Agile item would result in 16 Tasks. As these tasks are completed, at the end of each minute the trend of the remaining number of tasks can be depicted as in Figure 4.

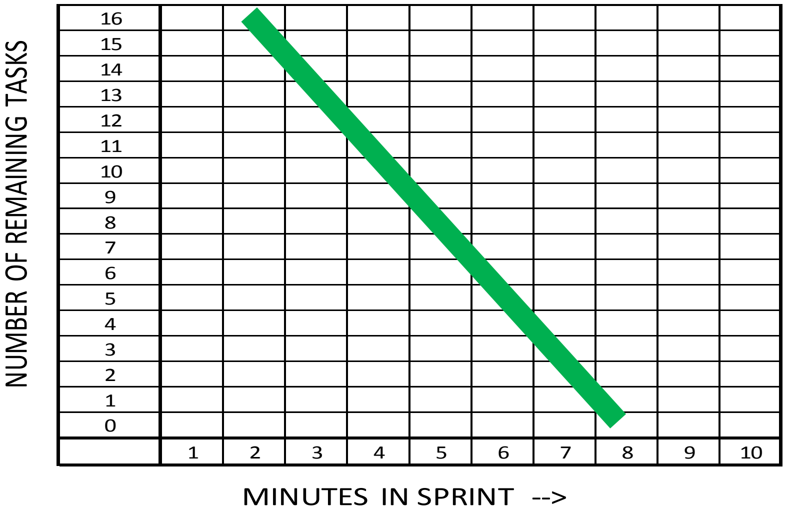


Figure 4. Burn-down chart of remaining tasks at the end of each minute

LEAN

What is Lean?

The core idea is to maximize **customer value** while minimizing waste. Simply, Lean means creating more value for customers with fewer resources.

Principles of Lean

1. Specify value from the standpoint of the end customer by product family.

2. Identify all the steps in the value stream for each product family, eliminating whenever possible those steps that do not create value.

3. Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer.

4. As flow is introduced, let customers pull value from the next upstream activity.

5. As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste.

AGILE

Agile Values:

1. Individuals and interactions over processes and tools

2. Working software over comprehensive documentation

3. Customer collaboration over contract negotiation

4. Responding to change over following a plan

Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

4. Business people and developers must work together daily throughout the project.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

9. Continuous attention to technical excellence and good design enhances agility.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Overall, this exercise of drawing the relationships between Lean and Agile can be structured as follows:

Sprint Time-box: 10 minutes

- Planning: 1 min.
- Executing, matching each Agile item to a Lean item: 7 min.
    - Plot the number of remaining tasks at each minute
- Reviewing the result with a demonstration: 1 min.
- Retrospective/Lessons Learned: 1 min.

As a reference and for more information see
http://en.wikipedia.org/wiki/Scrum_(software_development).

Example of a Solution

A plot of progress at each minute is recorded by putting an "**X**" in the cell at the intersection of the row for the number of remaining tasks and the column of the minute.

Below is a planned trend line in blue that will keep the work on track.
An example of the actual performance is plotted with the **X**'s.

- At the end of the 2nd minute all 16 tasks are remaining.
- At the end of the 3rd minute the first Agile value was matched to the Definition of Lean, leaving 15 tasks remaining.
- At the end of the 4th minute two more matches were found, leaving 13 tasks remaining.
- At each minute progress continues to be tracked until all 16 tasks are complete at the end of the 8th minute.

Refer to Figure 5 on the next page to see the trend on the burn-down chart.

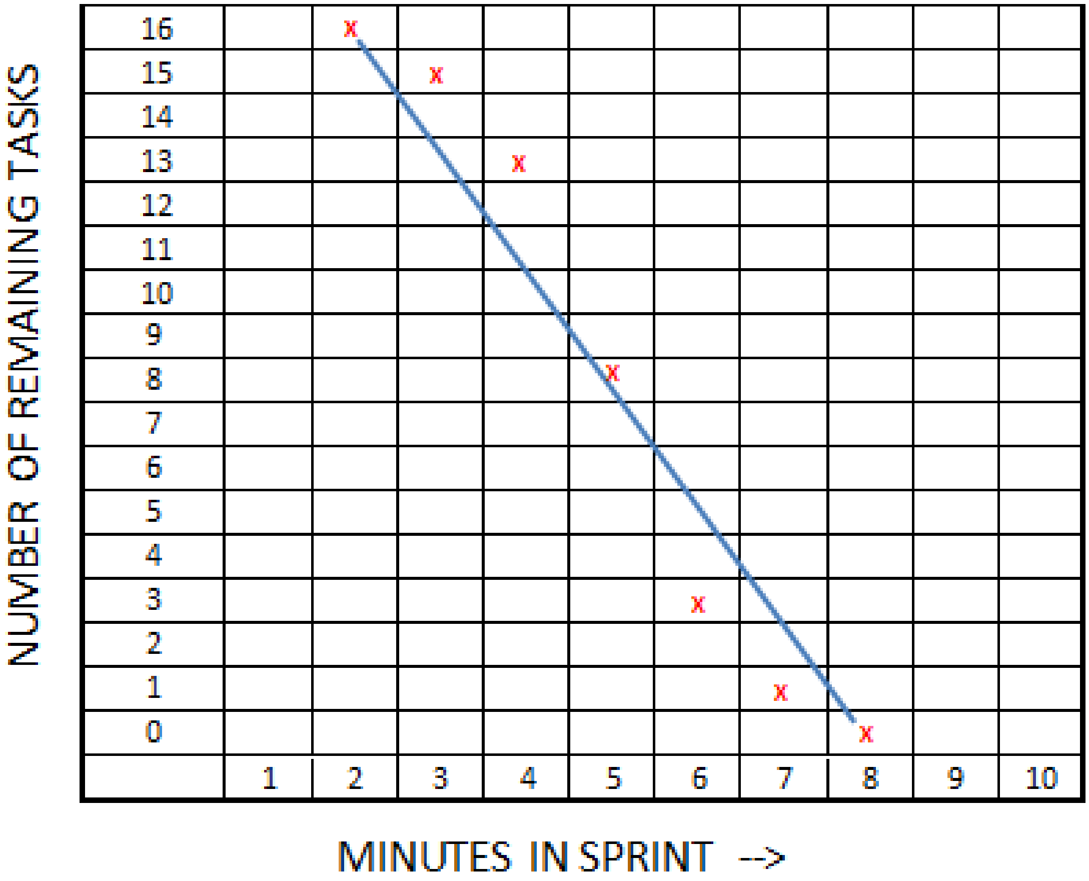


Figure 5. Exercise 1, burn-down chart of remaining tasks at the end of each minute

Some matches between Lean and Agile are indicated by the lines in the following table. More relationships can be found by the reader.

LEAN	AGILE
What is Lean? The core idea is to maximize **customer value** while minimizing waste. Simply, Lean means creating more value for customers with fewer resources.	**Agile Values:** 1. Individuals and interactions over processes and tools 2. Working software over comprehensive documentation 3. Customer collaboration over contract negotiation 4. Responding to change over following a plan
Principles of Lean 1. Specify value from the standpoint of the end customer by product family.	**Agile Principles** 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. 4. Business people and developers must work together daily throughout the project. 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. 7. Working software is the primary measure of progress.
2. Identify all the steps in the value stream for each product family, eliminating whenever possible those steps that do not create value.	5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. 10. Simplicity--the art of maximizing the amount of work not done--is essential.
3. Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer.	3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
4. As flow is introduced, let customers pull value from the next upstream activity.	2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
5. As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste.	9. Continuous attention to technical excellence and good design enhances agility. 11. The best architectures, requirements, and designs emerge from self-organizing teams. 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Remodeling a Kitchen

Lean and Agile can be applied to remodeling a kitchen. Some examples are in the following table. Please think of your additional examples.

Lean and Agile	**Associated Kitchen Remodeling Approaches**
What is Lean?	
The core idea is to maximize **customer value** while minimizing waste. Simply, Lean means creating more value for customers with fewer resources.	The customer wants a kitchen which enables efficient work with a natural based design. The kitchen architecture and design is created followed by building only those items that add significant value. Per the customer, the kitchen must continuously be clean and operational from Monday through Friday each week. The windows for disruptions due to construction are from Saturday morning at 8:00 AM through Sunday afternoon at 5:00 PM. At 5:00 PM each week the kitchen needs to be clean and operational with the planned functionality for that week.
Principles of Lean	
1. Specify value from the standpoint of the end customer by product family.	The most urgently needed item is a larger refrigerator, followed by a larger sink and disposal. The priorities continue with the island sink and reverse osmosis water purifier being a lower priority.
2. Identify all the steps in the value stream for each product family, eliminating whenever possible those steps that do not create value.	Define sections of the kitchen for each iteration. Map the flow of material from the source to the installed product. Optimize flow.
3. Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer.	Order and pickup material only as needed for the current iteration as it is needed. Build only the section of the current iteration and release it to the customer before the next iteration is started.
4. As flow is introduced, let customers pull value from the next upstream activity.	As one iteration is complete, pull the next iteration into action. Don't order material early if the iteration being pulled for does not need it.
5. As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste.	At the end of each iteration of building part of the kitchen, ask what went well to continue to implement, and ask what can be improved for the next iteration. Make associated improvements.

See the next page for the second part of this table.

Lean and Agile	**Associated Kitchen Remodeling Approaches**
Agile Values:	
1. Individuals and interactions over processes and tools	Have 15 minute daily standup meetings in the kitchen with the owner to review progress made and discuss the next steps. Refine the plan over time.
2. Working software over comprehensive documentation	Working sections of the kitchen add value to the project. Documentation is kept small for items including the budget, interface definitions, models for communication, and progress tracking.
3. Customer collaboration over contract negotiation	Our kitchen building team has earned a trusting relationship with the kitchen owner. Cost is kept down and time is kept short by frequent communication of needs and by updates of progress to meet the needs. Elaborate contracts are minimized.
4. Responding to change over following a plan	High level designs and plans are created. As the physical kitchen is built, section by section, the kitchen owner gets a clearer picture of the final product, thus prompting changes. Also, as needs evolve over time and the outside environment evolves, the desired product likewise evolves. The team follows the general plan and is flexible to the changing needs of the owner.
Agile Principles	
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Related to products, not just software, define the priority of sequence of kitchen features to provide in order. First is an updated larger refrigerator, followed by a larger kitchen sink, cabinets for glasses silverware and tableware, then followed by a larger capacity and quieter dishwasher. Other features of value for the customer will be listed to follow.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	The kitchen owner's needs will change over time, changing what is valuable for them. The planned kitchen features will evolve per changing owner needs. After the fourth iteration, with holiday coming, it was important to have a working double oven with a warming drawer functional, but not esthetically finished.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Every week deliver a usable, valuable section of the kitchen. The refridgerator is one of the first features to add.
4. Business people and developers must work together daily throughout the project.	Daily 15 minute standup meetings in the kitchen include the team and kitchen owner. Changes to the planned future features will be made based on frequent feedback.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	With great team members, ensure they have the information, skills, material, and clear vision of the end goal. Trust the team to find the best way to achive the goal.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Daily meetings live in the kitchen talking through what was done yesterday, what will be done in the next day, and what impediments the team members have. Have quick communication amoung the team in 15 minutes then break to resolve any issues and progress with the next task.
7. Working software is the primary measure of progress.	Progress is measured section by section as real value is added to the kitchen.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	The team planned to get the construction primarily done each Saturday, with Sunday as a buffer. Non construction work is done between weekends. This is a sustainable pace.
9. Continuous attention to technical excellence and good design enhances agility.	The team applies best practices for each job, including laying tile, installing cabinets and counters, plumbing, electrical work, etc. A safety margin is applied for future expansion and modifications of these and other elements.
10. Simplicity--the art of maximizing the amount of work not done--is essential.	We all listen closely and ask questions of the kitchen owner. We do what is valuable while not spending time, wasted, on other tasks.
11. The best architectures, requirements, and designs emerge from self-organizing teams.	The team discusses the goals at the start of every iteration. The team works closely together to determine the best methods to apply for each task.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	Each day and each iteration contain lessons to apply to improve. Structured iteration restrospective sessions are at the end of each iteration to celebrate what went well and what we should continue to do, and to make improvements where necessary.

With these thoughts, the kitchen remodeling can proceed in a Lean and Agile manner.

Remodeling a kitchen can be a significant inconvenience to the customer, with the timing often dictated by the builder. For reasons beyond the scope of this paper, it was decided that the kitchen will be remodeled by the family. The work will mostly be done on Saturdays with Sunday's as a buffer.

Two possible plans follow:

- Plan A: Tell the family we are closing the kitchen for 3 months while we remodel it ourselves.
- Plan B: Keep the kitchen open! Close only a part of the kitchen for 2 days while it is being remodeled. At the end of the 2 days this part of the kitchen will be updated and released back to the family for use. Additional design, planning, procurement, and other administrative work will be done between the construction events, thus not blocking any use of the kitchen.

Plan B is chosen to keep the kitchen open!

Within a set time-box, 2 days, replace one small part of the kitchen which will be upgraded and useable at the end of the time-box. There will be a short demonstration, retrospective, and celebration at the end of this second day. This is followed by a 5 day time-box for re-planning, procurement, and other administrative tasks. The 7 day cycle is repeated until the project is complete. It is estimated that this can be done comfortably with 12 sprints.

Note: If a contractor were to perform this work, the 2 day construction time boxes would be one day from 8:00 AM – 5:00 PM. The family could have uninterrupted use of the kitchen each day for dinner.

See the drawing of the kitchen layout below. Note that there are two adjacent parts of the kitchen. One is the pantry and the other is the main kitchen. The existence of the pantry provides additional flexibility in the sequencing of work. Without a pantry a slightly modified plan can be followed.

The numbers inside the dotted circles represent the 7-day sprint in which the corresponding part of the kitchen will be remodeled. The execution continues, following Scrum through a successful completion.

In summary, Agile and Lean are closely related. Agile and Lean principles can be applied to a kitchen remodeling project where specific approaches are defined to improve the project performance. A kitchen can continue to provide value to the family while being remodeled. A weekly stream of improved value can be provided without waiting until the end of the project to realize this value.

With this framework, there are various specific Agile methodologies that could be applied. For this project, with the sprints defined here, Scrum is recommended. For more information on Scrum see the book Agile Project Management with Scrum, Microsoft Press, 2004, by Ken Schwaber.

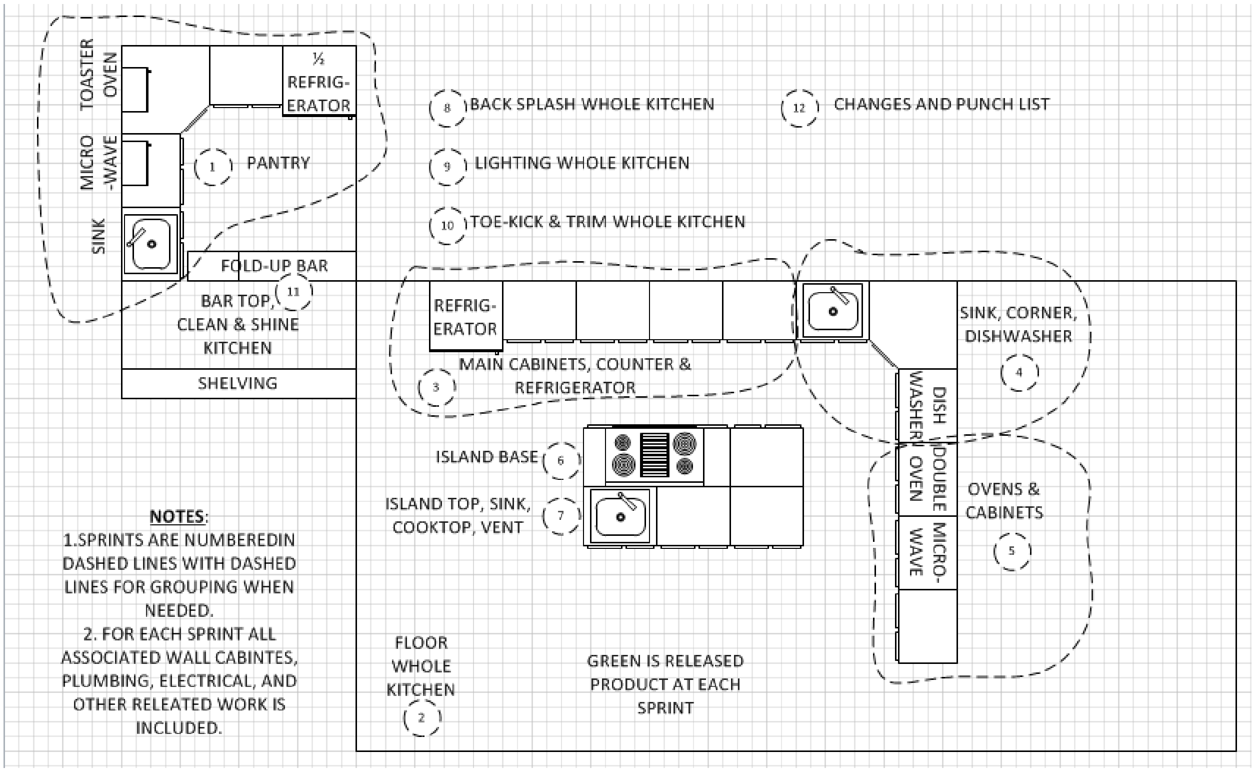

TOASTER OVEN
MICRO -WAVE
SINK
½ REFRIG-ERATOR
1 PANTRY
FOLD-UP BAR
BAR TOP, CLEAN & SHINE KITCHEN
11
SHELVING
8 BACK SPLASH WHOLE KITCHEN
12 CHANGES AND PUNCH LIST
9 LIGHTING WHOLE KITCHEN
10 TOE-KICK & TRIM WHOLE KITCHEN
REFRIG-ERATOR
MAIN CABINETS, COUNTER & REFRIGERATOR
3
SINK, CORNER, DISHWASHER
4
DISH WASHER
DOUBLE OVEN
MICRO-WAVE
OVENS & CABINETS
5
ISLAND BASE 6
ISLAND TOP, SINK, COOKTOP, VENT 7
NOTES:
1.SPRINTS ARE NUMBEREDIN DASHED LINES WITH DASHED LINES FOR GROUPING WHEN NEEDED.
2. FOR EACH SPRINT ALL ASSOCIATED WALL CABINTES, PLUMBING, ELECTRICAL, AND OTHER RELEATED WORK IS INCLUDED.
FLOOR WHOLE KITCHEN
2
GREEN IS RELEASED PRODUCT AT EACH SPRINT

Figure 5. Kitchen layout diagram with sprint numbers identified.

Other Examples of applying Agile beyond software projects

With the background of the Kitchen example, the reader can evaluate how to apply the Lean and Agile principles to the following types of projects.

- Highway construction
- U.S. Air Force Bare Base Construction
- New Product Development
- Any project of which you were on the team or observed

For information presented by Joe DiFalco, at the Pittsburgh PMI PDD please contact Joe@JDiFalco.com

Summary bullets follow:

- Agile and Lean can both be used on a variety of projects for improved performance.
- The execution of Agile has many applications for and beyond IT to improve value delivered.
- Tools are shown to take another step in the journey to become more Agile and Lean.
- Agile is a broad umbrella with some information here. Please see the bibliography below and find other sources for additional information on Agile methodologies, processes, and tools.

Fred Koos, PE, PMP, CSM
FredBKoos@IEEE.org
(Doing hard work when times are easier makes easier work when times are hard.)

Works Cited

“A Guide to Lean Enablers for Managing Engineering Programs.” *Joint MIT-PMI-INCOSE Community of Practice on Lean in Program Management.* Josef Oehmen, May 2012: Version 1.0. 1 Oct 2013 <http://www.pmi.org/Business-Solutions/EngineeringExcellence.aspx>.

Sarah Fister Gale. “Leaning into Savings.” *PM Network.* June 2013: 28-35 <http://www.pmi.org/Business-Solutions/~/media/PDF/Business-Solutions/Lean-Enablers/Lean-into-Savings.ashx>.

James P. Womack and Daniel T. Jones. *Lean Thinking, Banish Waste and Create Wealth in Your Corporation, Second Edition.* New York: Simon & Schuster, Inc. 2003 <http://www.lean.org/bookstore/ProductDetails.cfm?SelectedProductID=88>.

“Reference Materials for the PMI Agile Certified Practitioner (PMI-ACP)® Examination.” *PMI*. 1 Oct 2013 <http://www.pmi.org/en/Certification/~/media/Files/PDF/Agile/PMI000-GainInsightsAIGLE418.ashx>.

Kent Beck, Mike Beedle, et.al. “Manifesto for Agile Software Development.” *Agile Aliance*. 13 Feb 2001. 1 Oct 2013 <http://agilemanifesto.org>.

Kent Beck, Mike Beedle, et.al. “Principles behind the Agile Manifesto.” *Agile Aliance*. 13 Feb 2001. 1 Oct 2013 <http://agilemanifesto.org/principles>.

“What is Lean?” *Lean Enterprise Institute*. 1 Oct 2013 <www.lean.org/WhatsLean>.

“Principles of Lean” *Lean Enterprise Institute*. 1 Oct 2013 <http://www.lean.org/WhatsLean/Principles.cfm.>.

“5 Whys” *Wikipedia*. 1 Oct 2013 <http://en.wikipedia.org/wiki/5_Whys>.

“Scrum (software development).” *Wikipedia* 1 Oct 2013 <http://en.wikipedia.org/wiki/Scrum_(software_development)>.