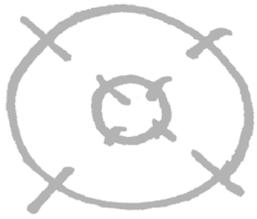


The Lean Change Method

Transform Your
Technology
Business through
Co-Creation and
Validated Learning

By Jeff
Anderson



OWN IS
5 REWARD



The Lean Change Method

Managing Agile Organizational Transformation Using
Kanban, Kotter, and Lean Startup Thinking

Jeff Anderson

This book is for sale at <http://leanpub.com/leanchangemethod>

This version was published on 2013-11-25



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#)

Tweet This Book!

Please help Jeff Anderson by spreading the word about this book on [Twitter!](#)

The suggested tweet for this book is:

I just purchased The Lean Change Method by @thomasjeffrey #leanchange

The suggested hashtag for this book is [#leanchange](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#leanchange>

to my wife Barbara, who has provided me with endless support and patience, without which this book would not be possible.

Contents

- 1 A Pattern Language for Agile Change 1**
 - The Agile Change Patterns Shift the Focus of Learning Depending on Where You Are in Your Agile Transformation 1
 - The Quick Win 3
 - Kernel Pilot 5
 - Kernel Adoption 8
 - Self-Starter 9
 - Capability Modernization 11
 - Overall Cadence Model 14

- Appendix 19**
 - Traditional Approaches to Technology Delivery Is No Longer Suited to Today’s Market . . 19
 - Agile Transformation Is Extremely Hard. 29

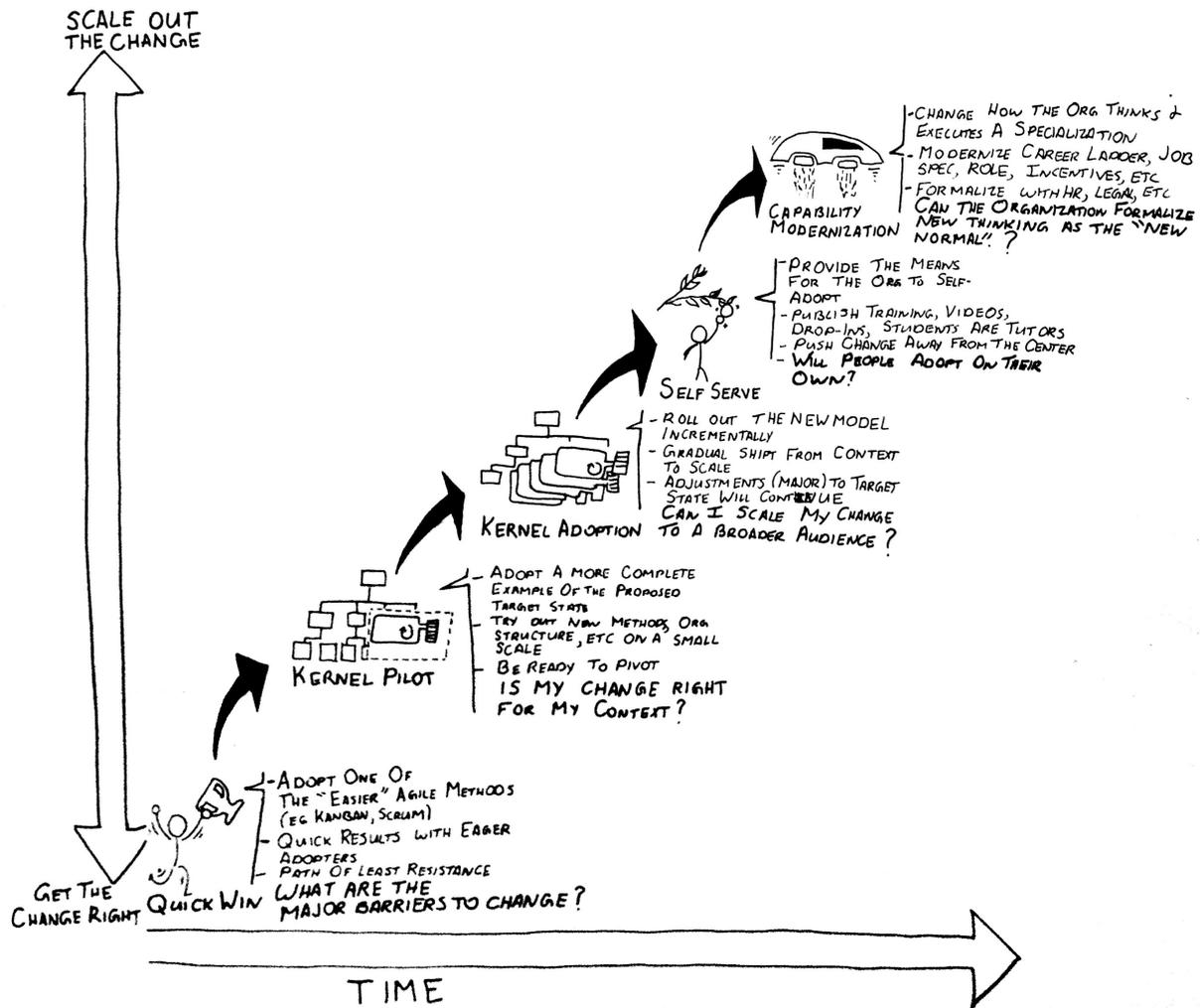
1 A Pattern Language for Agile Change

Planning and managing MVCs using a Change Canvas has allowed our team to recognize a reoccurring set of agile change “patterns”. Change agents can look to these patterns when looking for a source of inspiration when building an MVC. These patterns are represented Using the Change Canvas, providing a visual and compact way to articulate and discuss the different aspects of these patterns. The patterns listed in this chapter represent a first take at creating this kind of catalog, we are hoping that other agile change agents will be able to come up with other patterns.

If you have some other ideas, please let me know, I’d be more than happy to add them to this book, with attribution of course.

The Agile Change Patterns Shift the Focus of Learning Depending on Where You Are in Your Agile Transformation

When looking at the patterns together, they form a pattern language that helps you focus, where you’re learning should be depending on the progress your organization is made in adopting agile.



When starting to adopt agile or lean in your organization it makes the most sense to focus change on enabling **Quick Wins**. Rather than trying to validate whether you have the exact right set of target options for your exact context, focus on helping a small portion of the organization adopt a set of lightweight agile methods like Kanban or Scrum. This will identify major obstacles to change, some examples of these obstacles include an uninvolved executive, inattentive business owners, overly specialized organizational structure, or extremely poor morale. With quick wins you are trying to determine if the organization is ready to adopt “any” amount of agile. And if not, what are the major obstacles, and countermeasures that can be put in place.

Once learning increases through a successive of quick wins set of quick Win, you can start experimenting with introducing a more fulsome representation of your candidate target state. Introducing a **Kernel Pilot** involves introducing the set of organizational target state options representing the vision for the overall enterprise. This can include a number of agile methods, changes to organizational structure, and modifications to roles and responsibilities. While this change is bigger than the Quick Win, the focus here is still on learning. Focus has now changed from

learning about resistance to learning about whether assumptions behind the solution is incorrect.

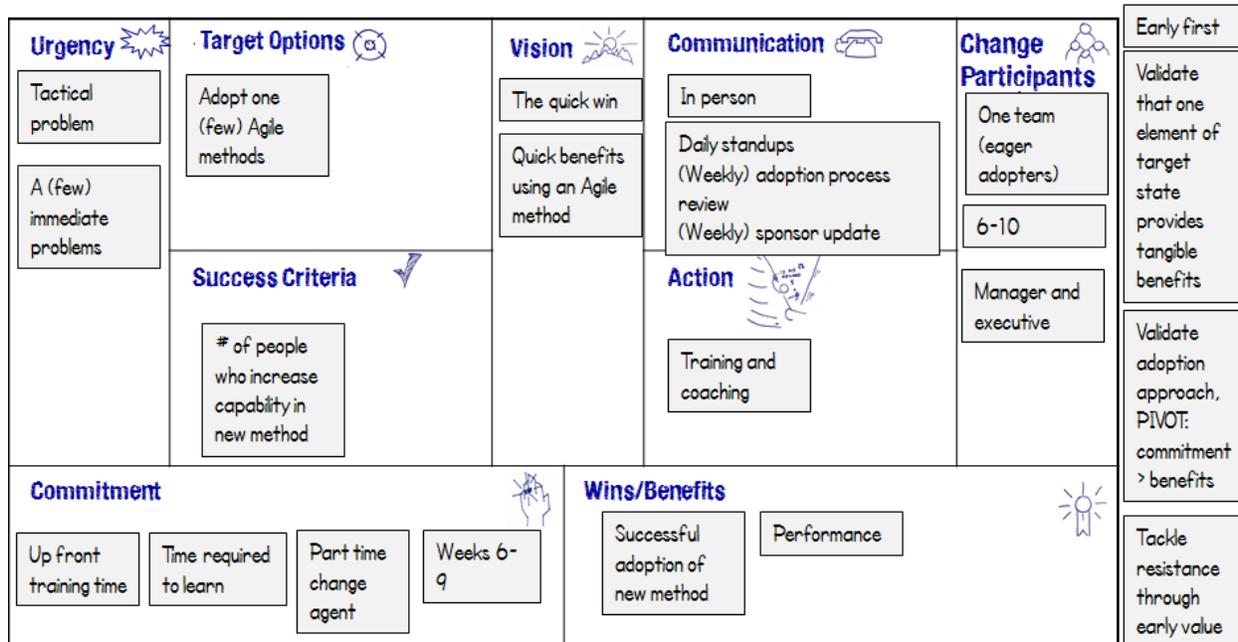
As one or more kernel pilot's are introduced into the organization, the change initiative switches from piloting to adopting. A change based on the ***Kernel Adoption*** pattern is focused on introducing the future state using successive, rolling waves. As we uncover more and more understanding of how the organization accepts change, and what approach is ideal for the organizational context, we can gradually switch learning from "what" is the target state, to "how" we can best facilitate adoption. This can be a subtle, and permeable distinction. The main point here is that it is okay to spend more time working with people and really understanding what's getting in the way of facilitating learning when starting down the road. At some point, we want to understand how to scale out our efforts to support a sustainable change plan.

Changes based on the ***Self-Serve*** pattern is another step in this direction. At this point, we should have enough understanding about what works in the organization, where we can standardize, where we can't. We should also understand what parts of our target state are relatively stable, and which parts continue to change rapidly. This allows us to start figuring out how to publish our knowledge in a way that the organization can consume on their own with minimal support from agile coaches or agile consultants. Many change management consultants recommend starting with defining some type of delivery model and method publishing it out, and getting adoption going. It's really only at this point, but I found it useful to get thoughts around "how" the organization is meant to work. At this point in the transformation, a lot of learning has taken place based on on the ground adoption, we now want to switch our learning to understand how we can push change out to the edges of the organization, supporting it in a way that creates a self learning environment.

Changes based on the ***Capability Modernization*** pattern are really about cementing the change and ingraining it into the supporting structures of the organization. New specializations and competencies are modeled as appropriate using a combination of career ladders, capability models, job specifications, and incentives. Functions like Human Resources, Finance, and even Legal play a role, making sure that employees are on board, educated, and compensated according to the new "normal (if there is such a thing as normal for an agile organization).

The Quick Win

Agile and Lean transformation are at their greatest peril when they first start. Quick and tangible results are a great way to prove the case for larger more ambitious change efforts. Early success helps mitigate resistance from naysayers, and provide the organization with confidence to invest further on more ambitious changes.



The Quick Win is typically targeted at a small number of eager adopters, an example being a medium-size team of approximately 6-15 FTEs. Typically, no more than 1-2 managers or executives would be direct stakeholders of the change. Select change participants that are capable of acting as a true guiding team, and champions for the larger change initiative.

The Quick Win is targeted at solving a few immediate and tactical problems. We are not looking at a strategic or long term return, we want to show results after a month or two.

The Quick Win is meant to be small in scope, adopting a single agile method, or a couple of closely related practices. Examples here include helping teams get started with Kanban or Scrum, or perhaps Agile Modeling.

Commitment often comes in the form of some part-time coaching following a couple of days of upfront training. The most important commitment comes in the form of time provided by change participants to learn new working habits. Ideal duration is no more than 6 to 9 week for a change modeled after the Quick Win pattern.

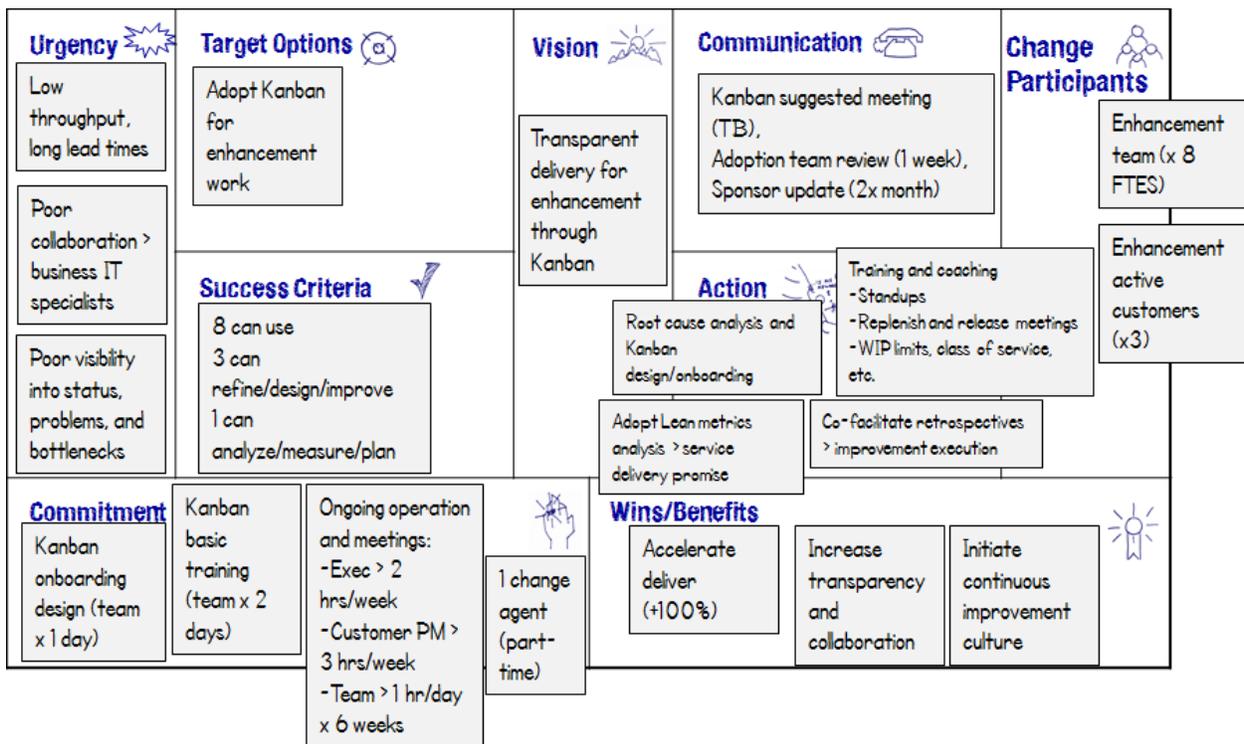
Benefits of this kind of change include change participants being successful at the new method as well as receiving some reasonable performance benefits in the short term. Success criteria is likewise measured in terms of the number of people who improve capability, and measured in terms of improved velocity, lead time, or quality.

The ideal communication approach for a Quick Win is in person and face-to-face as we want to support quick feedback. During the early stages of a change program, it is much more likely that we will need to make one or more dramatic change in direction. Enabling fast feedback is thus critical at the beginning of a change program, and is made more challenging when engaging with distributed teams.

Again, Minimum Viable Changes based on the Quick Win pattern are often introduced at the beginning of a agile transformation. We want to tackle resistance to the overall change program through delivery of early value.

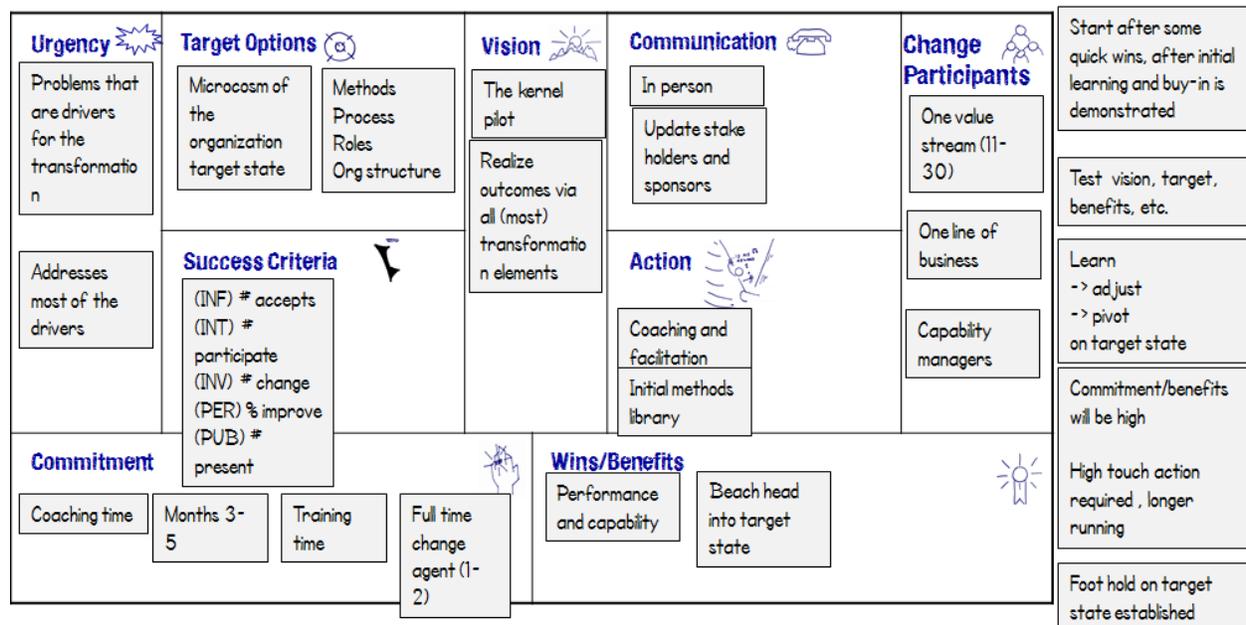
The Quick Win is a good way to validate an isolated component of a larger transformation. This type of change is also really good for validating adoption tactics, as well as determining if the relationship between commitments and benefits are remotely correct. Be ready to pivot on this point, as the commitment to benefit ratio is extremely hard to determine upfront.

Below is an example, a favorite Quick Win of ours, improving business agility through adoption of the Kanban method.



Kernel Pilot

After a number of Quick Wins have been implemented within an organization, there can be enough momentum to execute a more ambitious change, one that more broadly reflects the organization's suggested true north. Previously implemented Quick Wins should have provided some validation on different aspects of the potential target options, change actions, and required effort, reducing the risk of attempting something a little larger.



A Minimum Viable Change using the Kernel Pilot pattern is often targeted at an entire value stream, from intake to support, and all the steps in between. Often this can be represented as one “line of business” within the organization. It is crucial to involve managers and executives, both from the perspective of capability management as well as issue escalation and resolution.

The Kernel Pilot should address a significant portion of the problems that are also drivers for the overall agile transformation. We want to evaluate the validity of the agile transformation, but on a smaller scale.

Likewise, the target options for this Minimum Viable Change should serve as a reference point for the desired state of the overall organization. Once this Minimum Viable Change is complete, a discrete segment of change participants (i.e. a team or set of teams) will now be using new methods, possibly in a flatter, more network-based organizational structure, potentially leveraging new, wider role definitions.

Immersive coaching, facilitation, and even embedding outside expertise within delivery teams is often required to successfully help change participants shift to suggested target options. As the choice of methods and techniques become stable, some effort can go into developing and socializing a light weight methods library.

The larger scope and scale of this type of Minimum Viable Change means that a bigger commitment is required. Our experience is that it can take anywhere from 3 to 6 months to execute, and may require 1 or even 2 full-time change agents.

Benefits come in the form of improved performance and improved capability, but also in the form of validation; the Kernel Pilot can be thought of as a beachhead for the larger organizational agile transformation, and is the first cohesive step into this new direction.

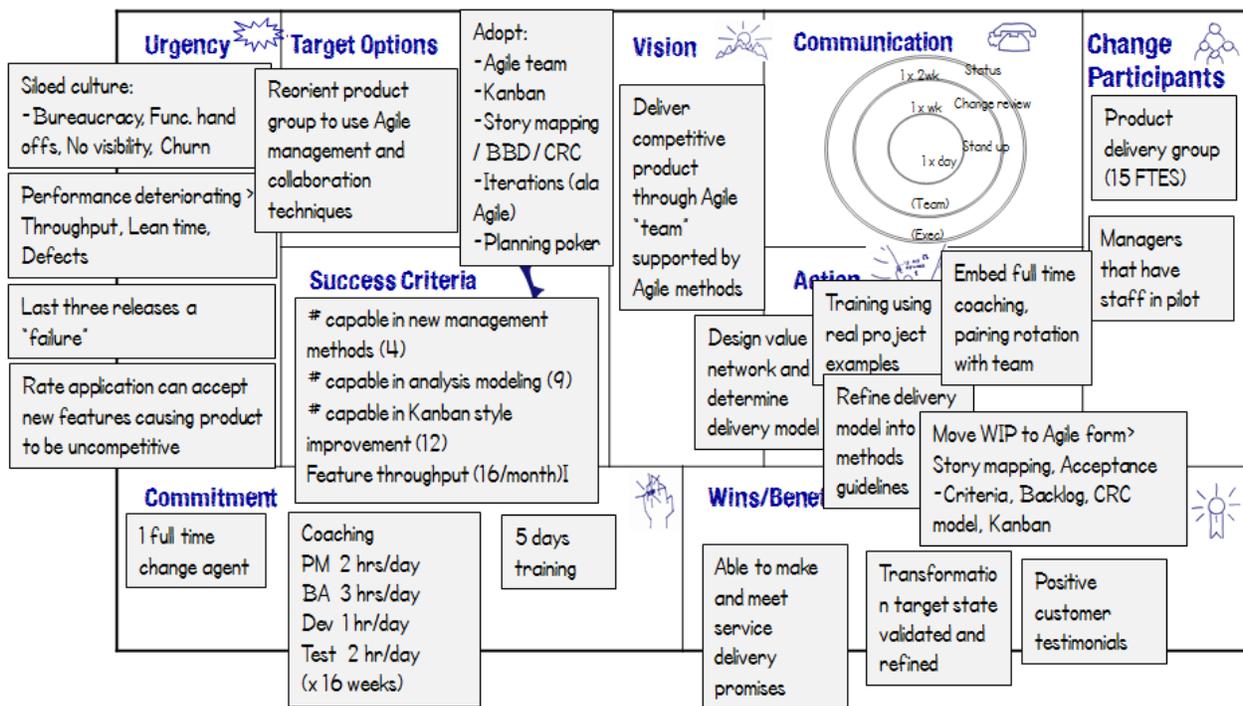
Again, it is often safer to implement a Minimum Viable Change based on the Kernel Pilot after

one or more Quick Win style changes have been successfully executed, this will reduce the risk of adopting the wrong kernel.

As this change reflects the first time that different components of the target options have been adopted in an integrated way, there will still be significant learning. Expect to adjust on all elements of the change, a change pivot may even be required, depending on what is learned.

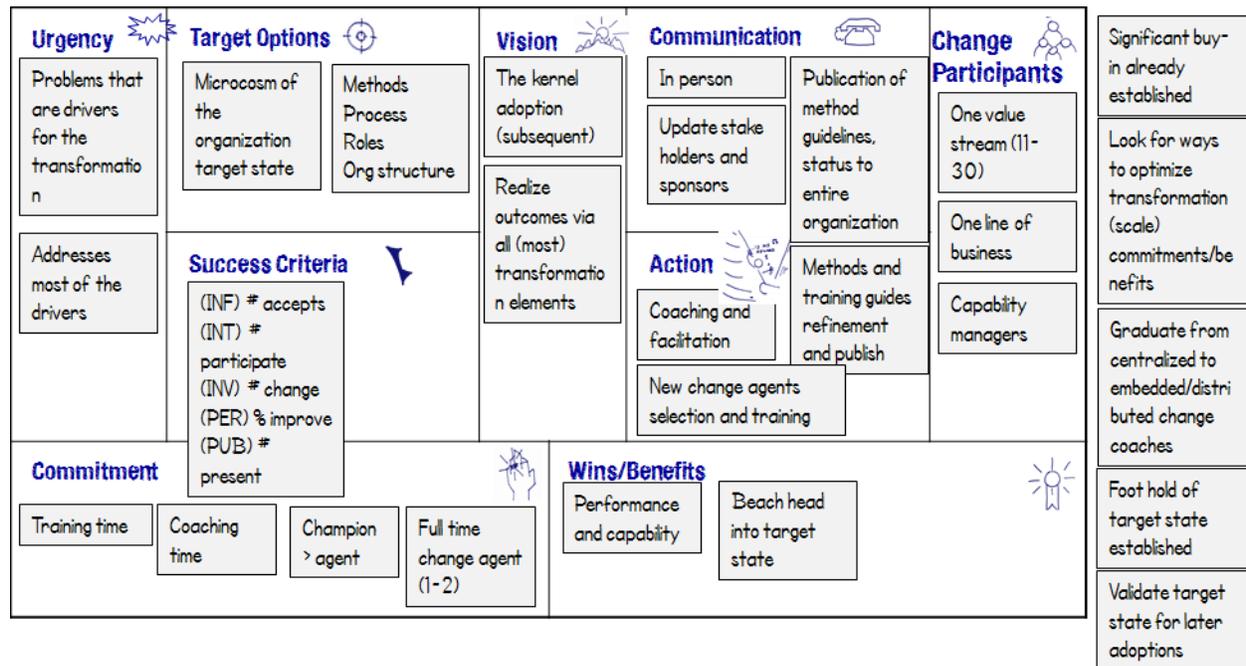
The relationship between commitments and benefits may seem off at this point in the transformation. This is because a significant amount of learning is still taking place on behalf of both change agents and change participants. Don't panic if a lot of high touch activities are still required to make progress at this point in the change. Optimization of learning effort across the organization can take place later. Remember that the purpose of a change using this pattern is to try to get a first foothold on the future target options of the organization.

Below is an example taken from our real world experience, where we helped knowledge workers within one product delivery group to adopt a cross functional, co-located agile team model supported by a number of management, planning, and modeling methods. This agile "suite" was our suggested stack for the target options of the majority of the organization. Implementing this change on a smaller group allowed us to refine the stack as well as evaluate assumptions behind the change program.



Kernel Adoption

As one or more Kernel Pilots are implemented within the context of a single agile transformation, the transformation can evolve out of pilot mode and into adoption mode. An MVC following the Kernel Adoption pattern is less concerned with validating aspects of the change solution, and switches focus to trying to optimize the rate of adoption. As a transformation progresses, we want to modify the kind of change support provided by change agents, graduating from high touch/high support tactics to ones that can scale out. Examples include self-study, peer-to-peer mentoring, and improvements driven solely by change participants.

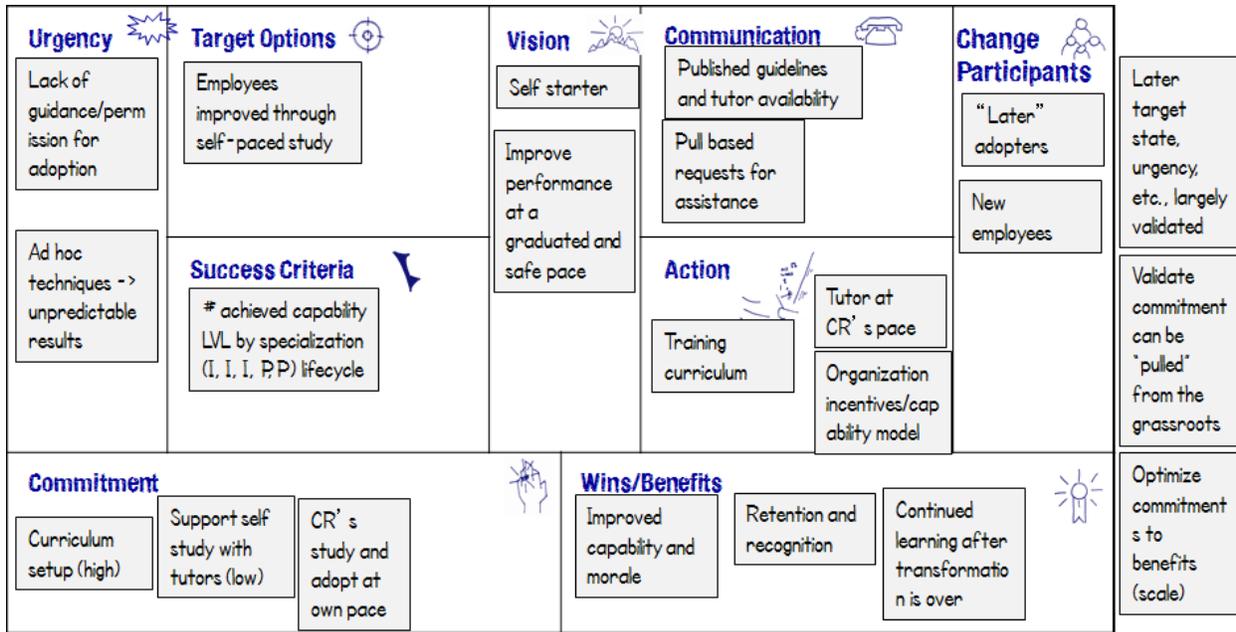


As an agile change is rolled out across the organization, more effort can go into refining, publishing, and socializing training guides, methods, and other tools. The idea is to help knowledge spread and learning scale to a larger audience. Change agents spend more time training other members of the organization to be true change champions, agile knowledge experts, and owners of methods and tools. High touch, in person coaching can now start to be graduated with other learning methods that can scale out to the entire organization, including self training, online knowledge repositories, and communities of practice.

The target options for an agile transformation is never “done”, so some aspects of the change solution will continue to require validation and may require a change in direction.

Self-Starter

A Minimum Viable Change modeled on the Self-Starter pattern attempts to provide an opportunity for change participants to guide their own adoption. Various improvement methods such as training material, peer working sessions, scheduled times for change participants to attend “tutor drop ins”, and mentorship programs are used to allow employees and management to self select their learning pace.



Both new employees as well as those targeted for later adoption will require the means to educate themselves on how the organization is trying to deliver, sometimes long after an agile transformation is considered “complete”.

A Self-Starter can also be useful in that it provides employees with permission to start moving to new approaches in advance of any “scheduled” adoption plan. The Self-Starter is also considered “safe”, adoption takes place at the comfort level and pace of change participants. Sometimes, curriculum and pull-based tutoring needs to be complemented with organizational incentives to adopt. These incentives can be as simple as recognition for employees who have achieved a certain level of capability.

Supporting self learning can take more effort to set up initially, especially if new training material and/or capability models are not already available. Some internal marketing to change participants may also be required as the Self-Starter kicks off.

It’s often recommended to support this effort with at least some kind of part-time tutoring. This tutoring can be scheduled, or serviced on an as needed basis. When supported properly, this kind of change scales well, and more than makes up for the initial effort. Besides providing the organization with the improved performance that comes with better capability, heightened morale

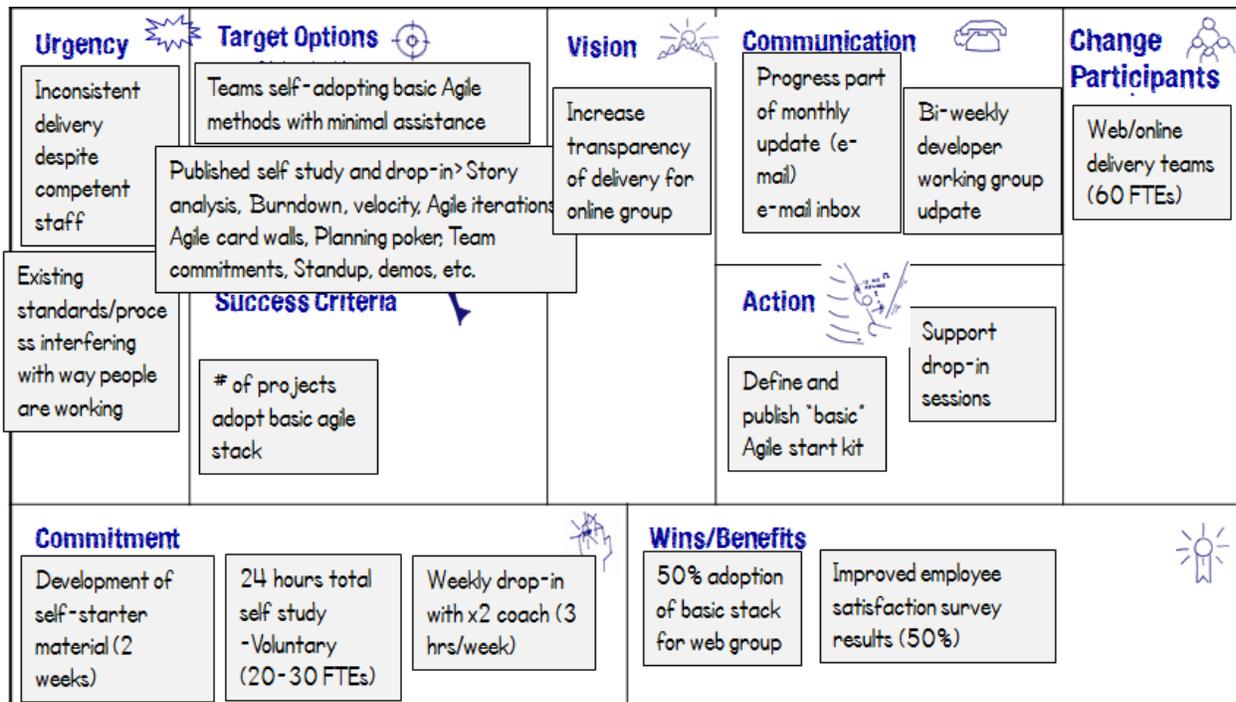
can lead to better employee retention. A good Self-Starter program will continue long after the agile transformation is considered “over”.

As stated previously a Minimum Viable Change following this pattern is better introduced later within the context of an larger agile transformation. The overall approach and applicability of specific methods should have largely been evaluated thanks to one or more previous MVC’s.

Self-Starters are good at evaluating whether employees within the organization can improve without dedicated change agent support, and “pull” help when and if required. Change agents should also be looking for ways to optimize the benefits gained compared to the effort required on the part of change agents.

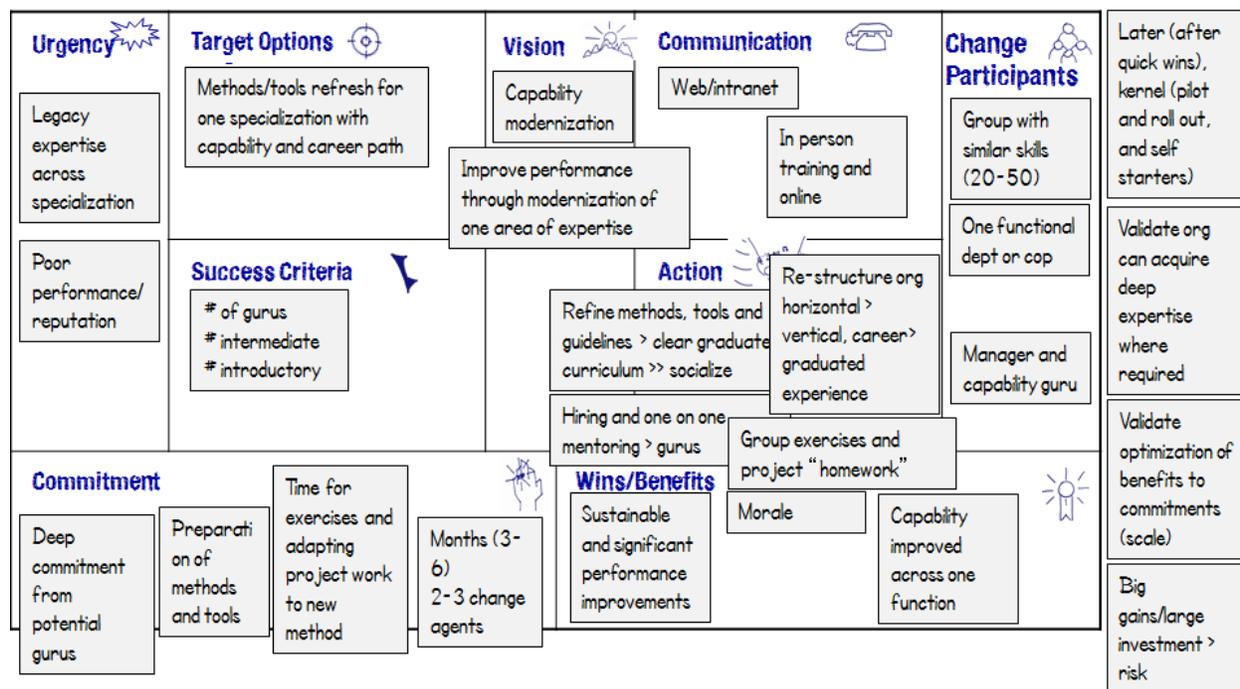
The example below is taken from an agile transformation that I was a part of. Midway through the transformation it became clear that we did not have enough coaches to support the demand to assist various teams in adopting agile methods. We had already conducted a number of pilots across the organization and had enough training courseware and other material to support the notion of a self-starter. This provided an avenue for employees who wanted to get started with agile but did not want to wait for dedicated coaching assistance.

This self-starter consisted of some guidelines and training exercises that would get employees familiar with basic agile methods such as iterations, using user stories, and collaborative planning. We published this material to the corporate intranet, and announced a weekly 3 hour drop-in where anyone interested could receive coaching and assistance and have their questions answered relating to any of these techniques.



Capability Modernization

True improvements for many organizations will mean taking an honest look at current capability, methods, and techniques and revitalizing them to support business agility. A Minimum Viable Change following the Capability Modernization pattern is focused on improving delivery and/or management techniques for a functional capability within an organization. In more traditional organizations this often mean a functional department, in more modern agile organizations this will mean a collection of employees who can fit into a particular role. Examples include retooling all of the developers within an organization, or providing training and coaching on agile management techniques for all managers and executives.



A change following this pattern typically targets a larger set of change participants, i.e. all employees that fit within a similar role (e.g. all testers). Any managers responsible for building capability for that role will also be similarly impacted.

Organizations tend to consider the Capability Modernization pattern when things are truly “broken”, expertise across the function is considered to be legacy at best, and the current level of maturity is causing a noticeable and growing impact in performance. These issues are having a visible impact on business outcomes.

The target options for this pattern is a complete refresh in terms of capability for that role, including career path, required skills, training and other tools and accelerators.

This considerable investment may involve restructuring the organization from a more traditional specialist model to a specialist/generalist hybrid, fostering both collaboration and cross functional

tasks sharing necessary to meeting demand variation and complexity . Action can also include rethinking how careers within the role can progress based on expertise.

A clear and graduated curriculum is often developed, and then introduced onto projects using a rolling wave approach. An effective approach can be to require that all training homework be conducted on real project work.

Effort is focused on building a number of “capability gurus”. These capability gurus act as knowledge leaders and method owners of the new capability. A combination of hiring externals, and mentoring existing employees is used to help build this team of capability gurus. These capability gurus act as senior keepers of the flame, and are responsible for ensuring a continued commitment to excellence, and pursuing the kind of quality found in the best of agile organizations.

This type of Minimum Viable Change requires a lot of investment on behalf of the organization. Staff wishing to become gurus need to commit the time required to master new skills to the point where they can train others. Gurus need to be carefully selected, not everyone is cut out for this kind of dedication.

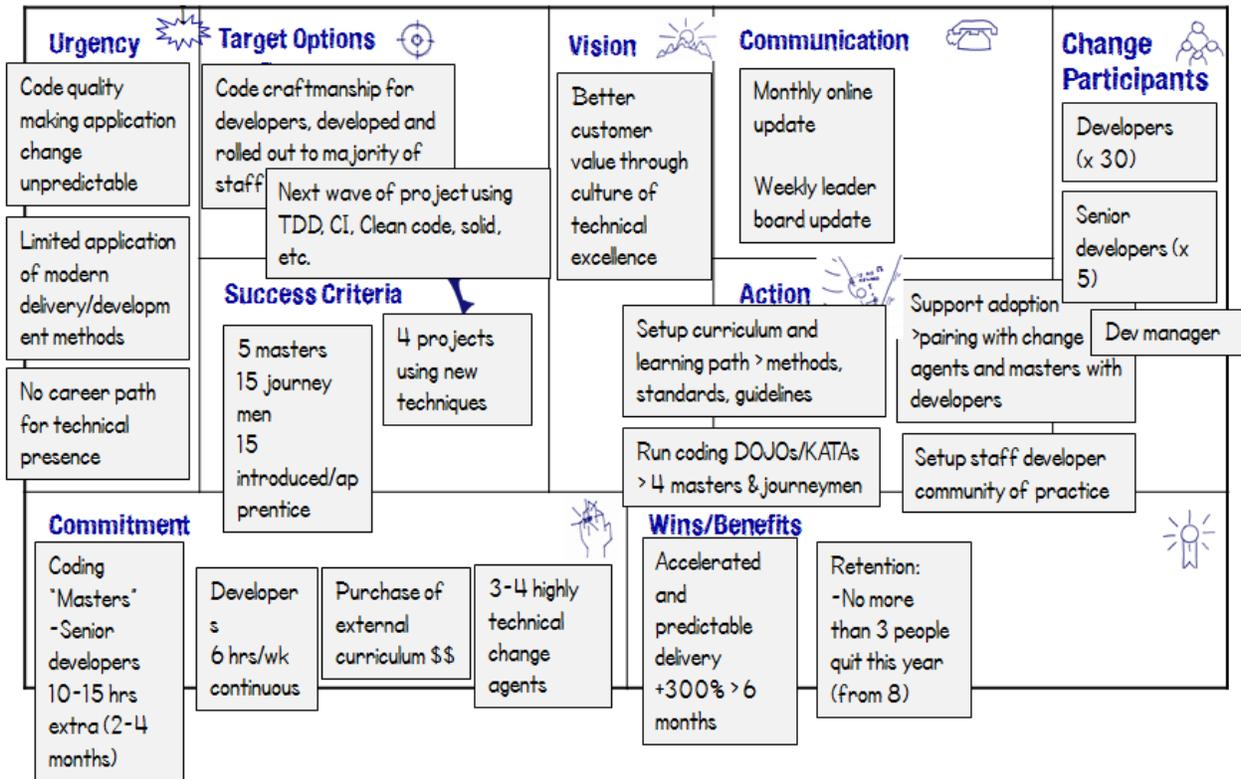
Methods and tools, and the training curriculum to support them need to be adapted to the context of the organization. Training capability also needs to be developed. This is also a significant investment.

Finally, deep and meaningful change in terms of adopting new methods will take time, this type of change will typically take many months to execute. Several full-time, and senior change agents are often required to support a change following the Capability Modernization pattern. Change agents must have years of experience in the selected methods and techniques.

When this type of change is done correctly, benefits to this type of organization are significant. Improvements in capability lead to higher quality, which lead to significant and sustainable performance increases over time. There are no real shortcuts to improving delivery outcomes. Equally important, is that this type of change shows a willingness to invest in people’s careers and in their capability, which improves morale.

In the context of a larger agile transformation, a change following this pattern should be done later, a number of Quick Wins, Kernel Adoptions, and even Self Starters should have already been introduced to the organization. These kinds of changes are smaller, require less investment, and are better at providing feedback on what works and what doesn’t. Before starting a Capability Modernization style change you will want to have already validated most of the assumptions behind your transformation. Key learnings from this style of Minimum Viable Change will come in the form of understanding how we can scale out the change to the rest of the organization.

A good example of this kind of change would be trying to achieve a state of “technical excellence” through adoption of methods that can be found within the software craftsmanship movement. This includes techniques such as test driven development, SOLID development techniques, continuous integration and continuous deployment.



Leverage Patterns to Build a Minimum Viable Change

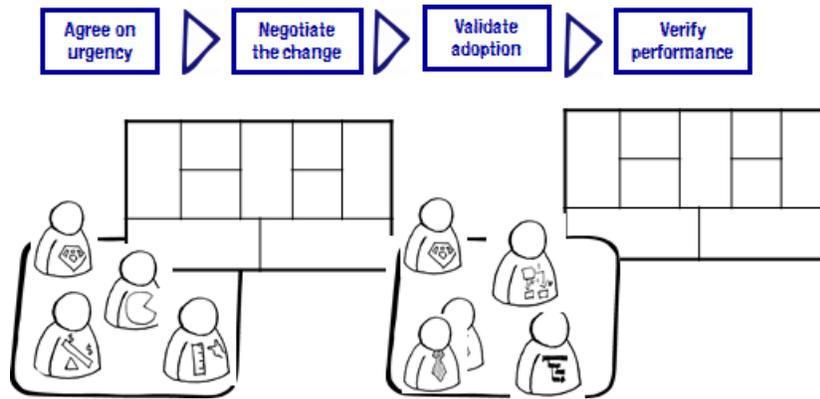
1. When designing your next MVC, consider adapting one of the patterns described in this chapter. How much of the pattern needs to be modified to suit your context?
2. If running a larger enterprise transformation, consider how far along you are in this journey, and select an appropriate pattern based on your progress. Have you passed the quick wins stage? Are you ready to pilot a consolidated kernel of the target options? Are change participants ready to self- start?
3. Are there any other patterns that you can think of that you could add to this collection? Please talk to me and I'll add them to the book with attribution to you of course!

Overall Cadence Model

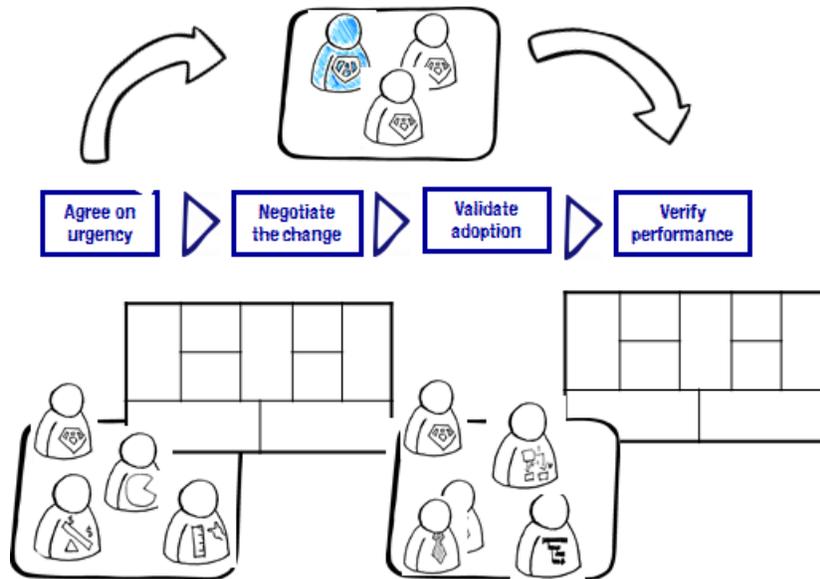
Large-Scale Transformations Require A Lot Of Coordination

Organizational transformations can require a lot of coordinating activities, if we look at transformations using the Lean Change method we could see the following:

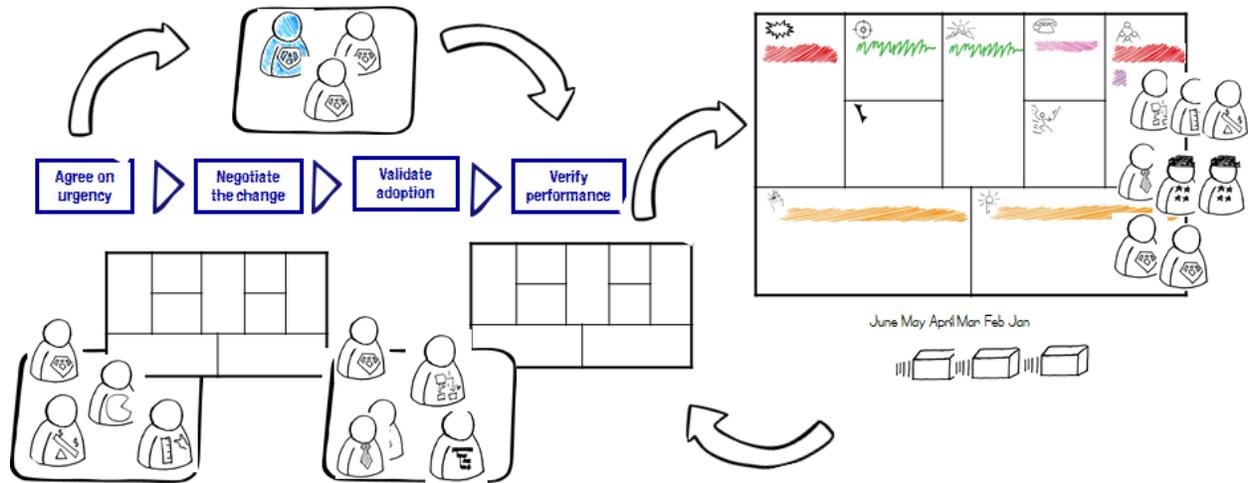
Individual change agents will collaborate closely with a change participant segment, one for each MVC being run.



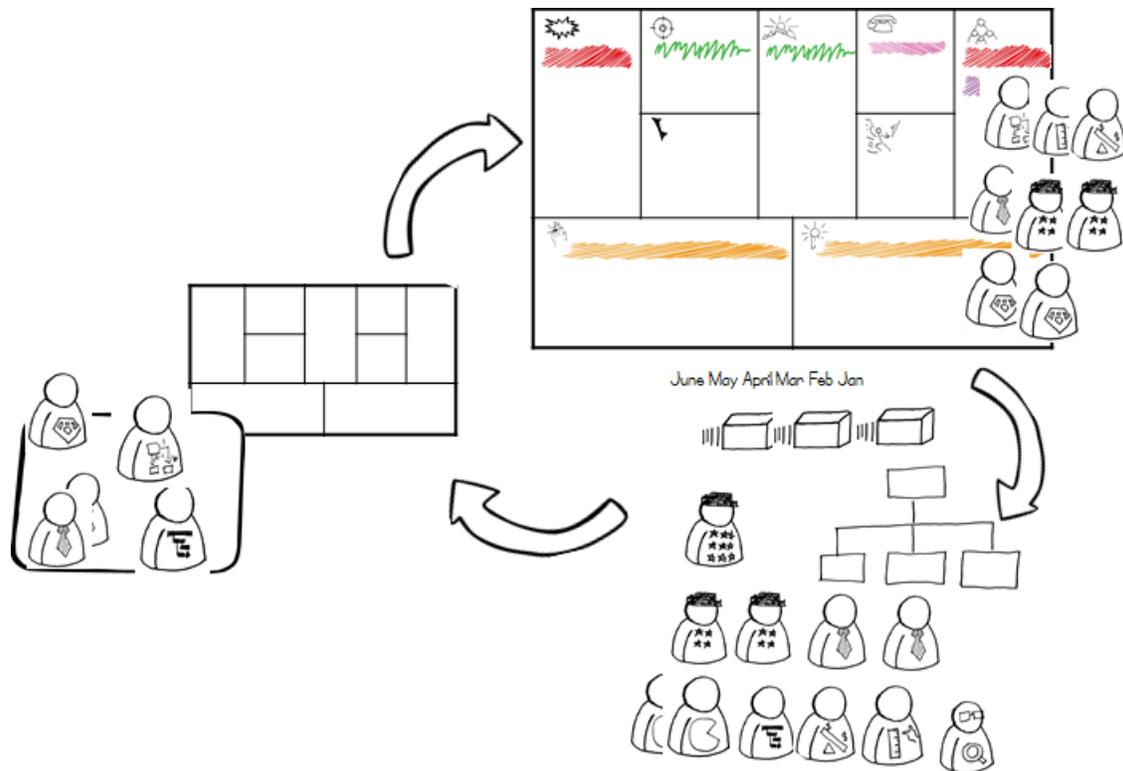
Change agents will meet to coordinate and learn from each others experiences in the field.



Improvement experiments will causes MVC Canvases to be updated, which may in turn cause the Transformation Canvas to be likewise modified. This may require running Transformation Canvas workshops. Changes to the Transformation Canvas will likewise trickle down to other MVCs and related experiments.



Modifications will need to be validated and socialized with other change agents, stakeholders, sponsors and the rest of the organization. Feedback will need to be incorporated to affected artifacts. All feedback will require further communication.



The Need for a Cadence Model

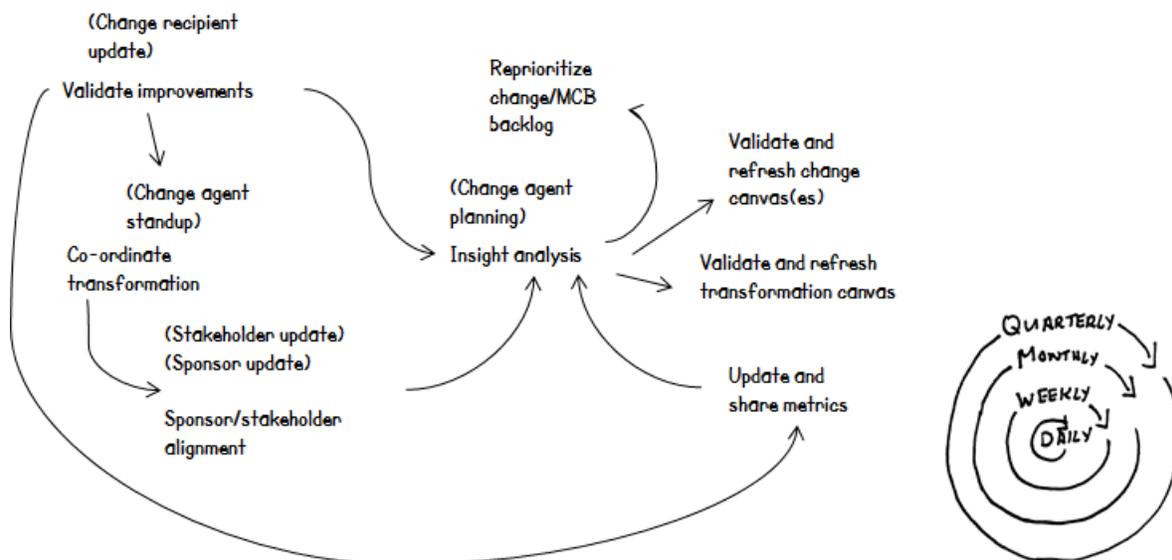
One way to manage the complexity of running a large-scale transformation is to set up some reoccurring workshops and sessions. The sessions and workshops make up a transformation

“cadence model”, defining the synchronization heartbeat for the change initiative.

Transformation Cadence Model Information Flow

The example that I’ll provide below comes from one of my experiences running a complex transformation, the organization had a lot of moving parts.

I encourage you to come up with your own cadence model as the choice to use some, none, or all of these sessions is left up to the discretion of the change agent team. Care needs to be taken to balance transformation needs with process complexity and overhead.



Change participant Update- Change agents spend dedicated time validating improvements for a particular MVC with their change participants.

Change Agent Stand Up - Change agents coordinate with each other, preferably using an agile standup style meeting

Stakeholder/sponsor update - Stake holders and sponsors are provided updates based on the information gained through implementing the change

Change agent planning session - change agents perform deep analysis to understand if the change is progressing on the right track.

In this example our team elected to run a two hour planning session with two distinct agenda items.

The first agenda item was to *analyze any insight* gained since the last planning session. Validating improvement experiments, running change standups, and getting feedback from sponsors and stakeholders, all contribute to generating insight. This insight would result in changes in direction, and likewise modifying change tactics. Changes to various artifacts would be required, such as modifications to the Transformation Canvas, MVC Canvases, and Improvement Experiments.

The second topic of the planning session was to coordinate changes to impacted Lean Change artifacts. Changes could include:

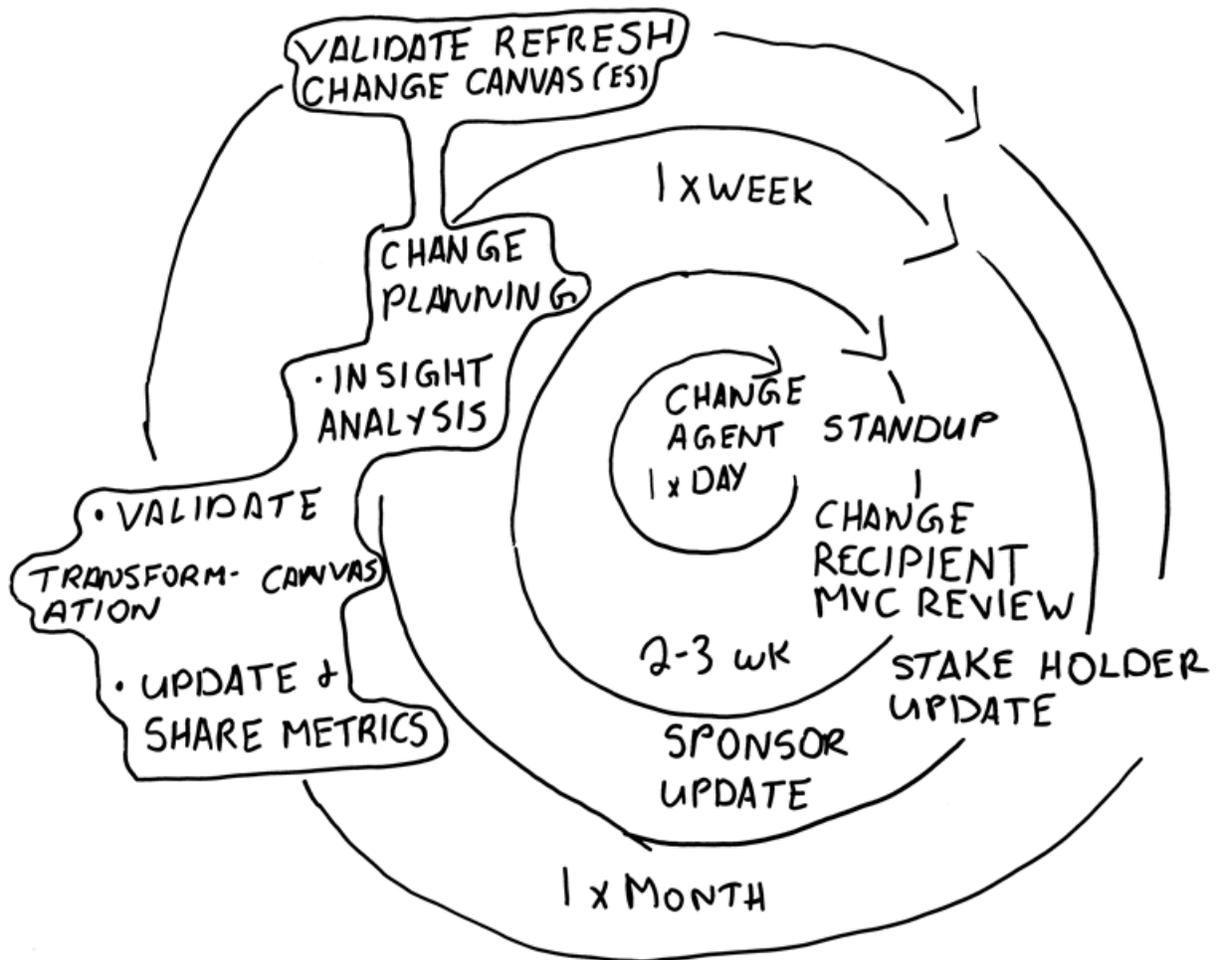
- reprioritizing the backlog of MVCs
- refreshing various MVC Change Canvas to ensure that all sections contain valid assumptions
- Refreshing the Transformation Canvas so assumptions were also valid
- ensuring that metrics and charts that illustrate aggregate performance, success criteria, and capability reflect the latest information.

Our team elected to hold the change planning session weekly, with the first part of the session dedicated to generating and analyzing insight. The second part of the session would focus on updating a particular type of Lean Change artifact, rotating the artifact of focus, on a weekly basis. This meant that each type of artifact was investigated for a potential refresh once a month (see the cadence model below)

Transformation Cadence Model Suggested Timing

As stated previously I recommend that each activity be conducted according to a set cadence, this makes it easier to coordinate various activities, and helps the change agent team establish a steady rhythm for the transformation.

Here is a cadence model that accompanies the above example, again modify it to support your particular context.



Come up with Your Own Cadence Model for Meetings, Review Sessions, and Workshops

1. Determine the overall flow of information for your transformation, remember no two transformations are alike, and you need to customize something that works for you and your initiative.
2. Determine what tactics you want to do to synchronize this flow of data, consider a combination of information radiators and explicit working sessions to help update various artifacts, get feedback, and communicate those updates to all change stakeholders.
3. Select a cadence model that describes the overall heartbeat for the specific activities. Will specific activities reoccur on a daily, weekly, monthly, or even quarterly basis? Will these activities take place just in time? Balance trade-offs between effort and process overhead with the need to communicate progress and enable organization wide learning.

Appendix

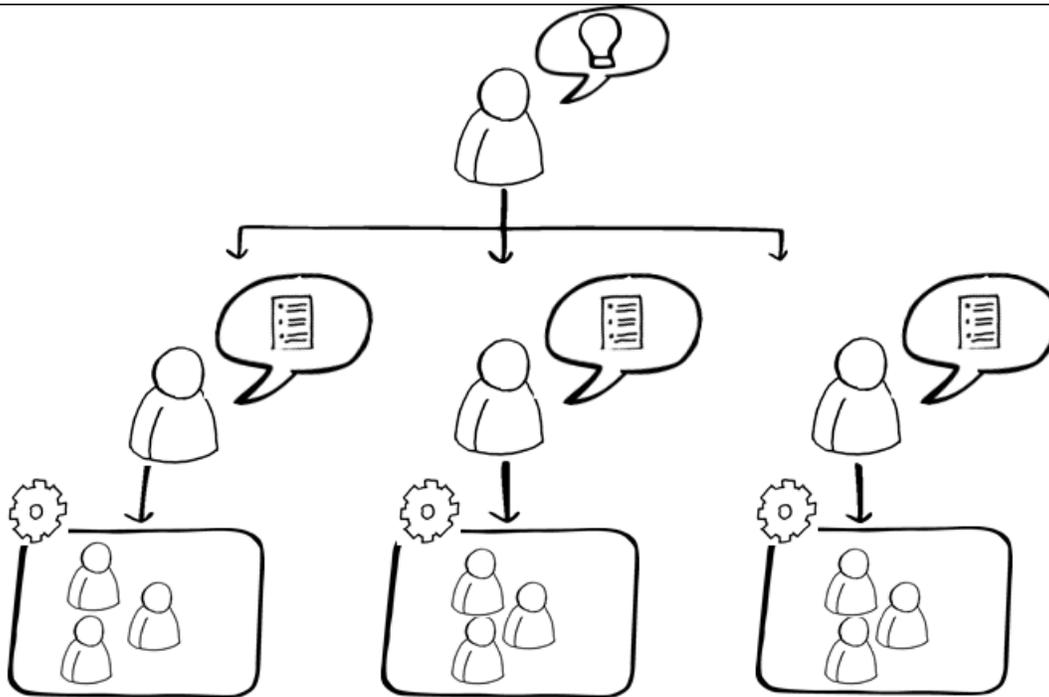
Traditional Approaches to Technology Delivery Is No Longer Suited to Today's Market

Traditional IT Management Methods Borrow Much of Their Thinking from the John Taylor School of Thought

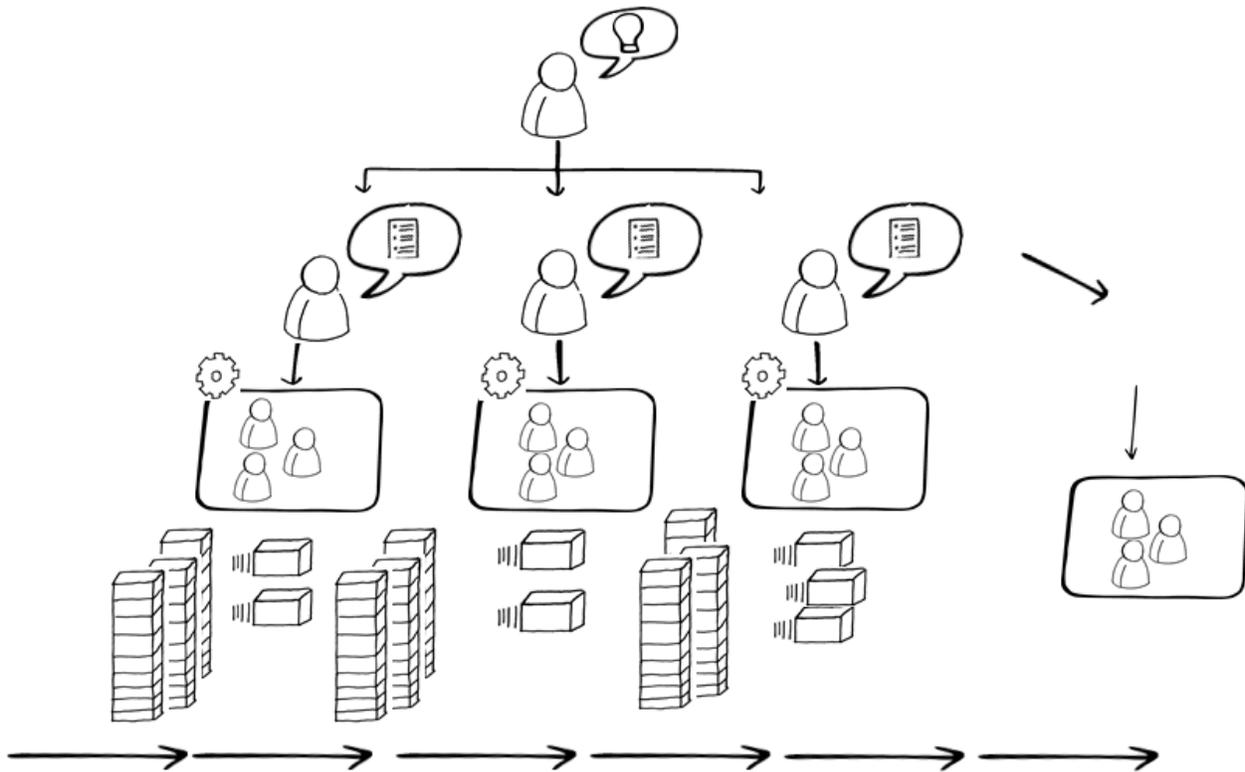
In the 1930s, this philosophy was based on the premise that organizations would be much more efficient if resources were organized by specialization. This would allow functional oriented departments to focus on efficiency through highly standardized and repetitive activity.

Activities were planned and coordinated through the use of functional managers who ensured that individual employees received adequate instruction and feedback on performance targets. Managers got their orders from executives who were responsible for determining the objectives of the organization.

In this approach, big ideas came from the executives and owners of the business, who determined what the organization should be doing and how it should be doing it. Managers were responsible for ensuring successful execution of plans. Individual tasks and activities were doled out to employees, and all coordination between functional departments required intervention by the management layer. The majority of employees were essentially cogs in a well-orchestrated machine; they did the work and were not required to perform any meaningful thinking.



Scaling out an organization simply required the addition of another functional layer. Organizations could successfully increase in size by expanding out horizontally and vertically, creating a more nested hierarchical structure.



One reason this approach works well is that managers and leaders were able to optimize the execution of highly specialized and repetitive tasks. This allowed workers to leverage economies of scale, lowering total cost of ownership for both in-process and complete inventory of goods.

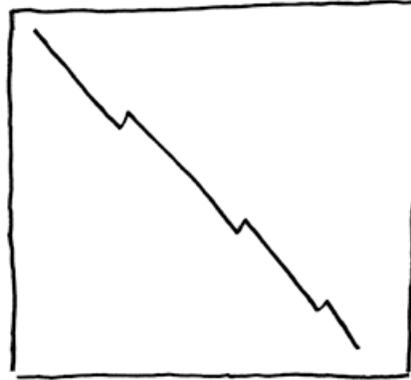
This Traditional Thinking Is Suited to a Previous Era, One of Economic Scarcity



Success relied on maximizing efficiency and providing goods and services to customers at the lowest possible cost.

This low cost was achieved through a command and control approach, meticulous planning and

coordination, and the development of robust standards and procedures that carefully lay out the most efficient way to complete a particular task.



Work was organized to leverage economies of scale; specialists were grouped together into functional departments, highly repetitive activities could be completed in large batches, and passed from one specialist group to the next. Customer demand was serviced in large quantities, again using a big batch approach.

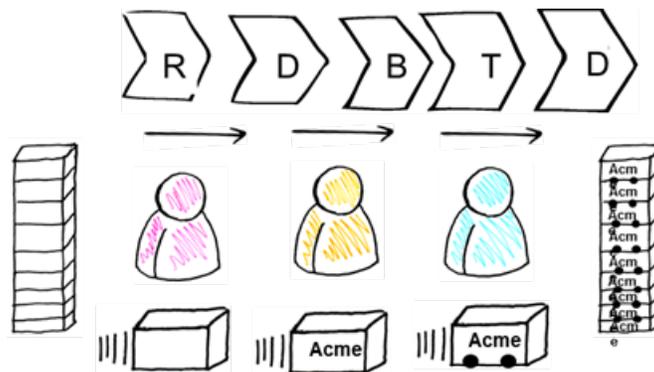


In a market where customer wealth was low, this approach effectively serviced customer demand. Economies of scale were easy to leverage, as a small number of products could be introduced to a large and undifferentiated customer market. Product lifecycles were also extremely long, taking years or sometimes even decades before one product would be displaced by innovation.



You can have any
car you like as
long as it's black!





Today's Business Environment Requires a Radically Different Approach

For many organizations today, market success is achieved through offering differentiated products and services to customers who can and are willing to pay a premium for the latest innovation.



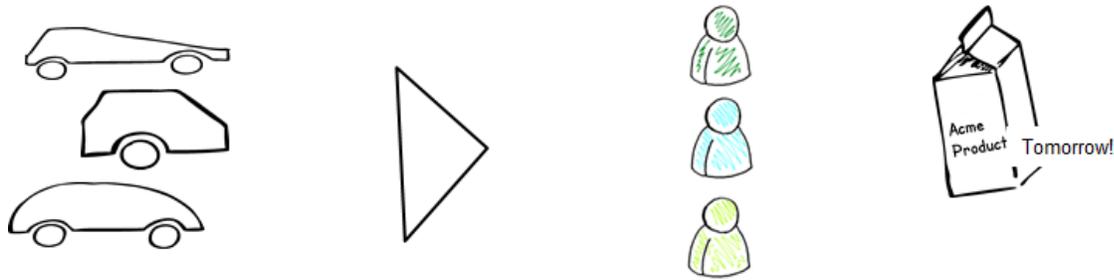
In this environment, organizations can be incredibly efficient, follow plans to perfection, and be incredibly effective at coordinating thousands of specialists, but still fail as a business.



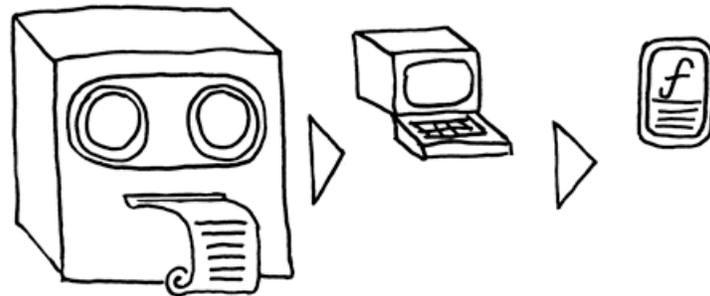
In today's challenging business environment, the biggest risk has shifted from building products cheaply to building a product that nobody wants. To borrow a line term from the lean startup community, the question becomes not "can I build it, but should I build it?"



Successful enterprises no longer provide a one-size-fits-all offering to their customers. Instead differentiated products and services are made available to a variety of well researched customer segments, in some cases products are offered as platforms to be uniquely customized to the individual end-users needs. Product lifecycles are now much shorter, with obsolescence coming in months, weeks, and even days.



In situations where organizations are still providing more commoditized goods and services with longer life cycles huge efficiency gains can be leveraged by taking advantage of the latest wave of technology innovation.

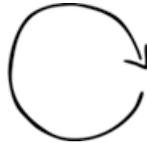


In this environment speed of execution becomes more important than cost of execution, and processes, organizational design, and management methods need to be designed to support speed.



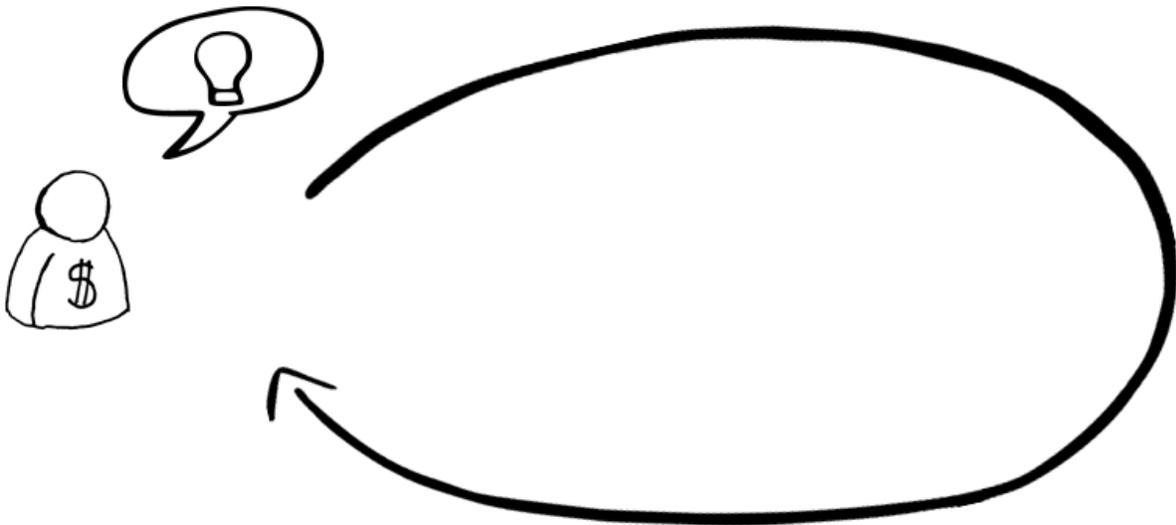
The most fundamental change between traditional management methods and ones that leverage lean and agile methods is the shift from plan driven processes to learning driven processes. In an environment where customer demand as well as the tools used to service the customer demand are constantly shifting, long-term plans, static processes, and economies of scale will work against you.

Instead, organizations need to be designed to manage feedback, and be able to adapt to constant change.



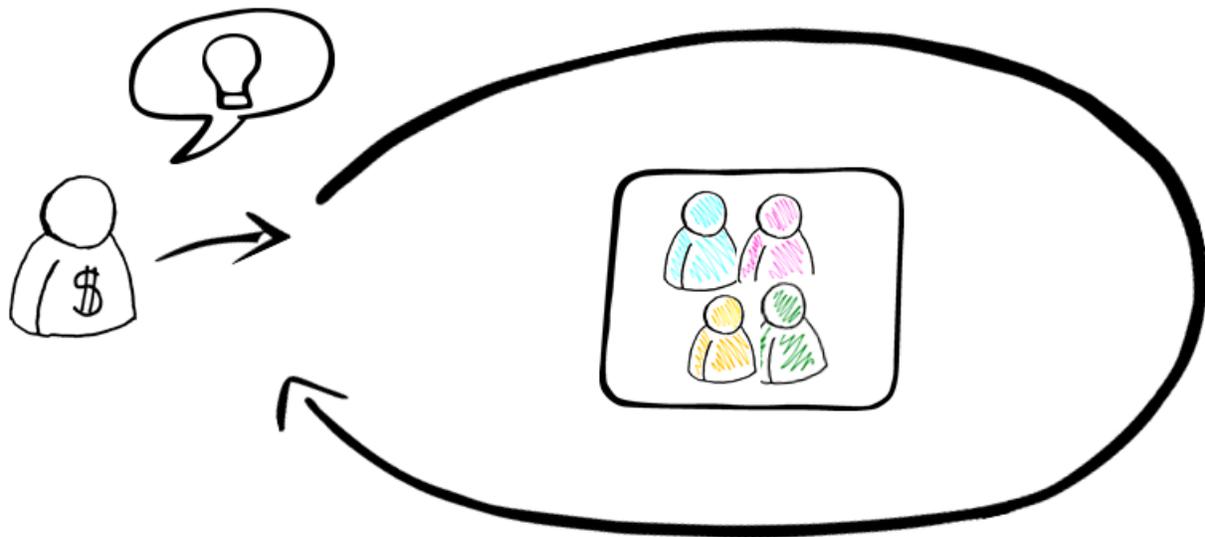
Agile and Lean Provide a Vision for Success in Today's Complex, Customer Driven World

Lean and agile are concepts that cover a variety of principles, methods, practices, and perspectives. Different approaches have unique and sometimes contradictory viewpoints. Nonetheless there are some fundamental differences between how an agile or lean organization operates and one using more traditional management methods. Perhaps one of the most important differences is that agile organizations attempt to achieve business agility by creating a frequent feedback loop with its customers. Agile methods do this by delivering work in short iterations, where working software is delivered in iterations that last between one week and one month. Organizations following a lean approach achieve this feedback using a just in time, pull-based approach, one that limits work in process to the organization's capacity.

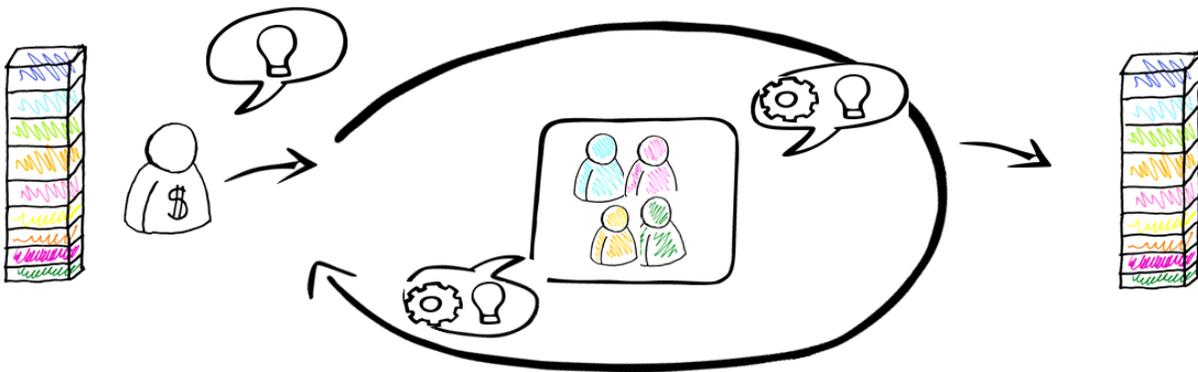


This feedback loop provided by frequent delivery enables the organization to continually learn based on the success of the last delivery. This feedback allows organizations to delegate the majority of decisions to the team level without fear of the organization slipping into dysfunctional behavior.

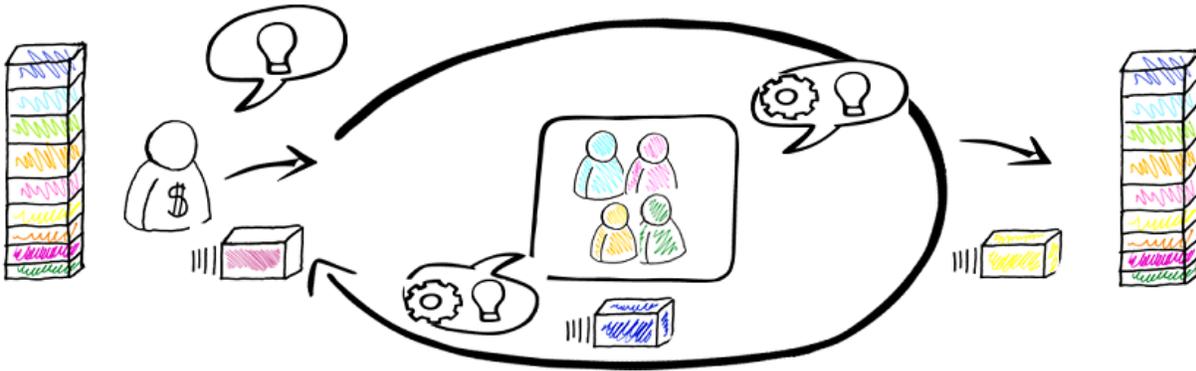
Teams are better able to learn, and better able to deal with complexity when they are made up of a diverse set of individuals. Most lean and agile methods recommend that teams are cross functional and contain the majority of skill sets that are required to consume customer demand and create a finished product.



This leads to another key principle in both the lean and agile world. In this model workers are responsible for both doing the work and coming up with the good ideas around how to complete the work. All team members are encouraged to be thoughtful about what they are doing, and challenge why things are being done a certain way, or why they are even being done in the first place. In this paradigm, workers are self-organizing, and the teams are largely self-managed.



Because feedback is critical to enabling this learning system, work is deliberately processed in small batches. The whole notion of economies of scale is discarded in favor of working with the lowest possible level of inventory. What this means is that workers are encouraged to work on only a one or two business value tasks at a time, and work on those to completion rather than trying to stay busy by working on many things in parallel.



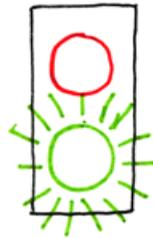
Frequent client delivery, cross-functional teams, empowered workers, and limited work in process all contribute to a system of continual self-correction and learning. Knowledge workers not only just get better insight into what customers want, but also learn how to optimize their internal methods, tools and processes to improve their own internal efficiency, speed and quality.

Another key difference between a traditional organization and an agile oriented one is the way that lean and agile teams look at quality, processes, and standards. Workers in this environment need to be constantly learning, and adapting.

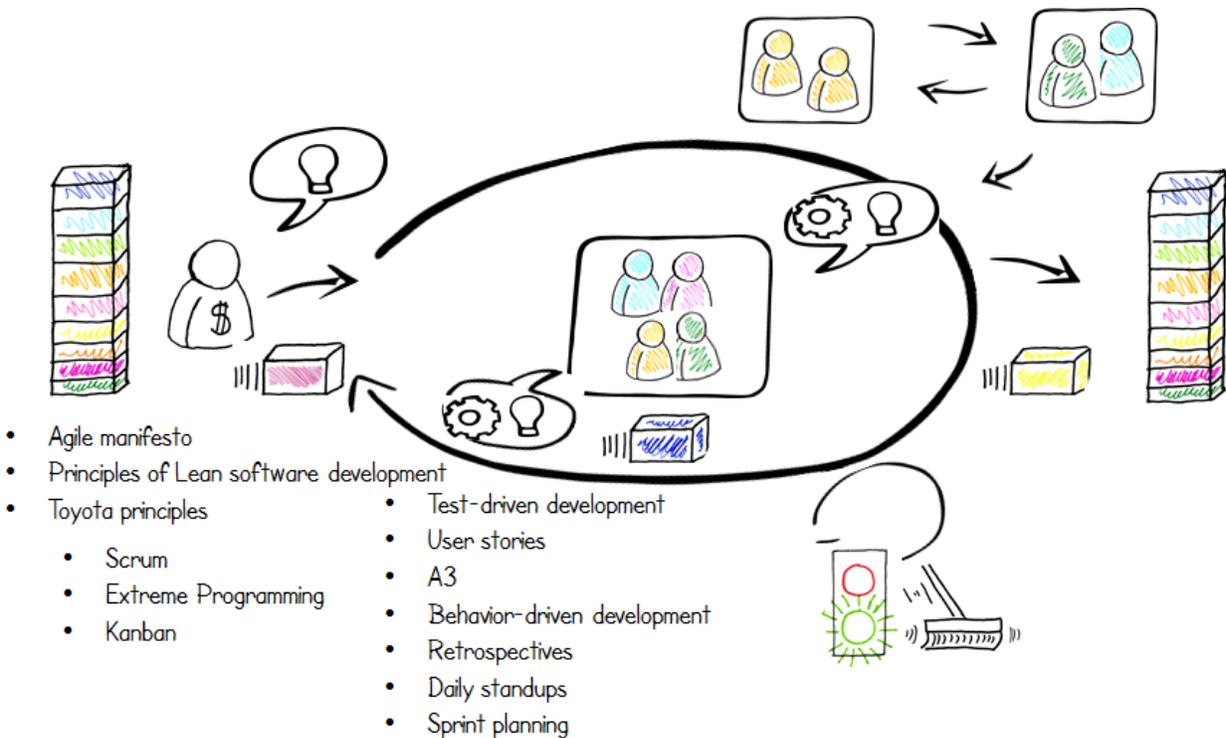
Poor quality interferes with this learning cycle. Both lean and agile methods recommend that quality be built into the process, rather than inspected for after-the-fact. Whenever a problem in quality is found, root cause analysis is conducted to get you to the source of the problem, not only to fix it, but to make sure that it never happens again. Processes and standards are constantly changing and evolving based on the insight gained through discovering quality problems, performing root cause analysis on those problems, and implementing countermeasures to ensure that those quality problems do not happen again.



Many agile and lean methods recommend the use of information radiators to share information across the team, with management, and the customer. Information radiators often come in the form of Kanban systems, agile card walls as well as simple low fidelity dashboards and charts.



Organizations can scale by organizing teams into a value network. This value network typically employs peer-to-peer communication enabled by specific key members belonging to more than one team. The notion of teams actually being a set of overlapping concentric circles is a good metaphor. For the most part, teams within a value network should be cross-functional, having workers who possess diverse skill sets and perspectives.



Occasionally teams within a value network will be comprised of a single specialization, similar to the model found in the more traditional management methods. The specialist approach may be chosen when there is no stable demand for a specific skill set in any of the cross-functional teams. This approach is also effective when communication requirements between specialists of the same skill set tends to be higher than those across skill sets. Functions like enterprise architecture, security, and infrastructure provisioning are frequently organized according to specialized teams.

The current body of agile and lean thinking has found expression in many forms, including a variety of principles, methods, and specific practices. This thinking provides guidance, inspiration,

and advice for those wishing to operate successfully in a complex market taking advantage of self-organization, feedback and learning, frequent customer delivery, and excellence.

- Agile manifesto
- Principles of Lean software development
- Toyota principles
 - Scrum
 - Extreme Programming
 - Kanban
- Test-driven development
- User stories
- A3
- Behavior-driven development
- Retrospectives
- Daily standups
- Sprint planning

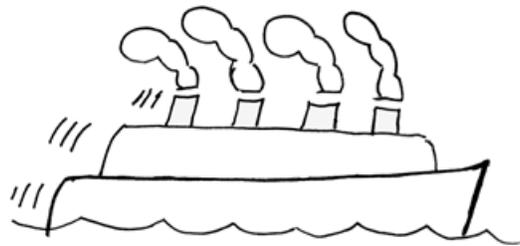
Agile Transformation Is Extremely Hard.

Executing on Lean and Agile Requires a Fundamental Shift In the Way Most Organizations Work

Organizations using traditional management methods rely on detailed planning, command-and-control, and a hierarchical structure. Work is processed by departments that employ specialists of a specific skill set. This work is typically passed across specialist departments in large batches.

By contrast, organizations using lean and agile methods place a much bigger emphasis on learning rather than planning, and on self-organizing teams who rely on a network structure to deliver business value. Work is consumed in short iterations, or just in time by teams. These teams are typically cross-functional, and contain the majority of skills required to satisfy a particular unit of customer demand.

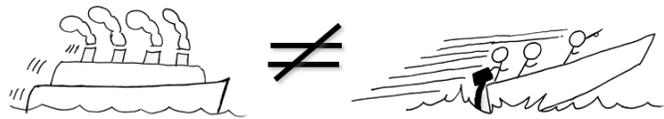
Organizations using traditional methods can be thought of as an industrial cargo boat or cruise ship. This design is incredibly efficient, and a lot of cargo or passengers can be carried across vast distances for a relatively low cost.



Organizations using agile and lean methods are a lot more like a speed boat. The cost per passenger seems to be a lot higher, but this is made up for in speed and maneuverability.



Metaphors aside, the real point here is that an organization using traditional methods does not look a lot like an organization using agile or lean methods. Just like a cruise liner and a speed boat may both be boats, they really have very little in common when it comes to operation, objectives, and capability. Agile and traditional organizations are really not the same thing. And the change required to transition from one to the other is substantial.



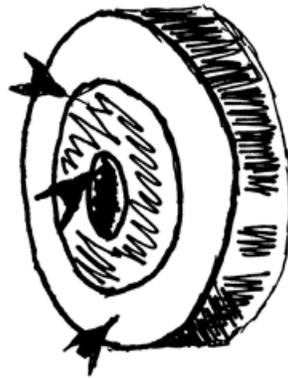
The Big C-Change traditionally used by Big-C consulting firms - Fundamentally Wrong for Agile Change

There is a huge industry focused on assisting organizations to make dramatic changes necessary to helping them survive. Most major consultancies have an army of practitioners and partners who have dedicated their careers to helping organizations rewrite the way they think and operate.

In the spirit of full disclosure, I have worked, and I continue to work in a “Big C Consulting” environment. During my experiences I have met more than a few very capable change consultants who have a lot of good advice to offer potential agile change agents. Several of these change consultants have provided me with valuable assistance in refining the Lean Change method.

That being said, it’s been my experience that these change consultants achieve success in spite of, not because of, the methods and tools that they tend to use. While specific consulting methods vary depending on the firm in question, our experience is that consulting change methods currently follow a typical execution pattern.

Consultants work with organizational executives, managers, and occasionally key team members to articulate a desired target organizational state based on a set of key business drivers.



Consultants then work with the client to examine the current state of the organization, compare it to the target, and come up with a set of gaps between the two.



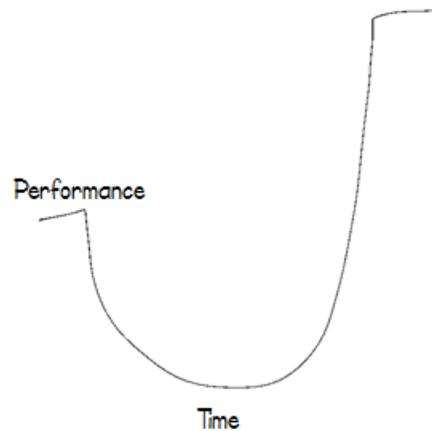
A roadmap is then developed based on prioritizing business drivers to come up with an implementation plan around how the organization will effectively transform to the desired state. This roadmap frequently comes in the form of a detailed plan that outlines key milestones, required activities, and effort required by both employees and external consultants.



This approach comes with a number of risks, especially when trying to affect dramatic change in an organization, such as transforming from traditional methods to one leveraging lean and agile thinking.

This change management approach typically results in the implementation of big C change. Dramatic changes are rolled out to the organization, sometimes on a department by department basis, resulting in wholesale shifts in job titles, processes, and technology. When any change is introduced into an organization, even a change that is good for that organization, a drop in performance will result. New methods need to be learned, new responsibilities take time to master, and new skills take time to acquire.

If the organization is able to successfully stick with the change, then performance will bottom out at the new, deteriorated level of performance. Eventually the desired changes will have the effect of improving performance, allowing the organization to reap the benefits of the target state.

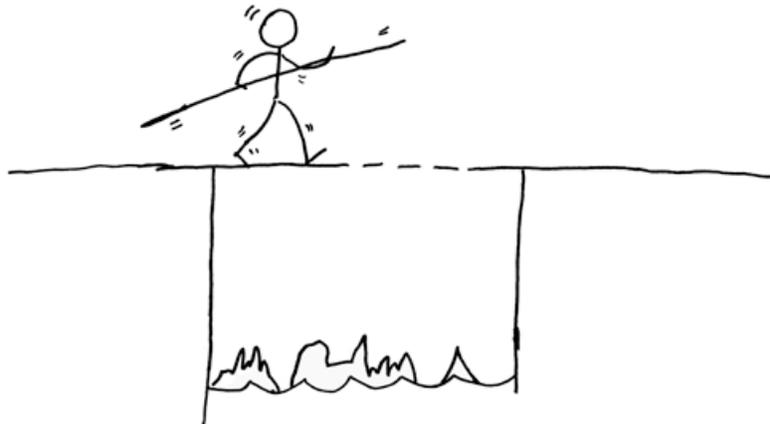


This however is an almost naive prediction of how big change actually occurs. In reality many big changes stall well before organizational benefits achieves the promises of the target state.

Organizations naturally resist change, and professionals within an organization will resent any force that asks them to change the way they are working. Fear is a major cause of change resistance; fear of losing one's job, fear of no longer being relevant, as well as fear of being asked to work in a way that one may not be used to.



Organizations are susceptible to abandoning the change project when organizational performance drops to its lowest. It's at this point that the change agent might find himself fired. Paradoxically, organizations that tend to operate at lower levels of maturity will drop more in performance when asked to make a large change, but will also have a lower tolerance for this drop, and be more likely to abandon the change as panic sets in.



Even if change agents and change champions can effectively keep the organization on track, and get the organization to the target state, the promise of improved performance may not materialize. The reality is that the suggested change may be wrong for the specific context of the organization in question.



As stated before, the agile and lean body of knowledge contains a diverse set of methods, practices, and principles, not all of these are equally suitable to every organization's context.

- Agile manifesto
- Principles of Lean software development
- Toyota principles
 - Scrum
 - Extreme Programming
 - Kanban
 - Test-driven development
 - User stories
 - A3
 - Behavior-driven development
 - Retrospectives
 - Daily standups
 - Sprint planning

Every organization has a different set of business drivers, level of maturity, willingness to change, and an endless host of other factors that will ensure that no two agile transformations are alike.

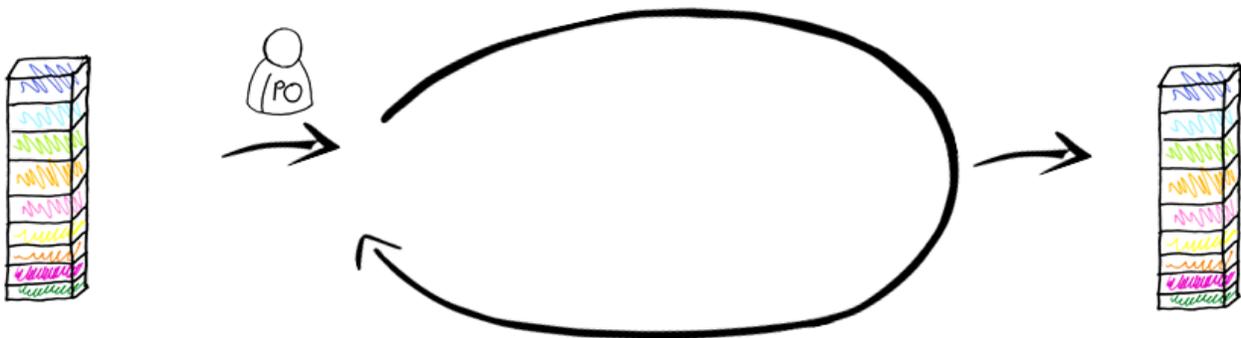
Etc.
 Maturity Business
 Culture maturity Etc.
 Technology Size
 Etc. Market
 risk

If we remember our original discussion around plan-driven methods, a key point is that they leave us open to the risk of building the wrong thing. This is true of products, software, as well as a change management target state. In an environment of high variation and complexity, faithfully completing a plan that was built in the beginning of an engagement leaves us open to creating something that has no value.

Using Scrum to Inspect and Adapt - A Better approach to Agile Change, but Often Results in Too Large a Change That Does Not Fit All Contexts

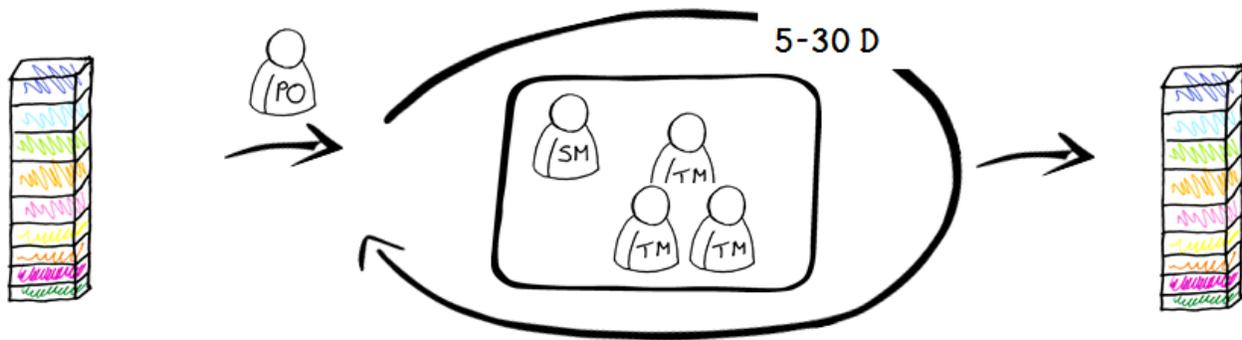
Many organizations attempting to increase their agility and transform their technology delivery capability to take advantage of lean and agile thinking have elected to use Scrum. Scrum is a deliberately minimalistic agile method that is meant to help teams get started with agile, as well as provide a framework to help technology knowledge workers adapt and improve.

Customer demand is placed into a product backlog, the content of this backlog is primarily the responsibility of a Product Owner, who coordinates with other business stakeholders to prioritize, refine, and otherwise groom this backlog. The Product Owner is considered to be the “one throat to choke” when it comes to making sure that the right product features are being delivered according to the desire of business. Work is typically delivered to the end-user in iterations of between 5 and 30 days. This iteration is a heartbeat for delivery, and is known as a sprint.



Software, (and other business valued work) is completed by a cross-functional, self-organizing, and

empowered team that is typically co-located. In order to encourage this cross-functional nature, Scrum specifies a very limited set of roles i.e.: a team member, a Scrum Master, and optionally a coach. Team members may have specific specializations, but are not constrained to only doing any one role. (e.g. a developer may test if it is helpful), Scrum Masters are expected to be servant leaders who help the team by removing impediments, protecting them from organizational politics, and providing other advice and guidance.

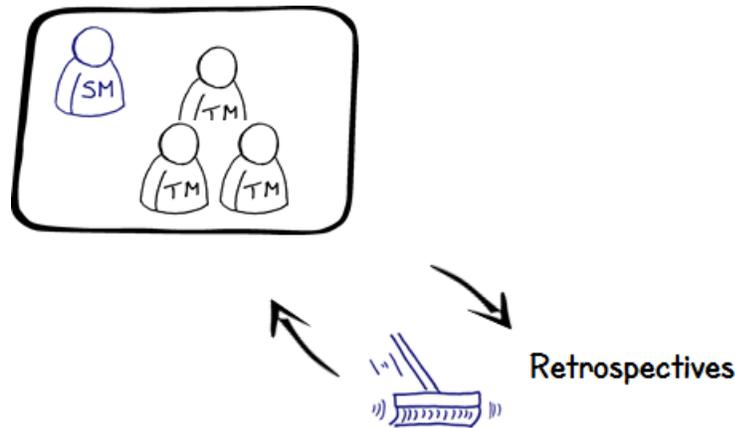


A particularly important part of Scrum is that teams only commit to delivering what they feel they can possibly accomplish as part of a specific sprint. Commitment is done at the team, rather than individual level.

Sprint cycles are typically accompanied by a very lightweight set of artifacts and meetings. Scrum provides advice on how to conduct Sprint planning sessions, daily Scrums, and product demos, as well as guidance in how to track velocity using burn down charts.

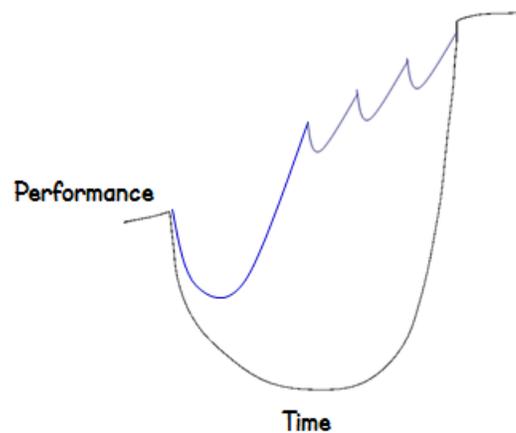


Scrum helps teams and the organizations they work for to become more agile in a number of ways. First of all, sprint cycles are designed to maximize customer feedback allowing teams to course correct and maximize customer value. Sprint cycles are also accompanied with what is known as a retrospective, a regularly recurring meeting where team members focus on continuous improvement and addressing existing issues and problems.



The iterative, customer facing nature of Scrum is also designed to make organizational dysfunction visible, encouraging the organization to make more systemic changes that are required to ensure that individual teams can be successful. This focus on organizational impediments provides the data required for change agents to transform to be more agile.

Using Scrum to help organizations transform share some, but not all of the risks found within a big C change approach. Because Scrum is a minimalistic framework, the performance impact of change from initial adoption is not as adverse as found in larger structurally-based transformations. Subsequent waves of change can also be smaller, as the pace of change can be graduated based on the impediments found by different teams.



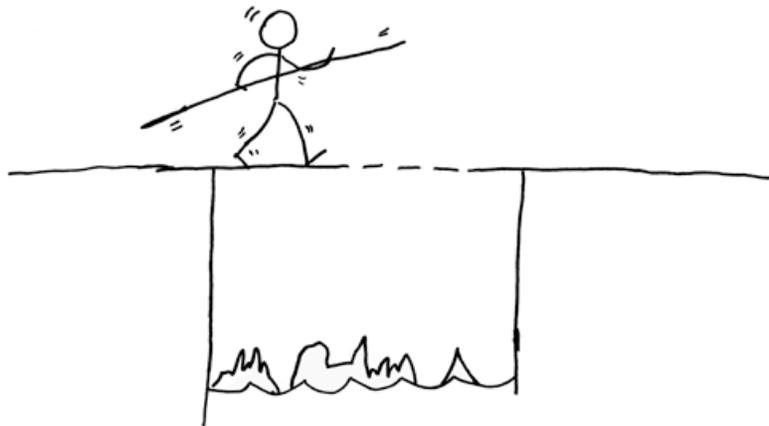
That being said, different organizations have faced a number of challenges while trying to adopt Scrum, and a significant portion of these organizations have “failed” to successfully adopt the method.

While Scrum is a lightweight method, it asks executives, managers, and staff to work in a fundamentally different paradigm. The whole notion of cross-functional teams, self-organization, and delivering in small iterations can cause major conflict with organizations that have grown up using traditional methods and traditional thinking. Many people resist the alien terms that come

with Scrum, and are threatened by this unfamiliar way of working.



What many folks adopting Scrum also fail to realize is that implementing it must be quickly followed by implementing other practices from the lean and agile world. The hyper agile model of Scrum breaks down if organizations are not willing to invest in significant, further changes.



Scrum also faces challenges due to the fact that a pure agile model is not always the right recipe for all organizations. Not all domains lend themselves to permanent, generalist, self-organizing team members. Not all work neatly fits into the notion of short, time boxed intervals. Some organizations are simply not ready for the large and sudden shift required to make Scrum successful.



Kanban - a Viral, Evolutionary Approach to Change Management, Many Contexts Require Significant Investment and Expertise to Be Truly Successful

Kanban has been quoted as being a “viral, evolutionary” change management approach designed to help organizations gradually become more agile over time.

Kanban is designed to provide an even more gradual way for technology knowledge workers to improve than the minimalist Scrum process framework. Knowledge workers start by mapping out their existing delivery process and visualizing this process using a Kanban system, typically manifested as a Kanban card wall.

Next	Analysis	Development	Test

The Kanban system is then directly connected to business stakeholders responsible for prioritizing and validating business valued work. Each customer is given a specific queue and allowed to place work on that queue based on the amount of capacity dedicated to that customer.



Next	Analysis	Development	Test

Delivery agility is introduced in the system by putting explicit limits on how much work can be consumed at a time. Very low work in process limits will force teams to adopt a much more agile and lean working style, collaborating frequently, resolving impediments quickly, and keeping quality high through best in class techniques. High work in process limits will allow teams to operate in a way that is much more similar to traditional and legacy methods. There will be a much higher tolerance for multitasking and working in silos. As a result, there will be less pressure to collaborate

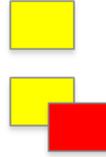
and continuously improve.



Next	Analysis	Development	Test
(3)	(2)	(3)	(4)

Current work in process is then visualized on the Kanban system based on where it is currently in the process. Work in process is expressed as work tickets, which may represent enhancements, user stories, or other business valued work.



Next	Analysis	Development	Test
(3)	(2)	(3)	(4)
			

Simple visual indicators such as color are used to annotate work so that technology knowledge workers and their customers can immediately recognize important attributes such as risk, priority and capabilities required to complete a specific unit of work. This allows knowledge workers to assign unique processes to different work types. For example, emergencies could be completed as soon as they are received regardless of what other work is in progress.

Hot colored tickets can also be used to annotate business valued work with associated blockers, and other impediments. This visualization connects both status and risk of work with knowledge workers and their managers on an emotional level, allowing them to see how much work is in process, as well as the health of that work. This encourages people to make better decisions and improve maturity.

Knowledge workers will work through the process using a pull-based approach. Using this method, work is only moved into a particular state when doing so will not violate the inventory limits for that particular state. This ensures quick turnaround of delivery work, or what is known as delivery flow. Delivery flow creates a high feedback system similar to that found in Scrum or other approaches that rely on short iterations.

Kanban helps organizations adopt a lean mindset, based on collaboration, continuous improvement and self organization. Kanban places a heavy emphasis on improvement through experimentation. Kanban borrows lean analytical techniques such as the use of statistical process control and cumulative flow diagrams to measure performance, flow, and lead time.

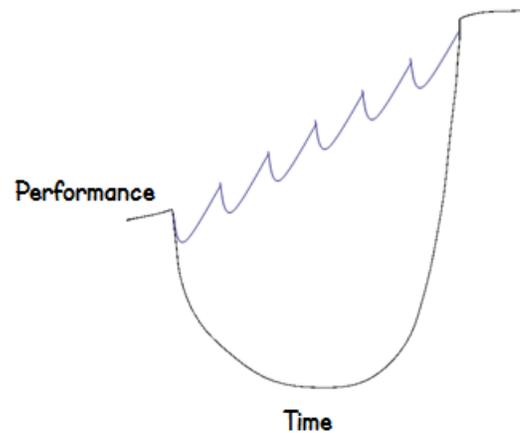


The combination of visualization, empirical-based improvement, and focus on flow allows knowledge workers to design customized process solutions that match their own context. Different Kanban systems are allowed to evolve at their own pace within the same organization based on the unique needs of the workers and customers involved in delivering business value. This process and method diversity is enabled through a common process improvement method.



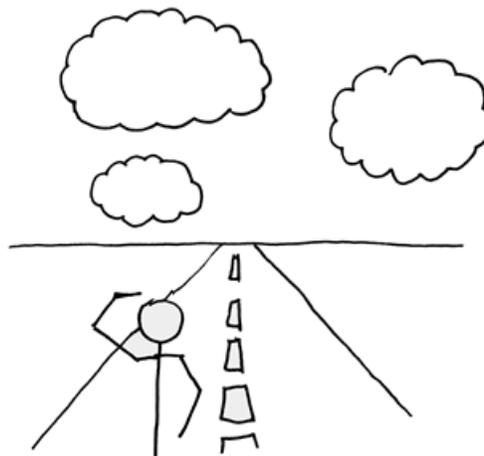
Kanban has been labeled as a meta-method, not a software delivery method in its own right. Technology knowledge workers and their managers are encouraged to design the right method for their own needs, using an incremental evolutionary approach. Kanban has also been described as a viral improvement method. Kanban can be adopted in a small part of the organization, which eventually encourages other workers to connect to that system, and adopt their own Kanban implementation.

When Kanban works as designed, it offers the least disruptive path to organizational change. Because getting started does not require anybody to change roles, change job titles, or change the way they are organized, more conservative and traditional organizations can get started with less impact than adopting Scrum.



Kanban contains many design elements such as process policies, work in process limits, and work types that are customizable to the context of the business value being delivered, this means that Kanban can be matched to the unique constraints of the organization.

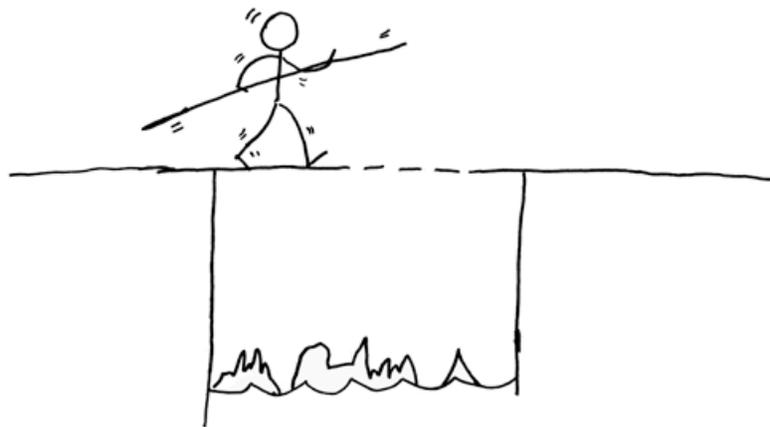
By design, a viral and evolutionary method can take a long time to provide desired performance improvement results. When asking an expert how long a Kanban improvement method will take to achieve high maturity, the answer is often as long as it needs to. While this approach may arguably be the one that makes the most sense, it is often an answer that most organizations desiring some type of transformation are not willing to or able to accept.



Kanban is also not designed to provide specific process answers for particular problems. Knowledge workers are expected to use Kanban to make problems visible, and engage in continuous improvement to make the system work better. Our experience is that this can actually cause churn and confusion in organizations with little practical agile experience. The wealth of options and the customizable nature of Kanban can actually make change harder for some organizations.



Finally, Kanban adoption can stutter just like Scrum due to the lack of existing maturity within the organization. Kanban systems that do not process fine grained units of work, similar to user stories found in Scrum and other agile methods, may exhibit a lack of feedback, slow performance, and poor poor stability. This can interfere with continuous improvement. As a result the initial jump to successfully adopting Kanban can be larger than it is first thought to be.



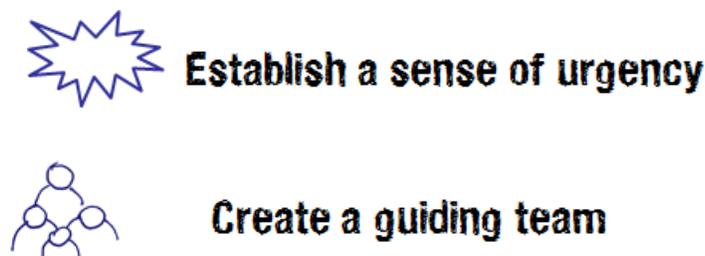
The Eight Steps of Change

John Kotter, in his text “The Heart of Change” describes an eight step change lifecycle, showing a number of case studies illustrating change agents working with an organization to enable significant change following these steps. The eight step change lifecycle has been used for over a decade to help guide large-scale managed change initiatives. Kotter argues that successful change requires emotional resonance, people being asked to change have to feel the need to change in their gut. Kotter recommends that organizational change start with building urgency, and institutionalizing quick wins using a coalition of eager change adopters.

This coalition of the willing is guided by a clear vision, good communication, and the right level of empowerment to create short-term victories. Momentum for the change is continued until the change is successfully incorporated into the new corporate culture.



Step number one is establishing a sense of urgency. Insightfully, John believes that most people are at least subconsciously aware of what is wrong with an organization, thus starting with a target options or vision is the wrong way to go. Instead, successful change agents should focus on establishing a sense of urgency within the organization. The idea here is to try to emotionally connect the rationale behind the change with the people who are being asked to change. Don't be over analytical at this stage, but rather focus on tactics that help people in the organization feel the need to change in their gut.

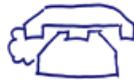


The guiding team then works on a change vision, a "true North" that can help guide the activities of this change.



Develop a change vision

Subsequently, the guiding team works on communicating as necessary to establish consensus and understanding across all change stakeholders of the the change.



Communicate for buy in

Executives, sponsors and stakeholders are all responsible for empowering action so that the guiding



Empower actions

team can make the changes necessary to realize the vision.

It is critical that the guiding team structure their change management effort so that they receive wins in the short-term, and not structure their change management activities so that benefits are backloaded.



Create short-term wins

The team, and its sponsors and stakeholders have to be careful to approach change in a sustainable way, and not giving up partway through



Don't let up

After the organization receives tangible benefits, effort switches to making sure that change sticks, and becomes a cemented part of the organizational culture.



Make change stick