

# BIMODAL IT

## Mode 2 Evolution Roadmap



# Work In Progress

This presentation was inspired and influenced by the works of many people, and I cannot possibly list them all. It has been my sincere aim to respect all copyrights and reference the authors as appropriate. If however, you feel I have not succeeded in some aspects of my intent, please contact me at my email: [Janusz.Stankiewicz@gmail.com](mailto:Janusz.Stankiewicz@gmail.com), to help me correct my errors. Thank you.

# Gartner's Quotes...



Enterprises are living in a digital Wild West, where there is tremendous opportunity, significant risk and lots of uncertainty. In this environment, technology is one of the great disruptors, and one of the great weapons... Because IT's conventional methods are ill-suited to respond to uncertainty, decrease cycle times and explore this new territory, a new imperative comes into play.



The ability of executive teams to forecast the future of their competitive environments, markets and value chains is diminishing, almost on a quarterly basis — just as these same teams are under pressure to find ways to mitigate stakeholder risks and exploit business opportunities. Meanwhile, disruptive digital trends crowd the horizon. The consumerization of IT, the Nexus of Forces (social, mobile, cloud, information and analytics) and the Internet of Things have triggered explosive digital demand.



The value created by technology has migrated out of the back office, into the front office, and straight into the hands of customers, citizens and employees. Internal pressures are also forcing the hand of the IT organization — the growth in shadow IT being a manifestation of users' desire to control their technological destiny, of their growing confidence in their ability to do so and of their dissatisfaction with the IT organization's current methods. Despite the many advantages of conventional IT approaches, they are ill-suited to respond to all this change.

# Gartner's Quotes...



The recent exploitation of information technology is not a curious trend in isolated pockets of a handful of industries. The whole world is becoming digital, and sooner or later, every commercial, industrial and service sector, public and private, will be “digitally remastered.” Digital is disruptive for most industries, overturning rather than extending accepted modes of operation. The digital divide between what the IT organization can provide and what the enterprise wants and needs is widening.



IT can reach a Project Bimodal state without much change required from the business. This requires four capabilities:

1. Creating a high-performance team that accepts uncertainty as part of the game
2. Creating agile, lean or other iterative development capabilities
3. Developing governance that is empirical, continuous and process based
4. Constructing a separate organizational structure for Mode 2.

We Are Living In Times Of Explosive Digital Demand Growth... Therefore New Paradigm For Software Delivery Methods Is Required

# Bimodal IT Defined

-  Bimodal IT refers to having two modes of IT each designed to develop and deliver information- and technology-intensive services in its own way.
-  Mode 1 is traditional, emphasizing safety and accuracy.
-  Mode 2 is non-sequential, emphasizing agility and speed.
-  Each mode has all the people, resources, partners, structure, culture, methodologies, governance, metrics, and attitudes toward value and risk that its operation requires.
-  New investment are deployed through one of the two modes, depending on the balance of needs. When the balance changes, existing investments and operations move between the two models.

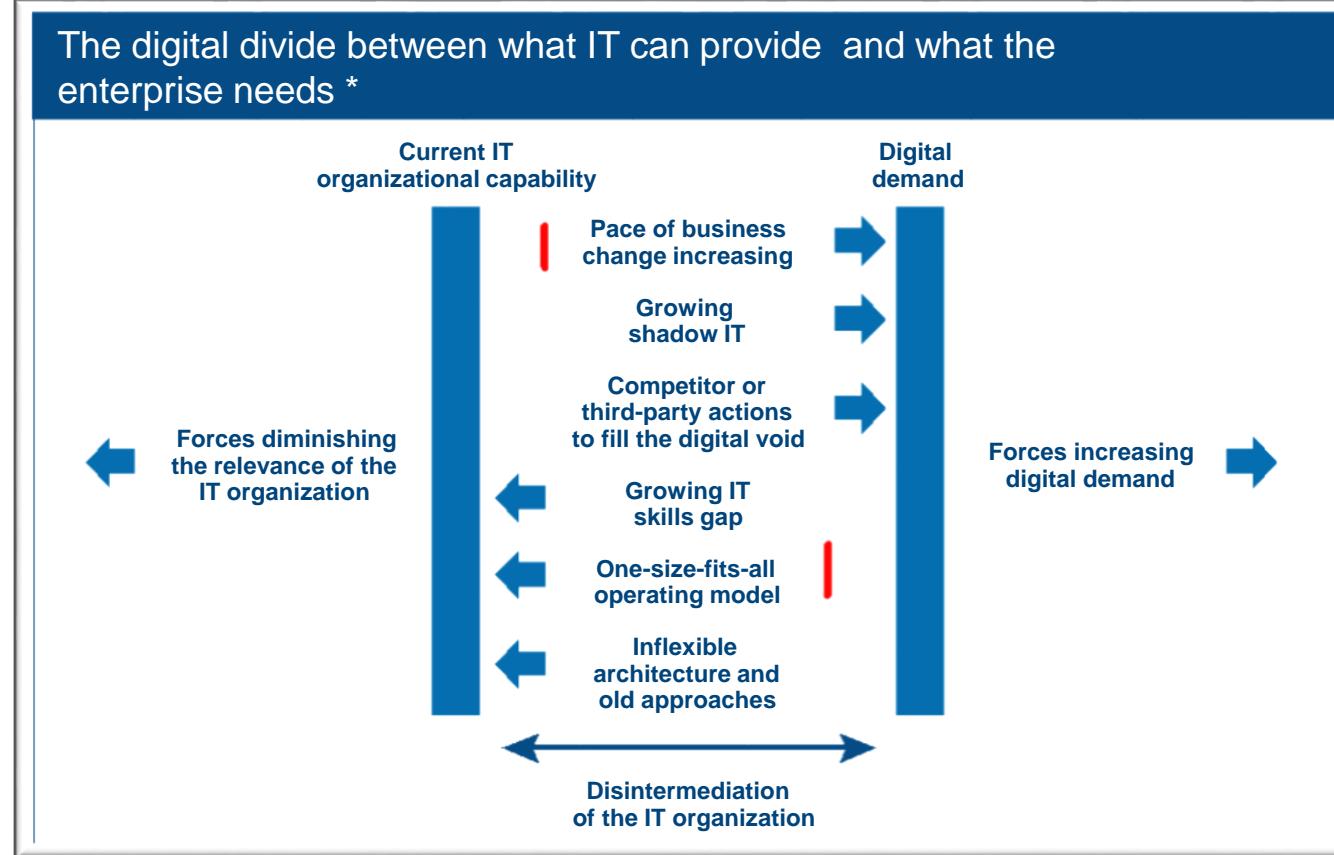
# Mode 1 vs. Mode 2

Bimodal IT: The two modes are deeply different — tweaking Mode 1 cannot create Mode 2 \*

	Mode 1	Mode 2
Think marathon runner	<b>Goal</b>	Reliability
	<b>Value</b>	Price for performance
	<b>Approach</b>	Waterfall, V-model, “high-ceremony IID”*
	<b>Governance</b>	Plan-driven, approval-based
	<b>Sourcing</b>	Enterprise suppliers, long-term deals
	<b>Talent</b>	Good for conventional processes and projects
	<b>Culture</b>	IT-centric, removed from customer
	<b>Cycle times</b>	Long (months)
Think sprinter		Agility
		Revenue, brand, customer experience
		Agile, Kanban, “low-ceremony IID”*
		Empirical, continuous, process-based
		Small, new vendors; short-term deals
		Good for new and uncertain projects
		Business-centric, close to customer
		Short (days, weeks)

\*Iterative and incremental development (IID) comes in two forms. The original IID methods can be considered condensed waterfall, with eight-week time boxes typical. This “high-ceremony IID” is suited to Mode 1 — e.g., the original Rational Unified Process (RUP). The second form of IID — “IID lite” or “low-ceremony IID” — entails much more concurrent work, reduced document needs and less process ceremony. It is better suited to Mode 2 approaches — e.g., OpenUP and Microsoft Solution Framework (MSF) for Agile Software Development.

# Why Do We Need To Start Now?



The Gap Is Widening... Bimodal IT Is A Powerful Capability For Addressing These Issues

\* Gartner

# What - Vision

"We need to figure out a way to deliver software so fast that our Customers don't have time to change their minds"

Mary Poppendieck



Applicable to Systems-Of-Engagement (Front-End Layer), partly to Systems-Of-Differentiation (Integration Layer).  
Not Applicable to Systems-Of-Records (Back-End Layer), which is Mode 1 Domain

## The 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 process 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.

## 12 Principles of Agile Software

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 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. Agile processes promote sustainable development. The sponsors, developers,
7. and users should be able to maintain a constant pace indefinitely.
8. Working software is the primary measure of progress.
9. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
10. Continuous attention to technical excellence and good design enhances agility.
11. Simplicity – the art of maximizing the amount of work not done – is essential.
12. The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

## Lean Principles



## THINK LESS IS MORE – ELIMINATE WASTE

**The 7 Wastes of Manufacturing**

- Inventory
- Extra Processing
- Overproduction
- Transportation
- Waiting
- Motion
- Defects

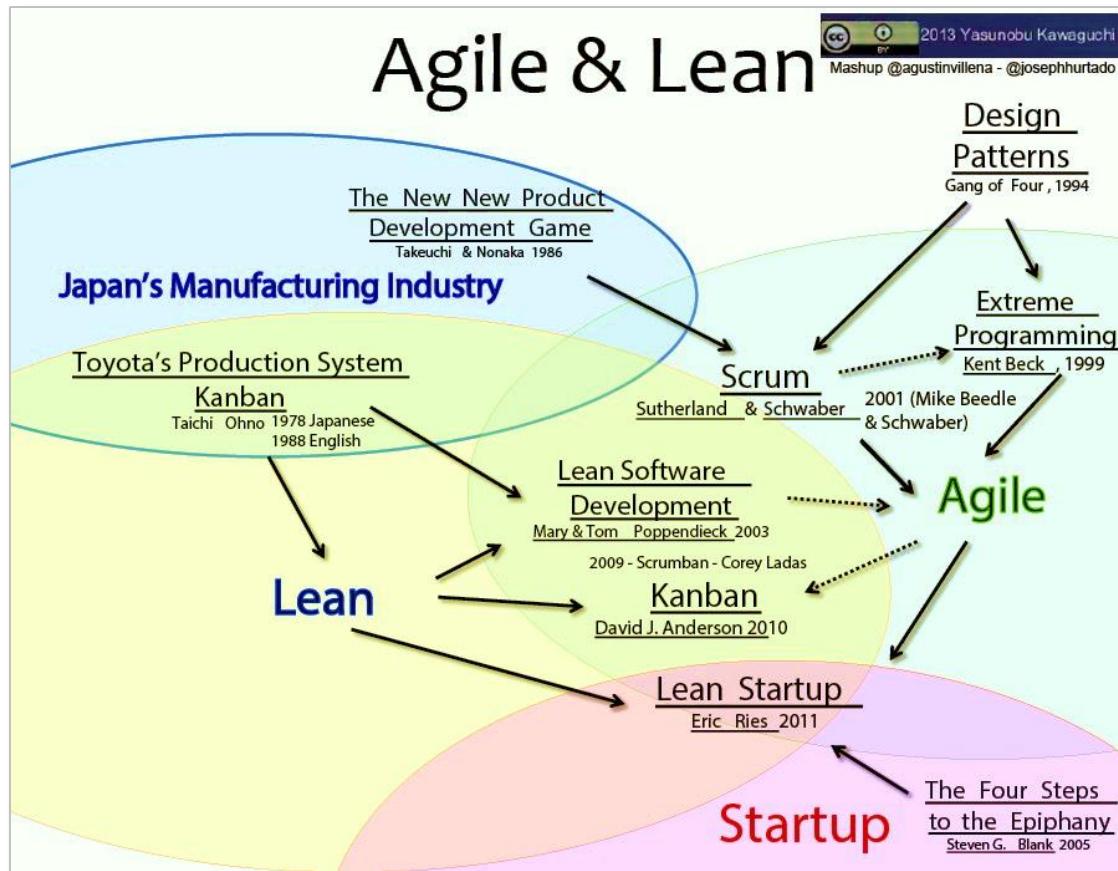
**The 7 Wastes of Software Development**

- Partially Done Work
- Extra processes
- Extra features
- Task Switching
- Waiting
- Motion
- Defects



## 7 Principles and 22 Practices of Lean Software Development

1. Eliminate Waste
  - Seeing Waste, Value Stream Mapping
2. Amplify Learning
  - Feedback, Iterations, Synchronization, Set-Based Development
3. Decide as Late as Possible
  - Options Thinking, The Last Responsible Moment, Making Decisions
4. Deliver as Fast as Possible
  - Pull Systems, Queuing Theory, Cost of Delay
5. Empower the Team
  - Self-Determination, Motivation, Leadership, Expertise
6. Build Integrity In
  - Perceived Integrity, Conceptual Integrity, Refactoring, Testing
7. See the Whole
  - Measurements, Contracts



## Agile\* or Lean\*\*?

Both are sharing the core underlying values of speed of customer value delivery, collaboration, and continuous improvements.

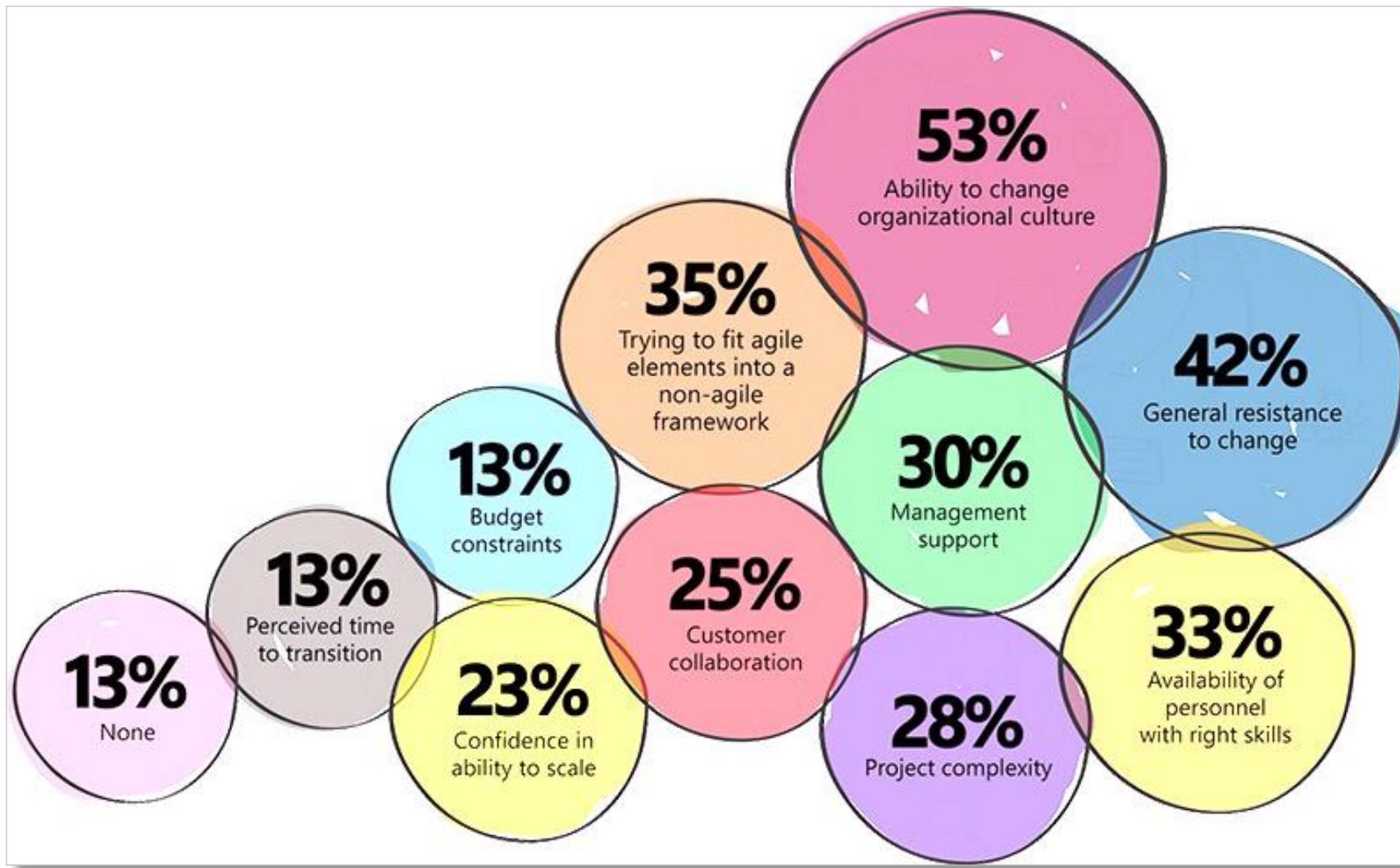
Both are leveraging similar or even the same principles, practices, tools, and techniques.

... But are using slightly different delivery mechanisms, and to begin with feel at home in different operating cultures.

### Methods and/or Process Frameworks:

- \* Adaptive Software Development, Agile Modeling, Agile Unified Process, Crystal Clear, Disciplined Agile Delivery, Dynamic Systems Development Method, Extreme Programming, Feature-Driven Development, Scrum
- \*\* Lean Software Development, Kanban, Scrum Ban

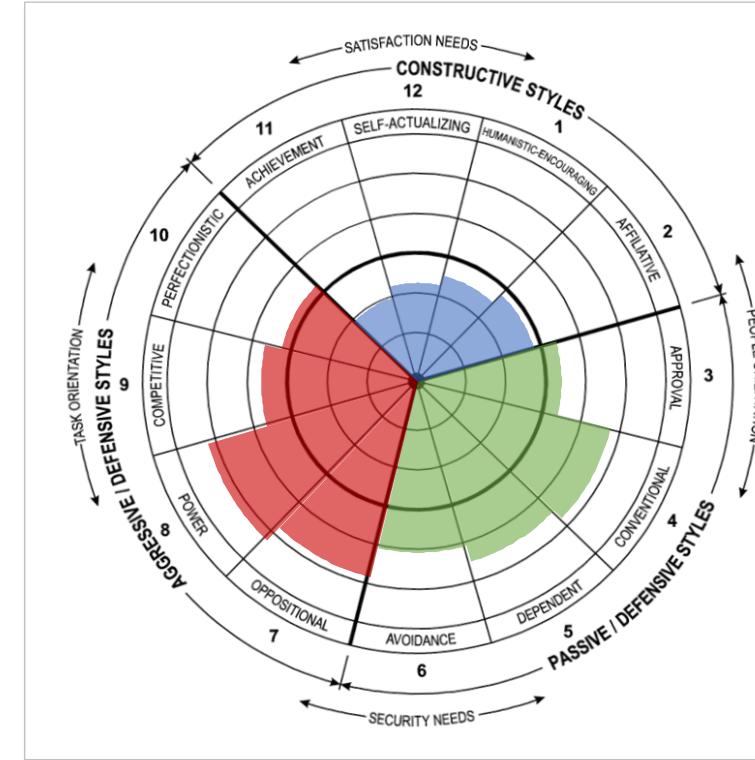
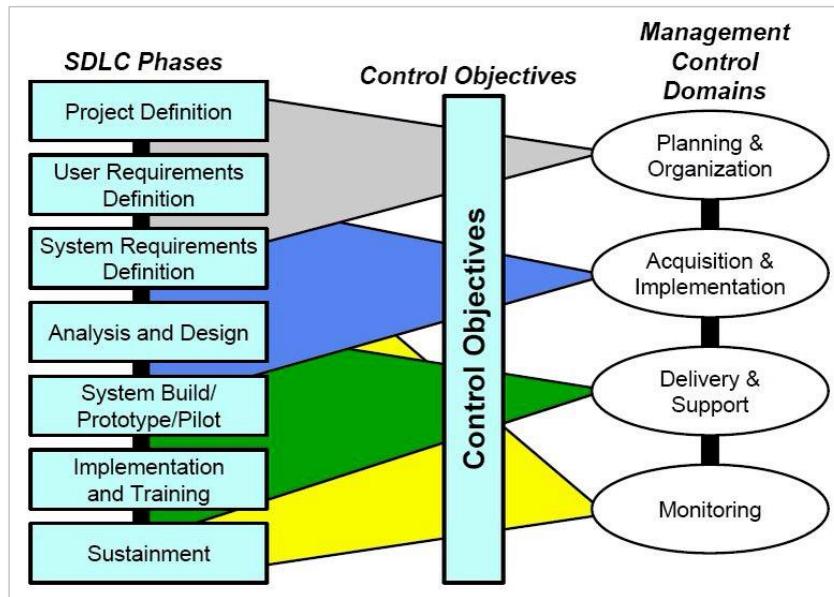
# Key Barriers For Agile Adoption



\* 8<sup>th</sup> Annual State Of Agile Survey by VERSIONONE, 2014

# **Where Are We Today?**

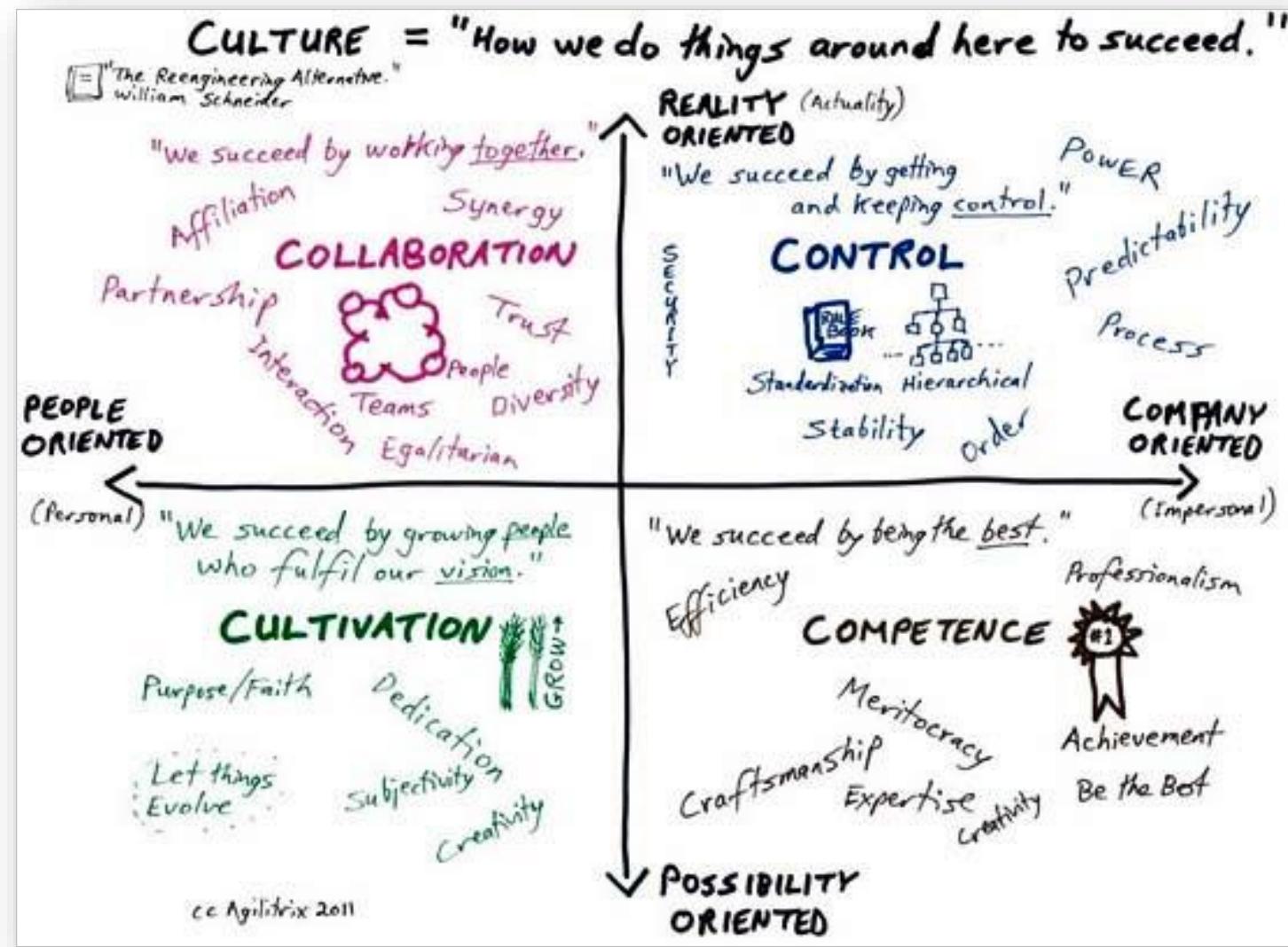
# Typical Mindset and Operating Culture In Command and Control Driven Organizations



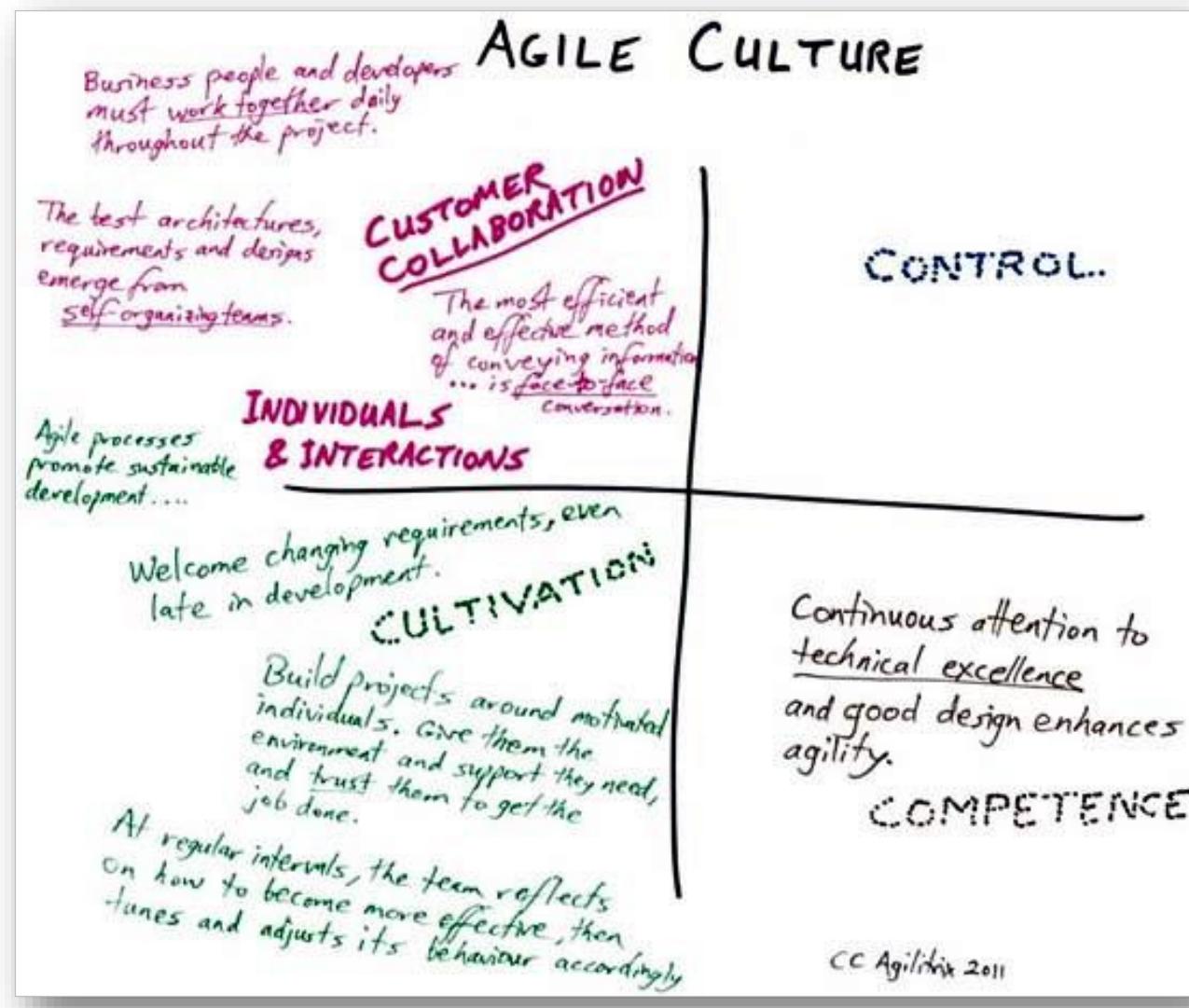
Strong Waterfall Mindset Deeply Rooted in Current Operating Culture

Operating Culture:  
High – Power, Oppositional and Conventional  
Low – Achievement and Self-Actualizing  
styles of behavior

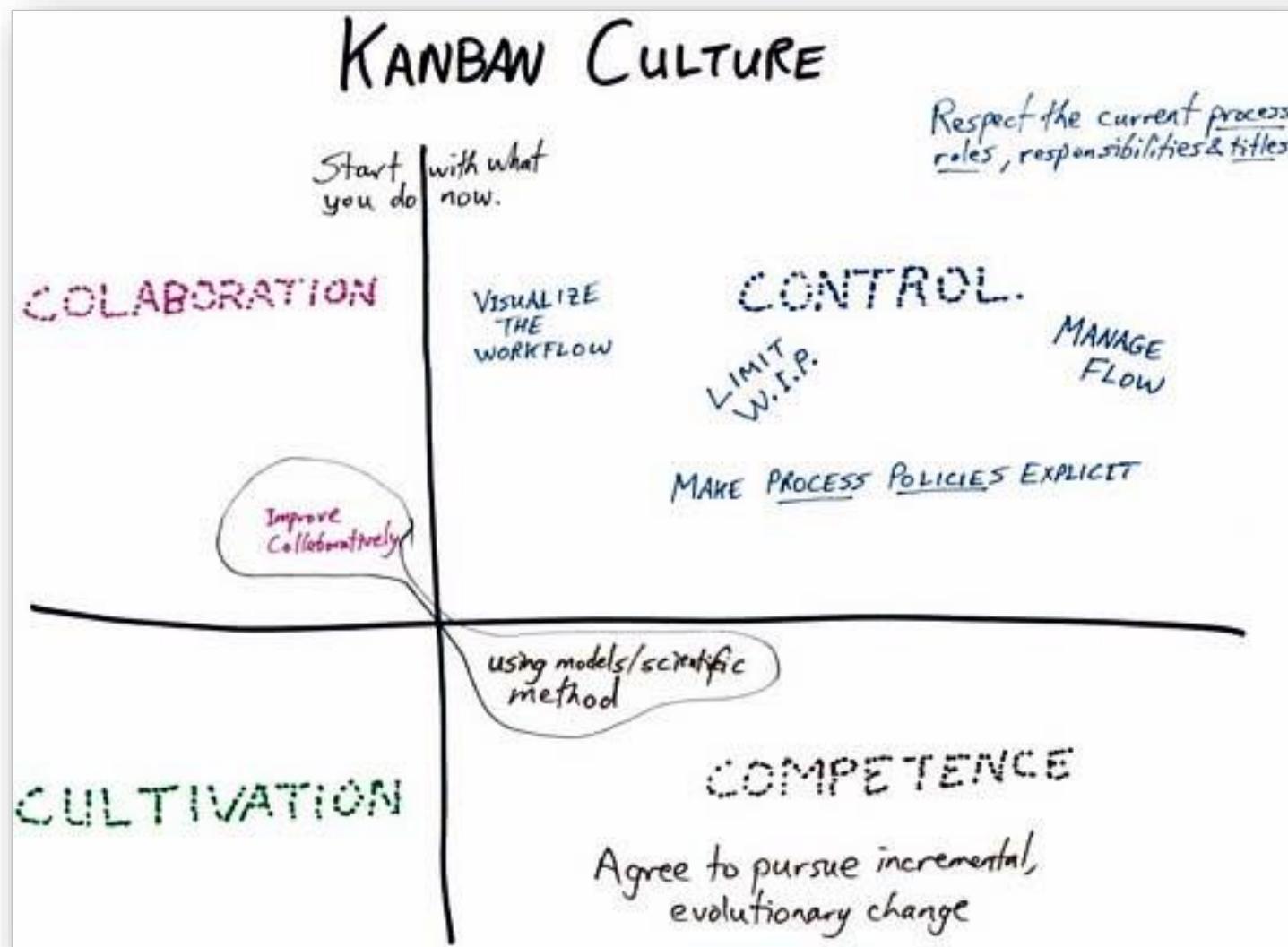
# Schneider Culture Model



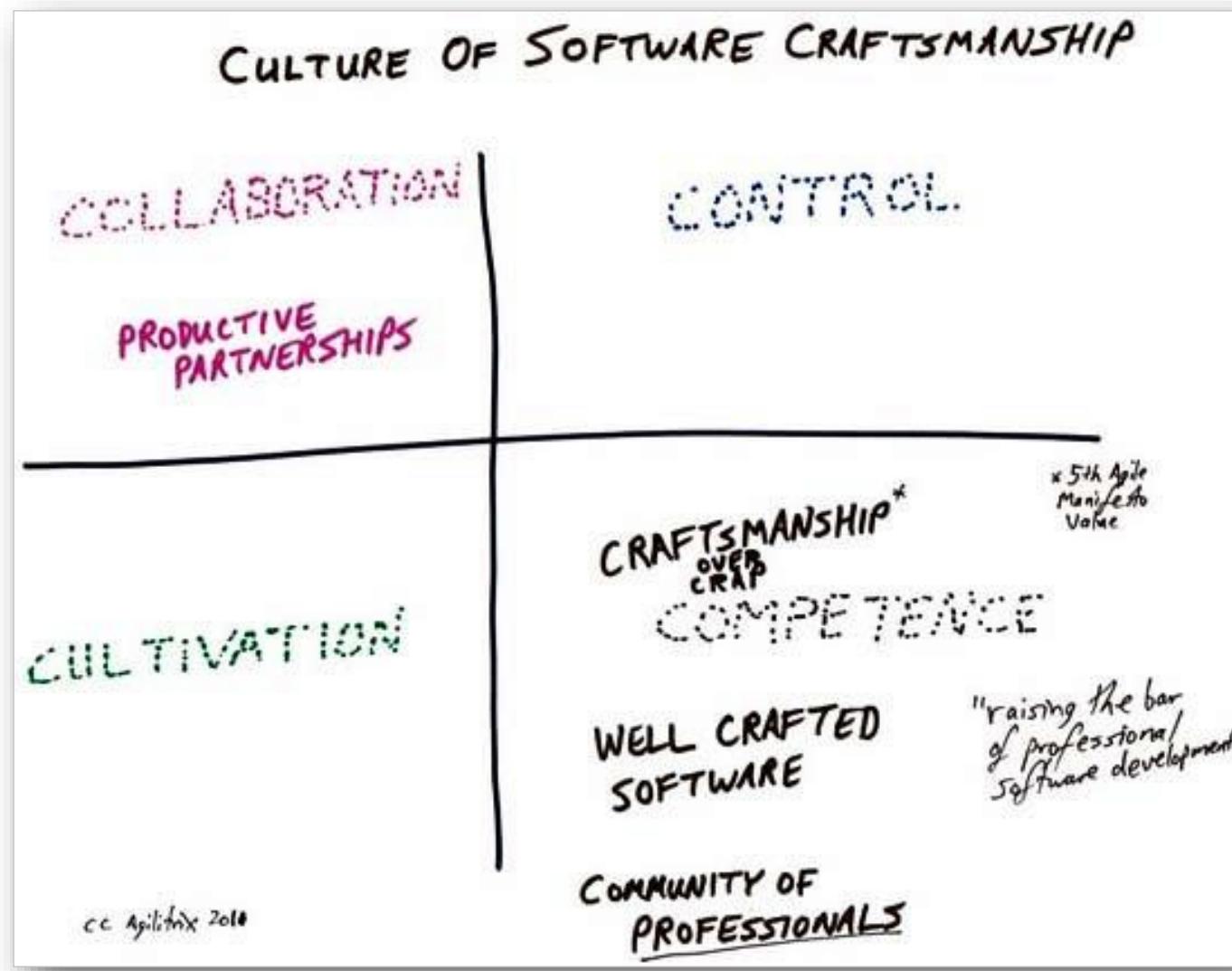
# Cultural Fit Analysis 1/5



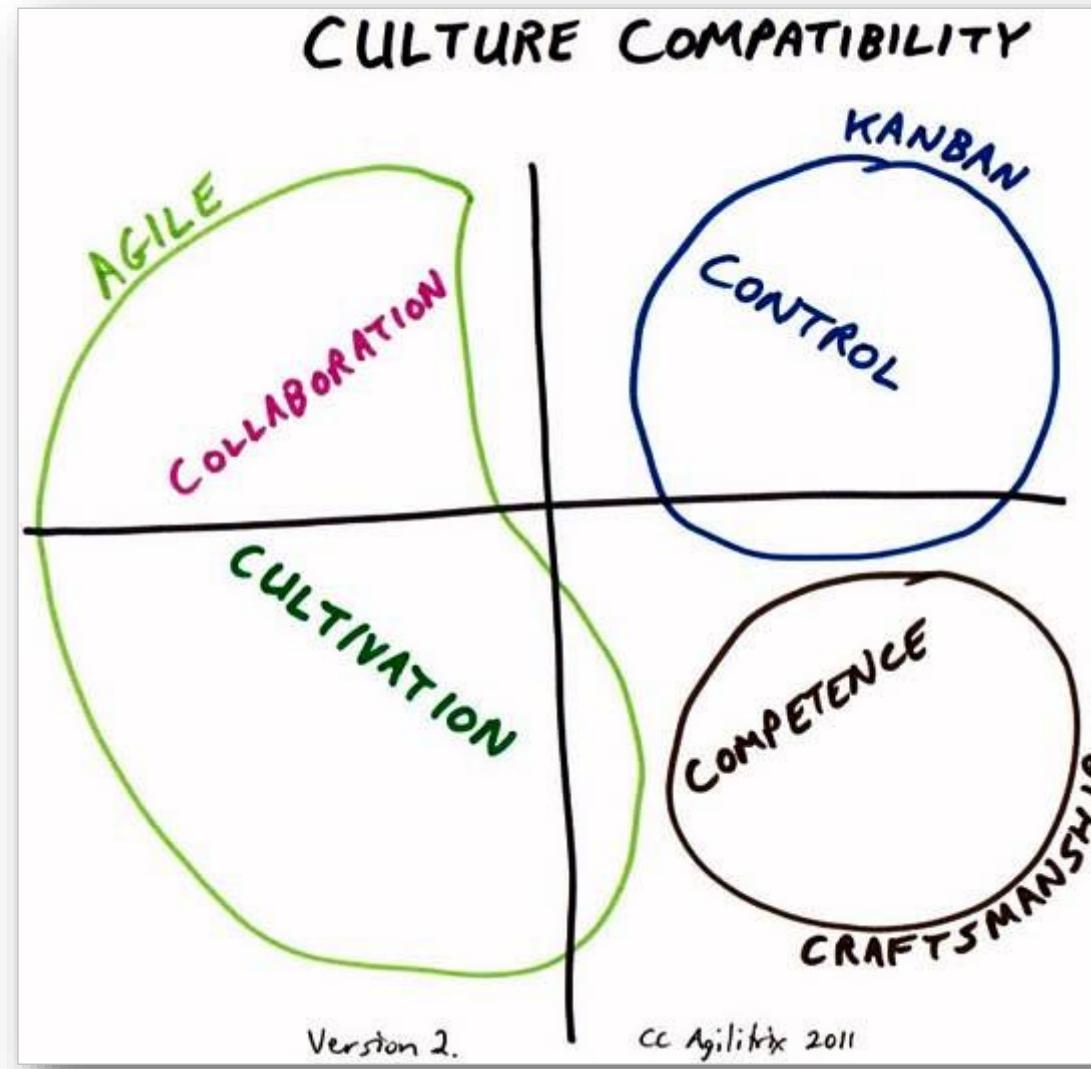
# Cultural Fit Analysis 2/5



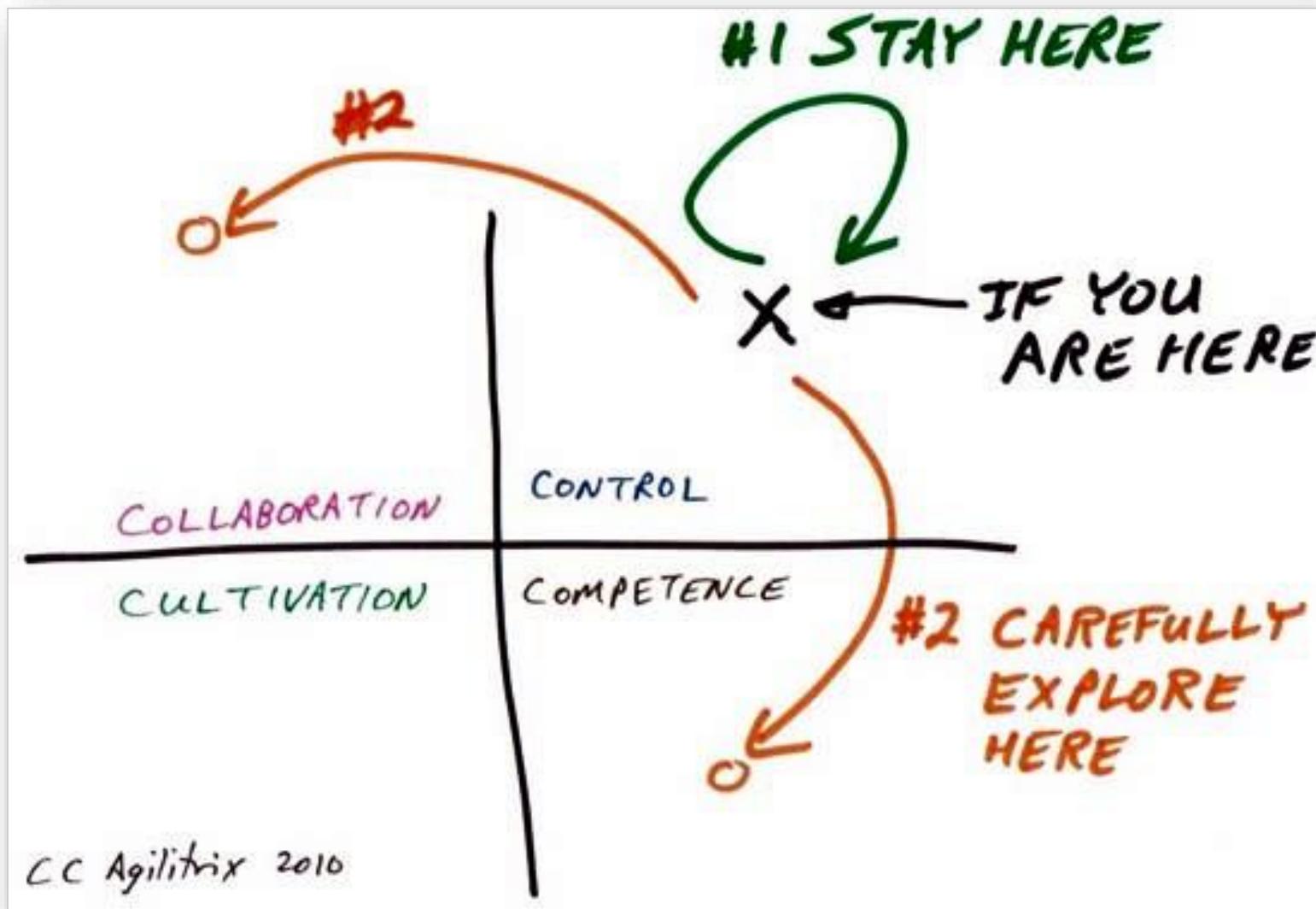
# Cultural Fit Analysis 3/5



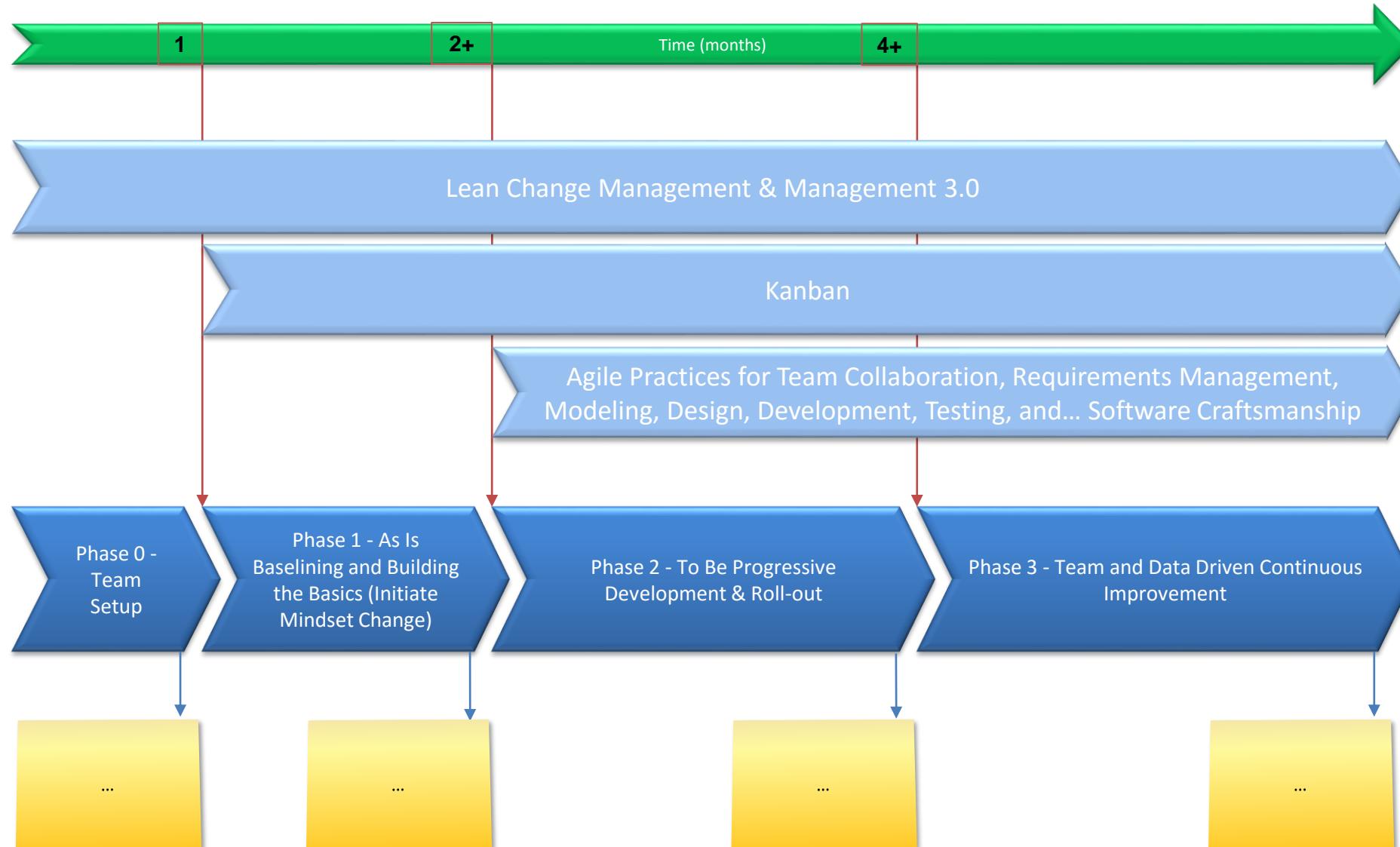
# Cultural Fit Analysis 4/5



# Cultural Fit Analysis 5/5

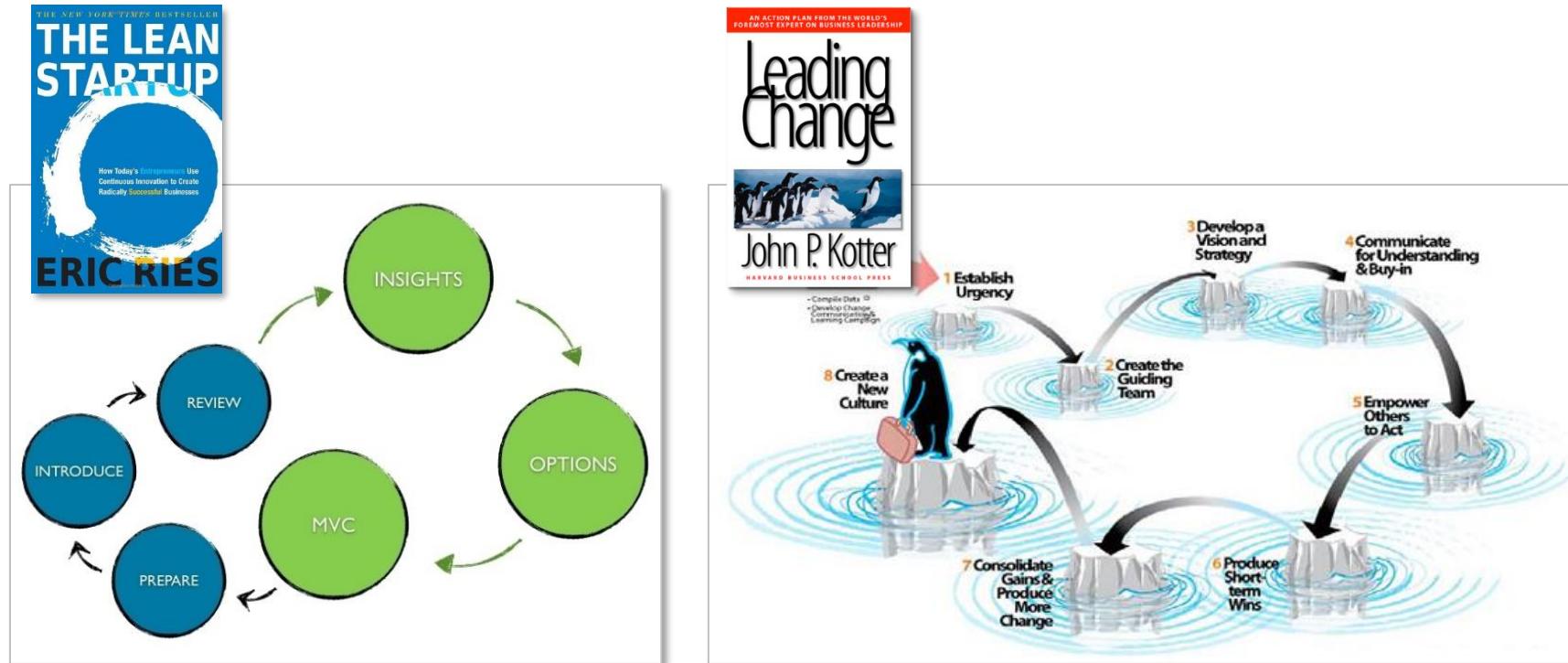


# Mode 2 Evolution Roadmap



# Lean Change Management

Lean change management is a system of innovative methods for effecting change in an organization's management, which was formulated by Jeff Anderson and Alexis Hui. This system brings together concepts such as Agile, Lean Startup, and Kotter's 8 Step Model to create a feedback driven approach.



# Lean Change Key Themes



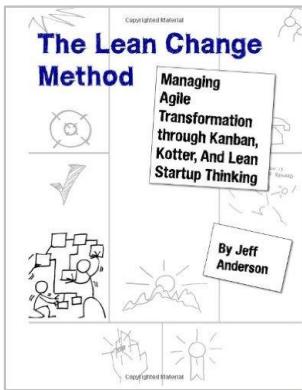
**Negotiated Change** - Negotiated Change approach demands that recipients of any change are co-authors and co-implementers of all aspects of the change that they are part of. Designated change agents, change stakeholders, and change recipients act as change *co-creators*, ensuring that suggested changes get the buy-in necessary to ensure that they are successful



**Validated Learning** - Lean Change advocates that any change plan and change target state model should be described as a set of assumptions, and change agents and other change stakeholders are responsible for validating these assumptions with explicit hypotheses.



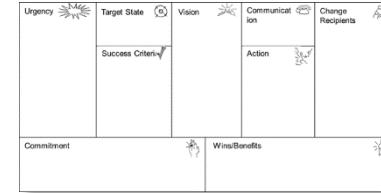
**Lean Change Requires Improvements Kanban** - For the lean change method to work it is required that team members adopt their own internal agile improvement method to help them identify impediments and other improvement opportunities. Most Lean Change implementations have elected to use Kanban as the improvement method of choice



# Lean Change Components



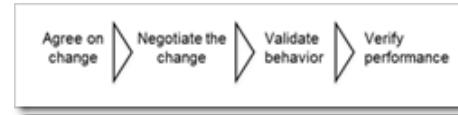
**Change Canvas** - The canvas is an informal "plan on a page", which cover most parts of Kotter's 8 Step Model



**Minimum Viable Changes** - Smallest possible change that will enable learning whether a particular change will provide sustainable improvement.

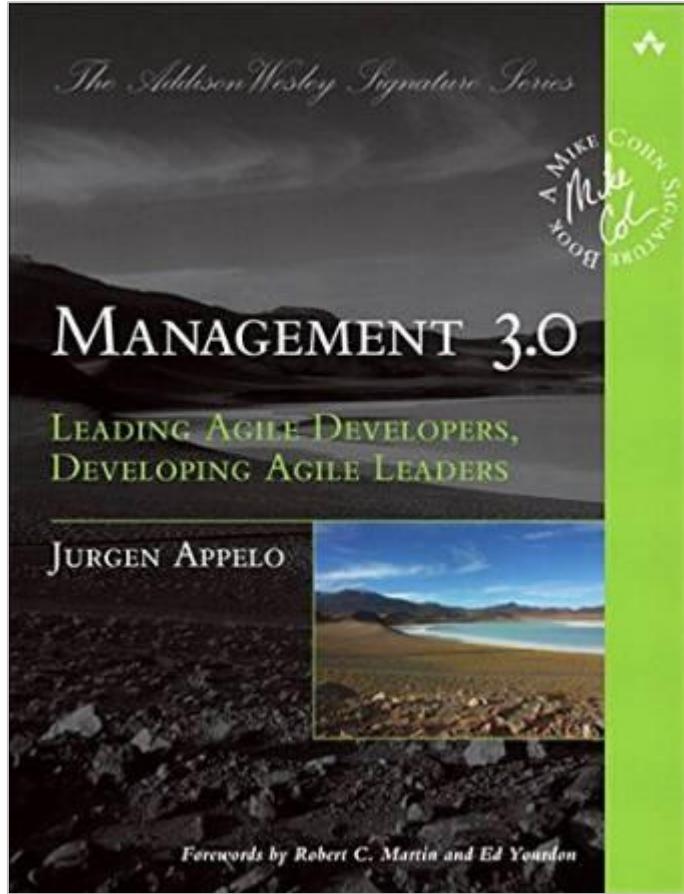


**Validated Change Lifecycle** - Minimum Viable Changes are introduced to the organization through a Validated Change Lifecycle. This lifecycle has been defined to maximize ability to accelerate negotiation and learning necessary to creating a successful change.



**Capability and Performance Metrics** - Lean Change also provides a number of ways to measure the impact of specific changes. The first perspective is the ability of change recipients to adopt, and ultimately excel at new agile and lean methods and techniques. The second perspective is the impact of these techniques on actual delivery performance and value.

# Management 3.0



“Management 3.0 brings together the best thinking in the field of complex adaptive system, Agile management, and Lean product delivery to suggest a pragmatic framework for effective management in the 21<sup>st</sup> century. To be successful in the face of rapidly changing market conditions, we must create organizations that enable our people to adapt, with a minimal amount of oversight and direction. Management 3.0 gives us a roadmap for leading teams in the face of profound uncertainty. Jurgen [Appelo] has made a significant contribution to the field of Agile management and leadership.”

Mike Cottmeyer, Agile Coach, LeadingAgile

# Management 3.0



## Management 1.0 = Hierarchies

Some people call it scientific management, whereas others call it command-and-control. But basic idea is the same: An organization is designed and managed in a top-down fashion, and power is in the hands of few.



## Management 2.0 = Fads

Some people realized that Management 1.0 doesn't work well out-of-the-box, so they created numerous add-on models and services with a semi-scientific status, like the Balanced Scorecard, Six Sigma, Theory of Constraints, and Total Quality Management. Being add-ons to Management 1.0, these models assume that organizations are managed from the top, and they help those at the top to better "design" their organizations. Sometimes it works; sometimes it doesn't.



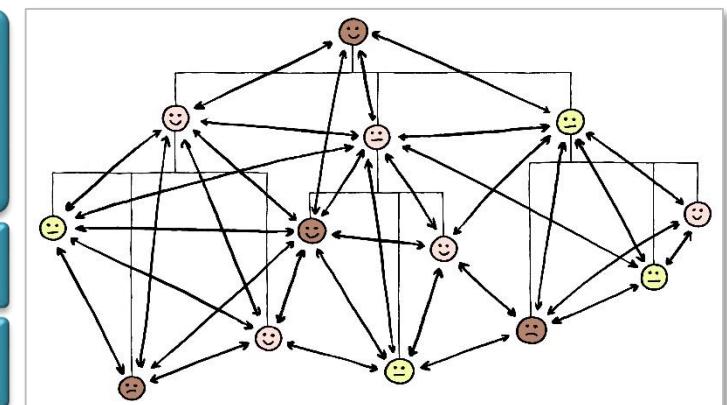
## Management 3.0 = Complexity

People may draw their organizations as hierarchies, but that doesn't change that they are actually networks. Second, social complexity shows us that management is primarily about people and their relationships. It makes us realize that we should see our organizations as living systems, not as machines.

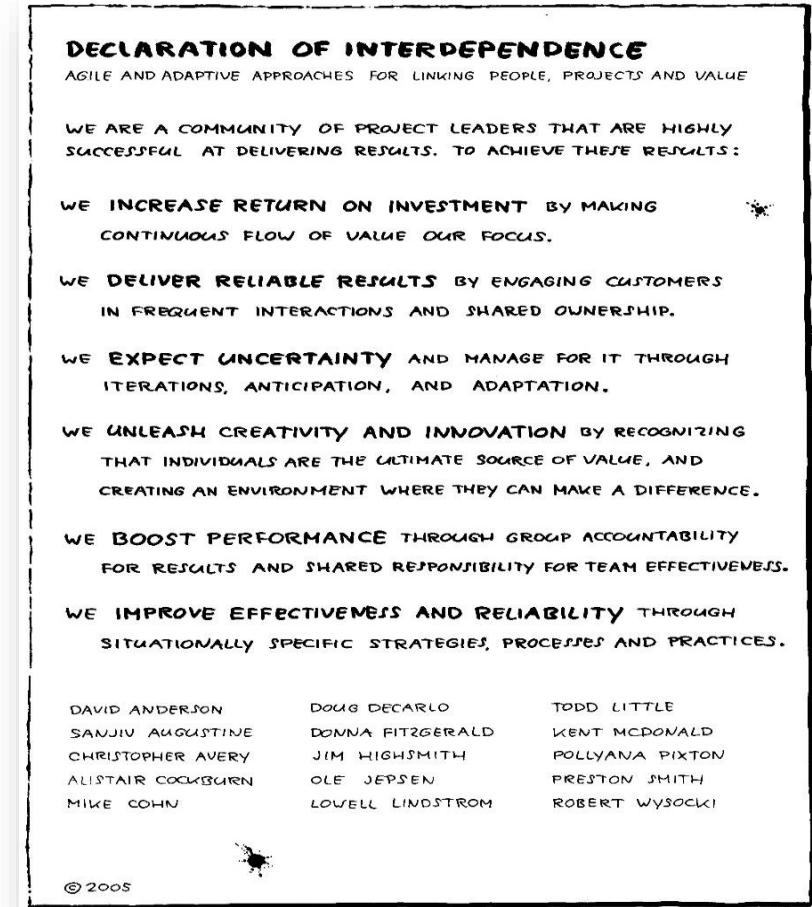
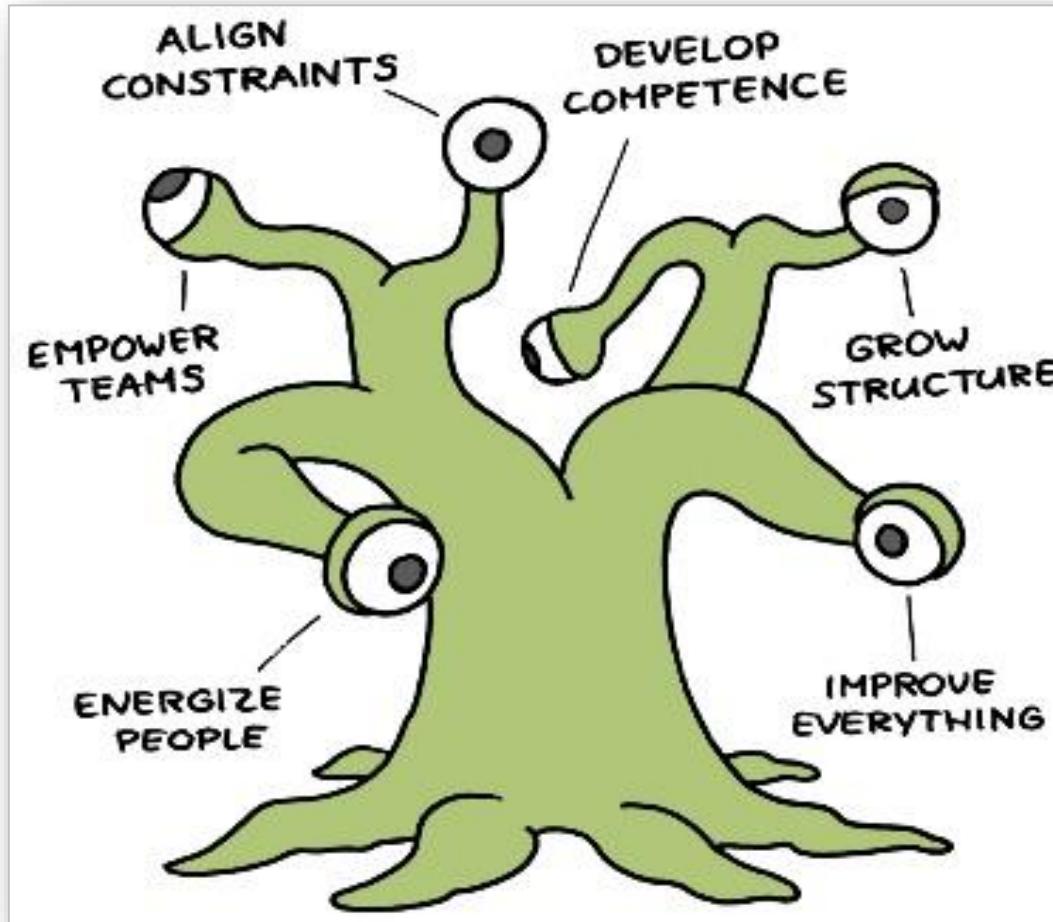
A team is a ***complex adaptive system (CAS)***, because it consists of ***parts*** (people) that form a ***system*** (team), which shows ***complex*** behavior while it keeps ***adapting*** to a changing environment

A software team is a ***self-organizing system***. Support it, don't obstruct it

Agile managers work the ***system around*** the team, not the people ***in*** the team



# Martie, Jurgen Appelo's Management 3.0 Model & Declaration Of Interdependence



The management hierarchy is a basic necessity (but nothing to brag about) and the bulk of the work is done in a social network of peers: leaders and followers. Communication flows through the network. Authorization flows through the hierarchy

# Best Quotes From Management 3.0

The job of management is not to select best ideas; it is to create a system that allows for the best ideas to emerge.

An organization passes the T-shirt Test when employees proudly wear a t-shirt with the company logo.

There is no shortage of ideas. There is a shortage of commitment to making them happen.

Creative people will game the system. The trick is to exploit this creativity to make the system more resilient.

If your best experiences in life are all vacations, then maybe you shouldn't return to work tomorrow.

The world has never been changed by those who just accept bad service or bad products.

Your company does not become Agile by adding "agility" to the required competences on people's annual performance reviews.

You cannot simply change an organization's culture. What you can change are your own behaviors and your influence.

Assume all your email is red by the NSA, leaked to the press, analyzed by your enemies, and forwarded to your mother-in-law.

Complaining that colleagues won't adopt your idea is like an entrepreneur complaining that money won't flow in his direction.

You won't have a first mover advantage for long if somebody else has the fast learner advantage.

Never forget that better principles, not better practices, are what organizations really need.

You get the best out of your employees when you treat them as entrepreneurs.

We should celebrate learning, not successes or failures.

Before you try to measure someone else's performance, please explain how you measure your own.

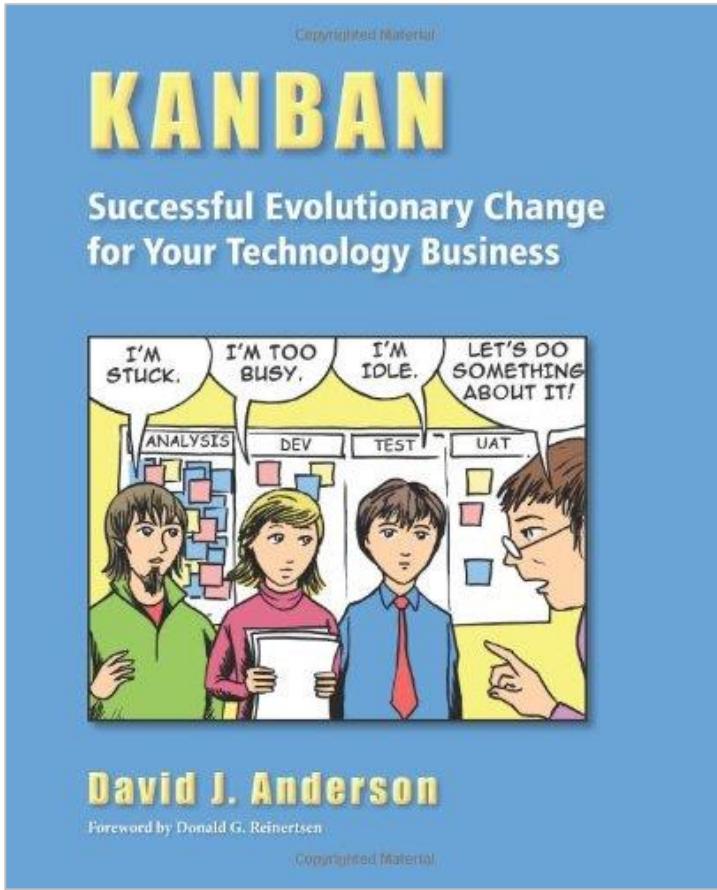
Personal coaching is not the manager's primary task. Manage the system, not the people.

Organization grow in the direction of the questions that people ask each other.

Delegation is not a binary thing. There are shades of grey between a dictator and an anarchist.

Treating employees like adult human beings might be common sense, but it is not common practice.

# Kanban Method



The name 'Kanban' originates from Japanese [看板], and translates roughly as "signboard" or "billboard". It was formulated by David J. Anderson as an approach to incremental, evolutionary process and systems change for organizations. It uses a work-in-progress limited pull system as the core mechanism to expose system operation (or process) problems and stimulate collaboration to continuously improve the system. Visualization is an important aspect of Kanban as it allows to understand the work and the workflow.

# Kanban 9 Values

- |                  |                 |
|------------------|-----------------|
| 1 Transparency   | 6 Leadership    |
| 2 Balance        | 7 Understanding |
| 3 Collaboration  | 8 Agreement     |
| 4 Customer Focus | 9 Respect       |
| 5 Flow           |                 |

# Kanban 4 Foundational Principles..

- 1 Start with what you do now
- 2 Agree to pursue evolutionary change
- 3 Initially, respect current processes, roles, responsibilities, and job titles
- 4 Encourage acts of leadership at every level in your organization from individual contributor to senior management

# Kanban 6 Core Practices

- 1 Visualize
- 2 Limit Work-in-Progress (WIP)
- 3 Manage Flow
- 4 Make policies explicit
- 5 Implement feedback loops
- 6 Improve collaboratively, evolve experimentally (using models and the scientific method)

# Mapping: Values vs. Foundational Principles

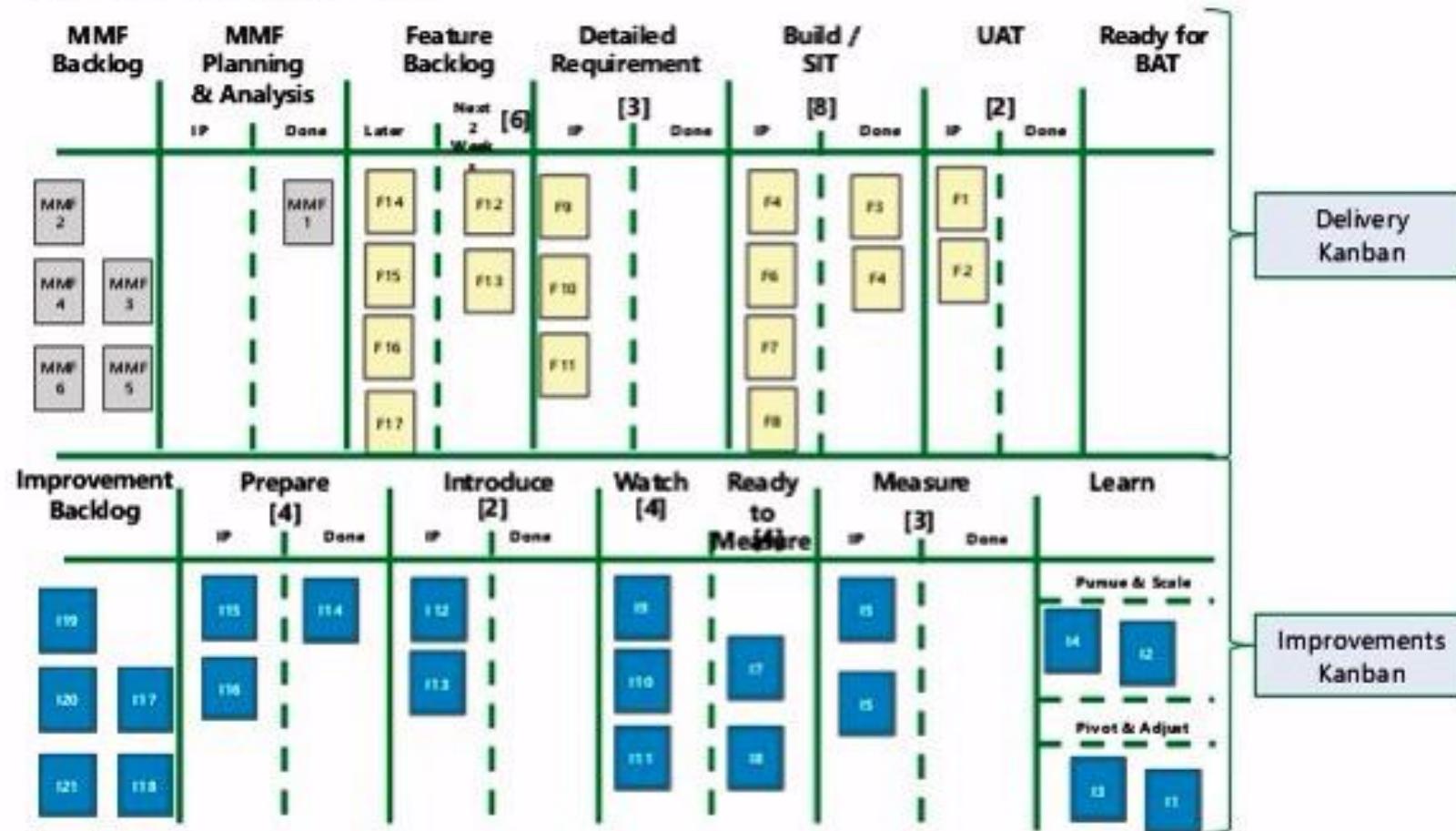
- ➊ Understanding: Start with what you do now
- ➋ Agreement: Agree to pursue evolutionary change
- ➌ Respect: Initially, respect current processes, roles, responsibilities, and job titles
- ➍ Leadership: Encourage acts of leadership at every level in your organization from individual contributor to senior management

# Mapping: Values vs. Core Practices

- Transparency: Visualize
- Balance: Limit Work-in-Progress (WIP)
- Customer Focus, Flow: Manage Flow
- Transparency: Make policies explicit
- Transparency: Implement feedback loops
- Collaboration: Improve collaboratively, evolve experimentally (using models and the scientific method)

# Kanban Board Sample

Improvements can be tracked and managed on a Kanban Board adjacent to the Team's Delivery Kanban



# Software Craftsmanship

**Software craftsmanship** is an approach to software development that emphasizes the coding skills of the software developers themselves. It is a response by software developers to the perceived ills of the mainstream software industry

## Software Craftsmanship Manifesto

As Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software, but also **well-crafted software**

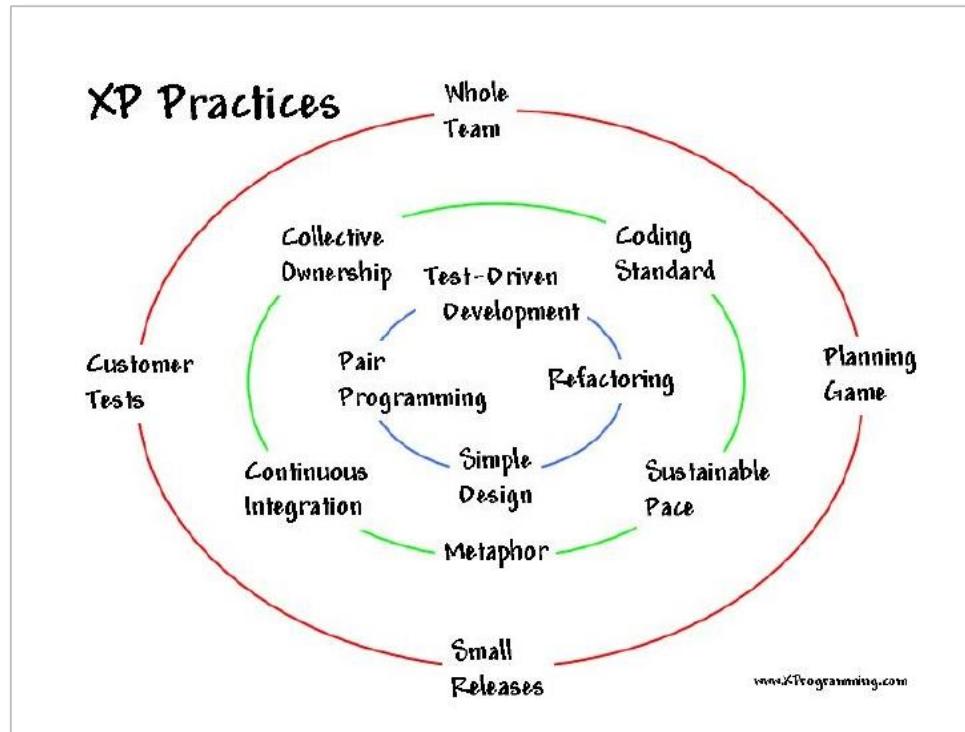
Not only responding to change, but also **steadily adding value**

Not only individuals and interactions, but also **a community of professionals**

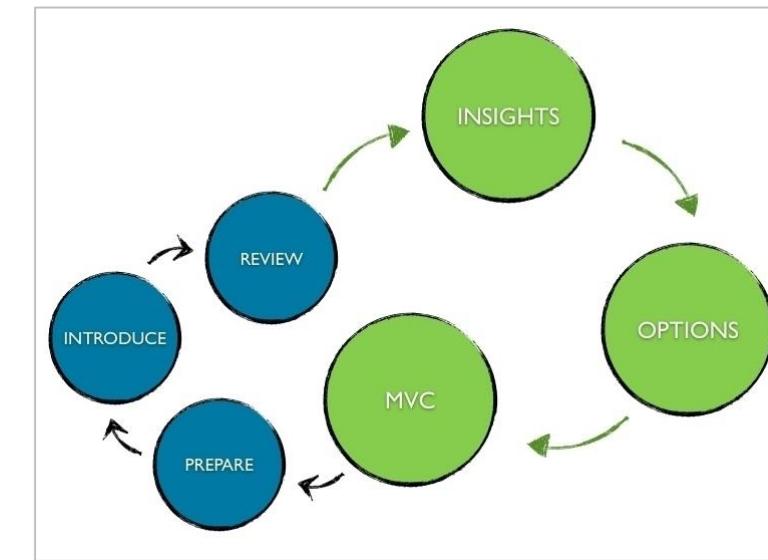
Not only customer collaboration, but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

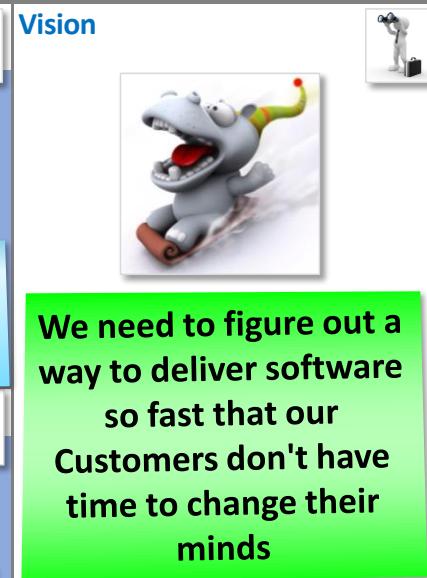
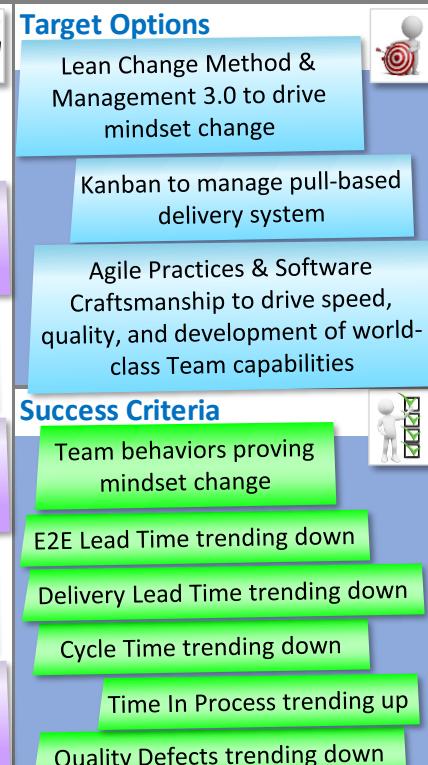
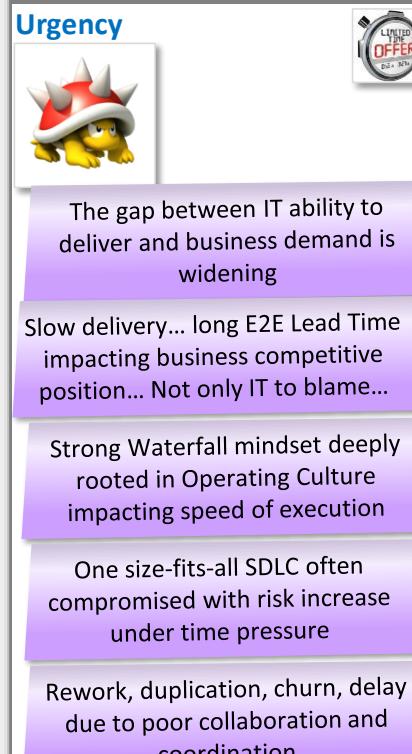
# Sample Practices... Initial Insights....



There Are More Out-There...But  
Remember Intention To Follow  
Lean Change Management Cycle  
Philosophy

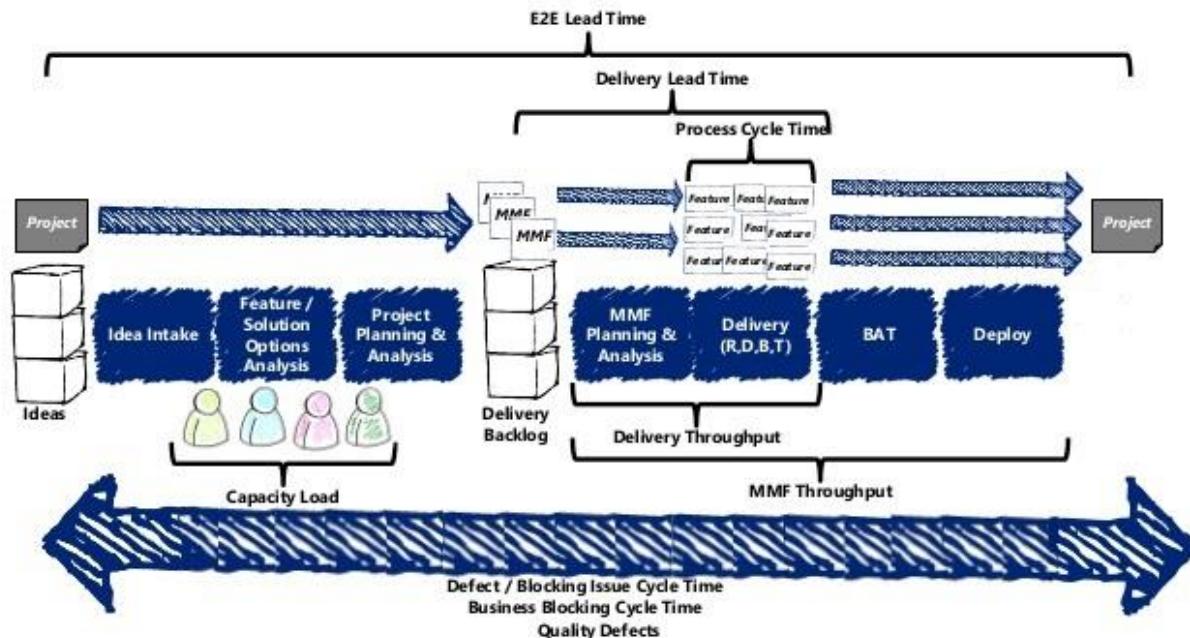


## The Lean Change Canvas



# Lean Metrics Defined

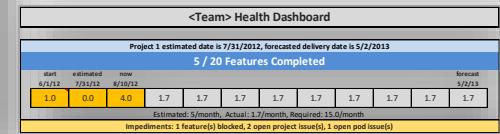
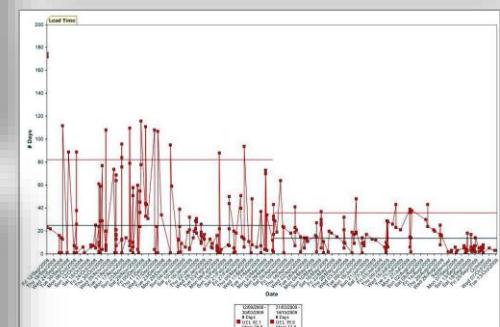
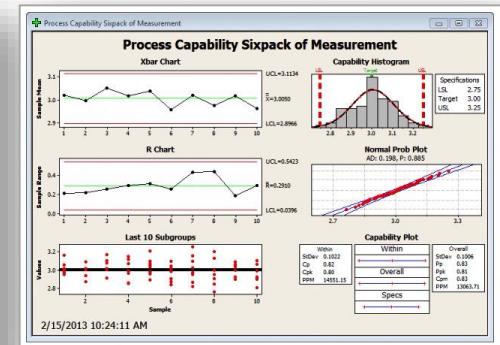
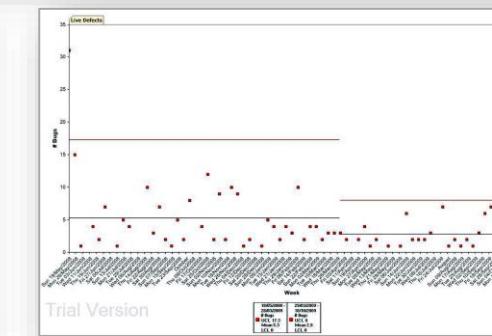
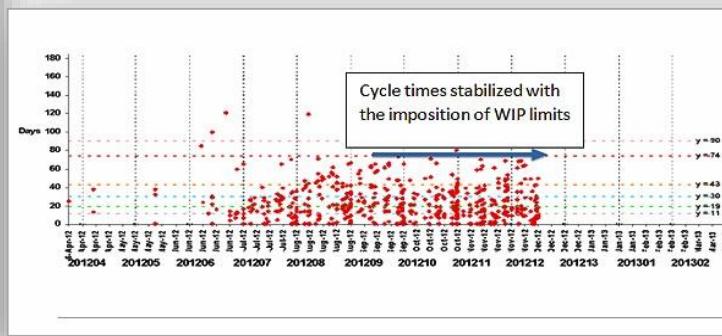
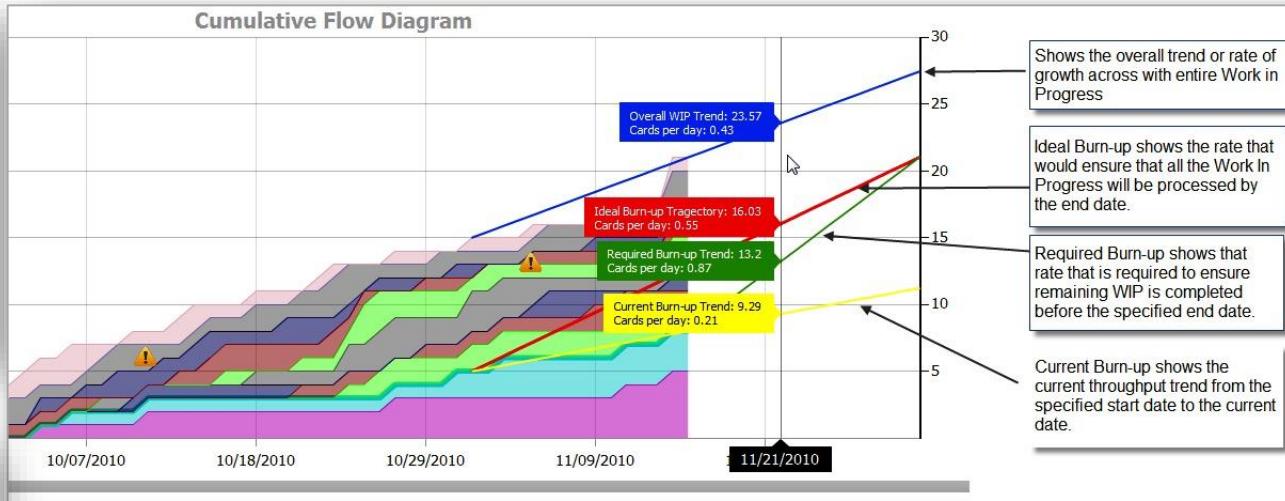
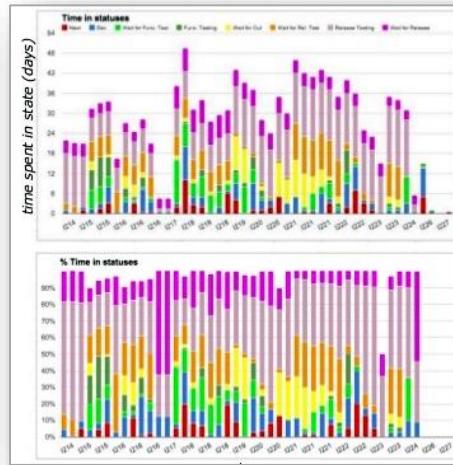
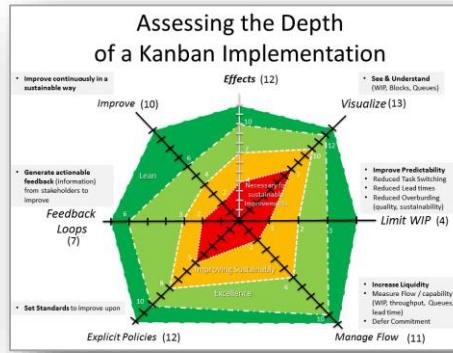
**Lean metrics can be used to measure flow of discrete units of demand, along with predictability, agility and quality**



**Lean Metric enable an organization to measuring the effectiveness of processing multiple types of demand**

Metric	Description	Measures
<b>Delivery Lead Time</b>	Elapsed time of a Story between the Product Backlog and Delivery Complete Provide an understanding of the minimum time it was required to service a request.	Agility
<b>Wait Time</b>	Elapsed time a story has spent waiting for a downstream team to begin work. Provide insight into the efficiency of hand-offs between member	Agility
<b>Delivery Throughput</b>	The number of story ready for release per month (or desired frequency) Provides understanding of overall efficiency and productivity	Capacity
<b>Capacity Load</b>	The ratio of individual work items being processed to number of people in the organization Measures overall organizational maturity, and serves as a leading indicator for cycle times, and predictability as well as quality	Collaboration
<b>Story Target Conformance</b>	The percentage of stories that were completed within the agreed upon estimate (i.e. estimate vs. actual). Enables a meaningful comparison between the initial estimate and the actual result. Allows better estimation of similar stories in the future	Predictability
<b>Defect / Blocking Issue Cycle Time</b>	The amount of time it takes to resolve a defect or an issue that is preventing work from being completed. Measures the capability of managers and executives to resolve problems	Quality
<b>Blocking Rate</b>	Percentage of stories in progress that have been blocked on one or more occasions during a specific time period. Shows how well the organization is a coordinating work that has been prioritized across the enterprise	Collaboration
<b>Failure intake</b>	Percentage of stories in progress that entered the system of work because of previous story was not properly implemented the first time. Points to some optimizations of efficiency over quality, and number of defects that have escaped to the market.	Quality
<b>Defect Density</b>	Percentage of stories in progress that have one or more defects raised against them anywhere in the process. Indicates poor collaboration and communication across different teams.	Quality

# Process Capability Charts Sample





# Who?...Two Pizzas Team



7 People with right blend of personalities, attitudes, and skills with the following roles:

- 1 Leader and Key Liaison with “external world”, both within IT and with Business
- 2 Systems Analysts/Designers/Developers
- 4 Developers



Owns set of selected Front-End fairly isolated applications from the rest of Applications Portfolio with the following key attributes:

- Serving area of Business Executive stakeholder who should be good prospect for open cooperation
- Right mix of operational deployment risks
- Potential to create appropriate volume of workload for the team of this size



3 Sources of backlog:

- Projects Portfolio – dominant to begin with
- IT Orders – growing in scale in Phase 3
- Maintenance Requests, Bug Fixes

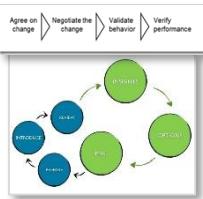


Upstream entry point – Initially Design. Will move up as Team matures

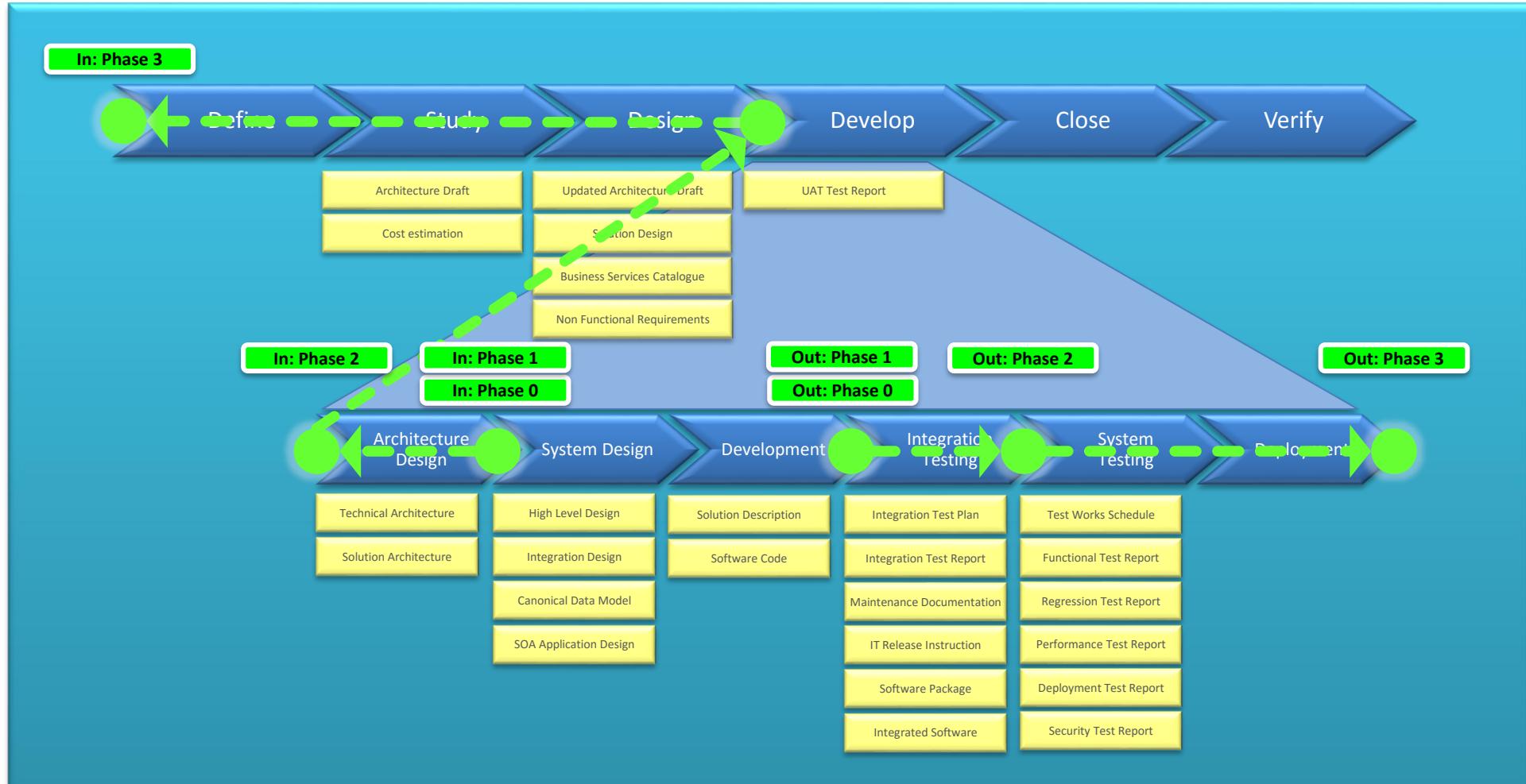


Downstream exit point – operational deployment risk based selection.

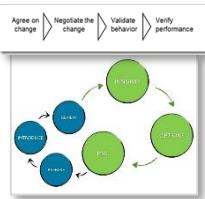
Initially Integration Testing. Will move down as Team and automation (Continuous Integration, Continuous Testing, Continuous Deployment, and DevOps) matures



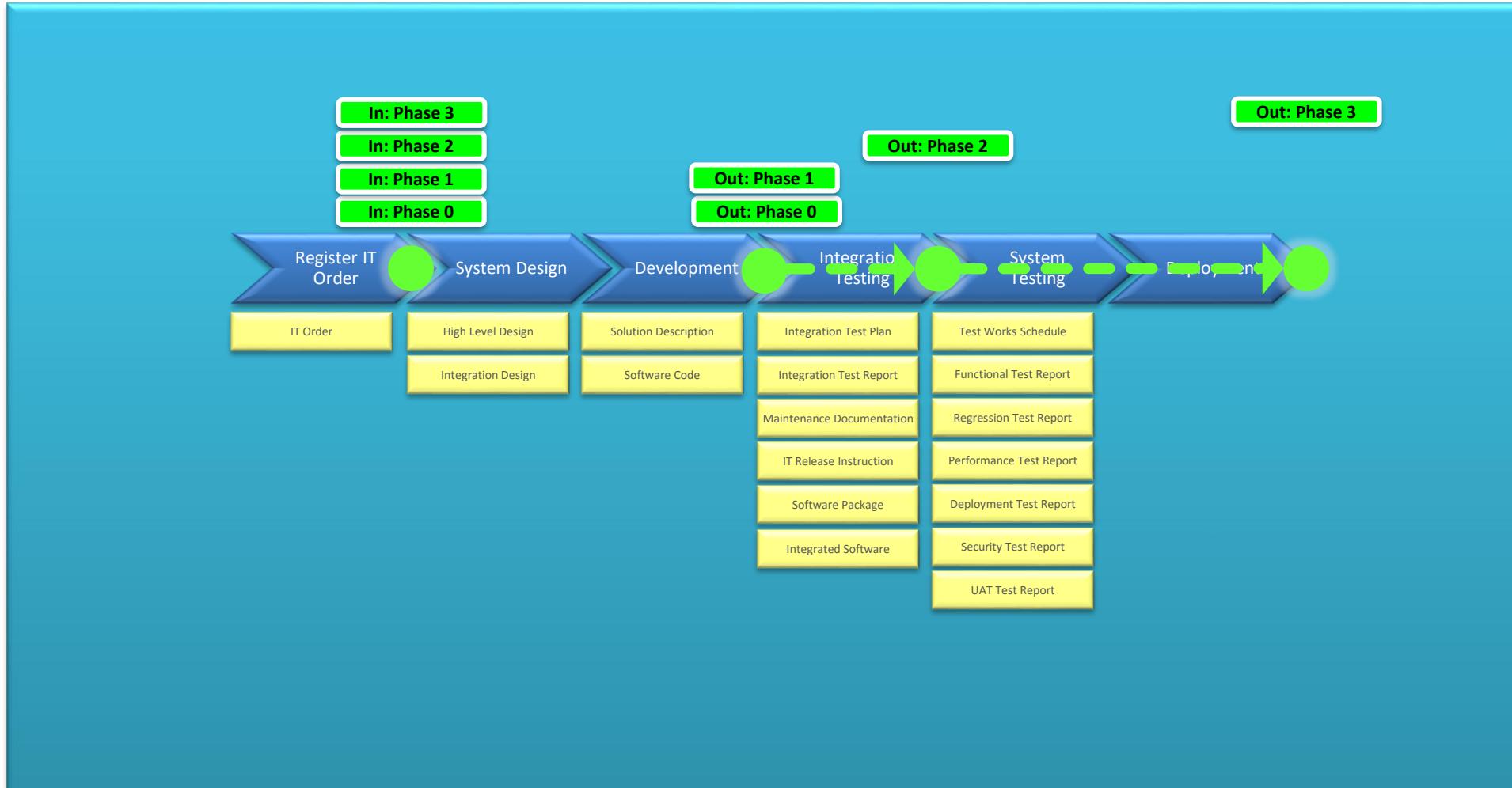
# Upstream Entry & Downstream Exit Points Expansion For Projects



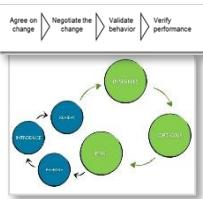
Governance Frameworks, Principles, Practices And Techniques Applied, As Well As Artifacts Created Between Upstream Entry & Downstream Exit Points For Full Mode 2 Projects Evolve Toward Agile Specific, As Per Results Of Experiments From Kanban Improvement Boards



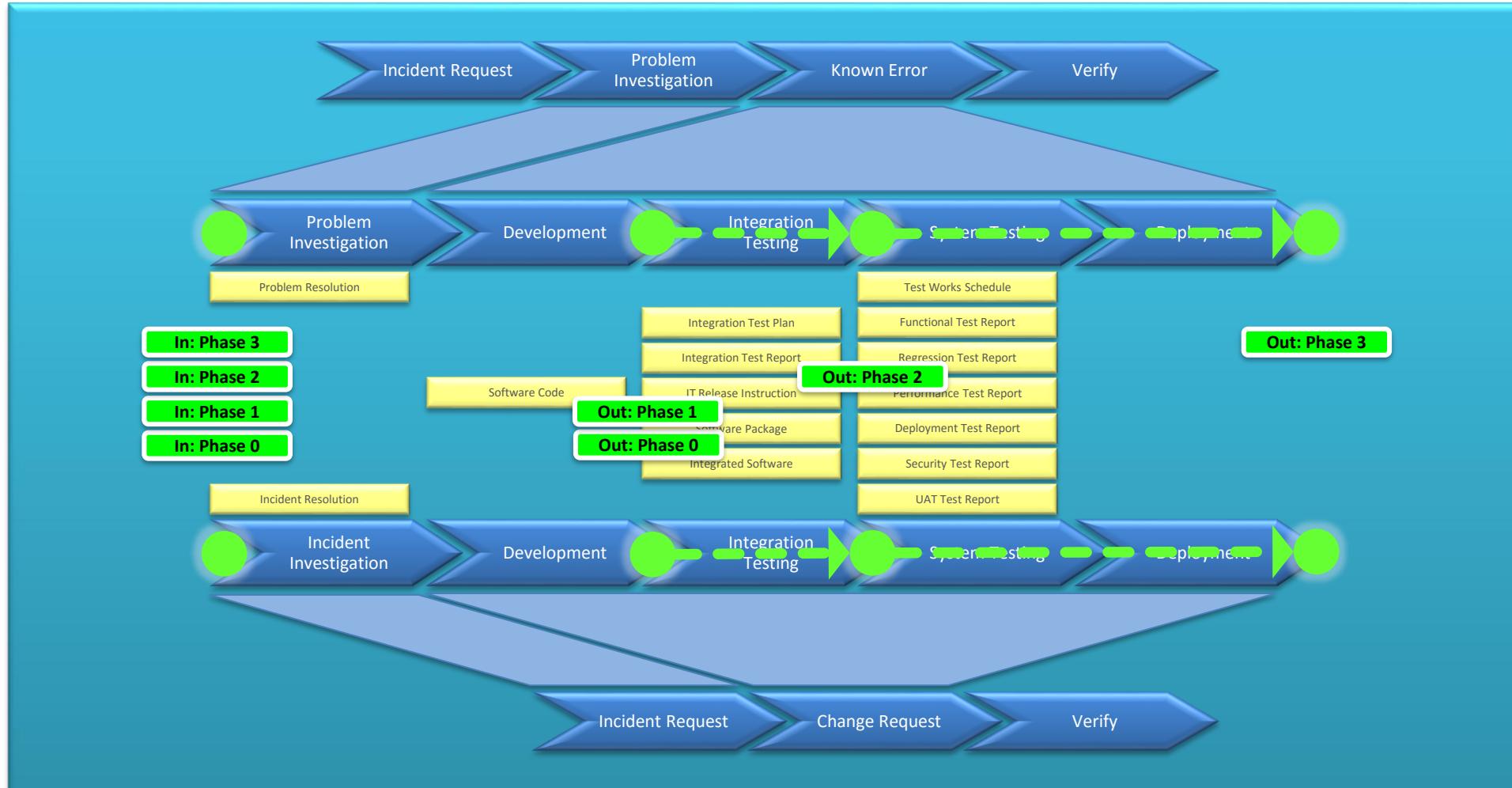
# Upstream Entry & Downstream Exit Points Expansion For IT Orders (Small Projects)



Governance Frameworks, Principles, Practices And Techniques Applied, As Well As Artifacts Created Between Upstream Entry & Downstream Exit Points For Full Mode 2 Projects Evolve Toward Agile Specific, As Per Results Of Experiments From Kanban Improvement Boards



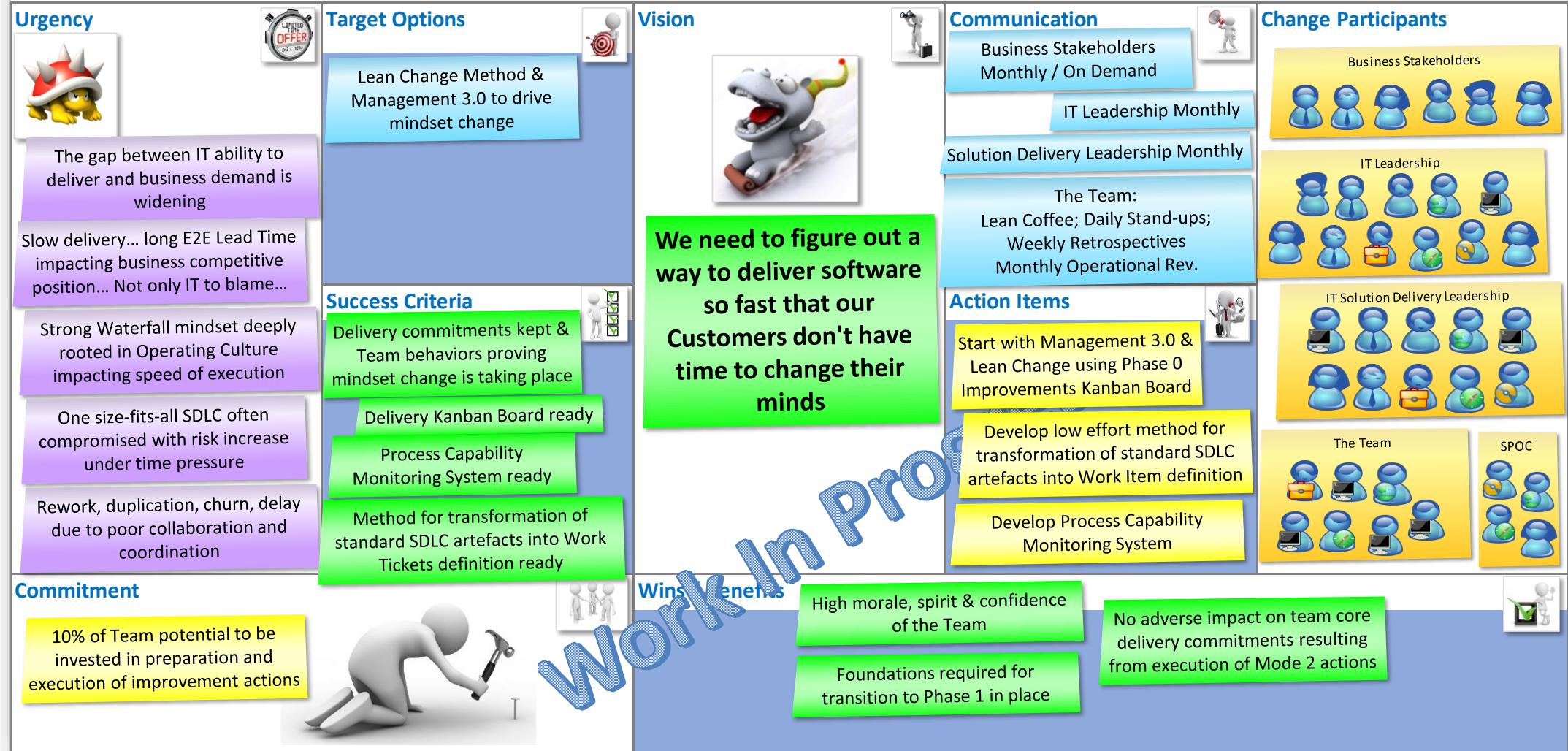
# Upstream Entry & Downstream Exit Points Expansion For Maintenance



Governance Frameworks, Principles, Practices And Techniques Applied, As Well As Artifacts Created Between Upstream Entry & Downstream Exit Points For Full Mode 2 Projects Evolve Toward Agile Specific, As Per Results Of Experiments From Kanban Improvement Boards

## The Lean Change Canvas

## Phase 0 – Team Setup



# Phase 0 – Improvements Kanban Board

Improvements Backlog	Next [2]	Prepare [2]		Introduce [2]		Learn	
		Doing	Done	Doing	Done	Doing	Done
Detailed Value Stream Mapping							Pursue & Scale
Delivery Kanban Board – No WIP limits							
Process Capability Measurement System							
Epics, Features, User Stories for Intake decomposition into Work Tickets							
Business Use Case for Intake decomposition into Work Tickets							Pivot & Adjust
Planning Poker for Work Ticket complexity estimation							
T-shirt size Work Ticket complexity estimation							
Pirate Metrics							
Defect Class of Service	Expedite Class of Service	Fixed Date Class of Service	Standard Class of Service	Intangible Class of Service	 <p><i>Work In Progress</i></p> <ul style="list-style-type: none"> <li>Policies:</li> <li>- Definition of Done <ul style="list-style-type: none"> <li>- Pull priority</li> <li>- Other...</li> </ul> </li> </ul>		

## The Lean Change Canvas

## Phase 1 – As Is Baseline and Building the Basics

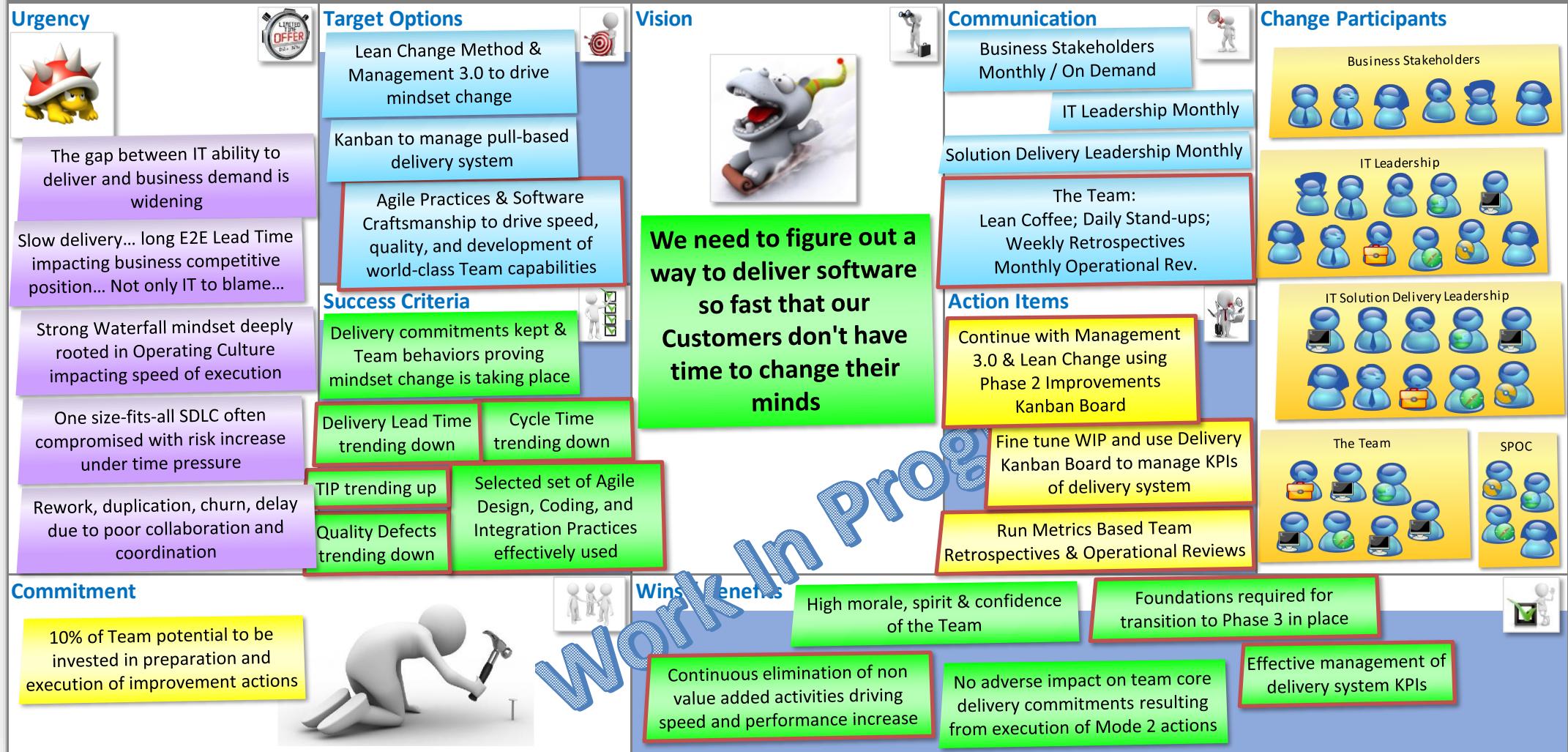


# Phase 1 – Improvements Kanban Board

Improvements Backlog	Next [2]	Prepare [2]		Introduce [2]		Learn	
		Doing	Done	Doing	Done	Doing	Done
Clean Code							<a href="#">Pursue &amp; Scale</a>
SOLID Principles							
Collective Code Ownership							
Pair Programming							
Refactoring – Mikado Method							
Coding Kata							
Coding Dojo							
Code Retreat							
Net Promotor Score Introduction							
Concept of Team Stand-ups							
Concept of Team Retrospectives							
Concept of Team Operations Review							
Defect Class of Service							
Expedite Class of Service							
Fixed Date Class of Service							
Standard Class of Service							
Intangible Class of Service							
Policies: - Definition of Done - Pull priority - Other...							

## The Lean Change Canvas

## Phase 2 – To Be Progressive Development &amp; Roll-out



## Phase 2 – Improvements Kanban Board

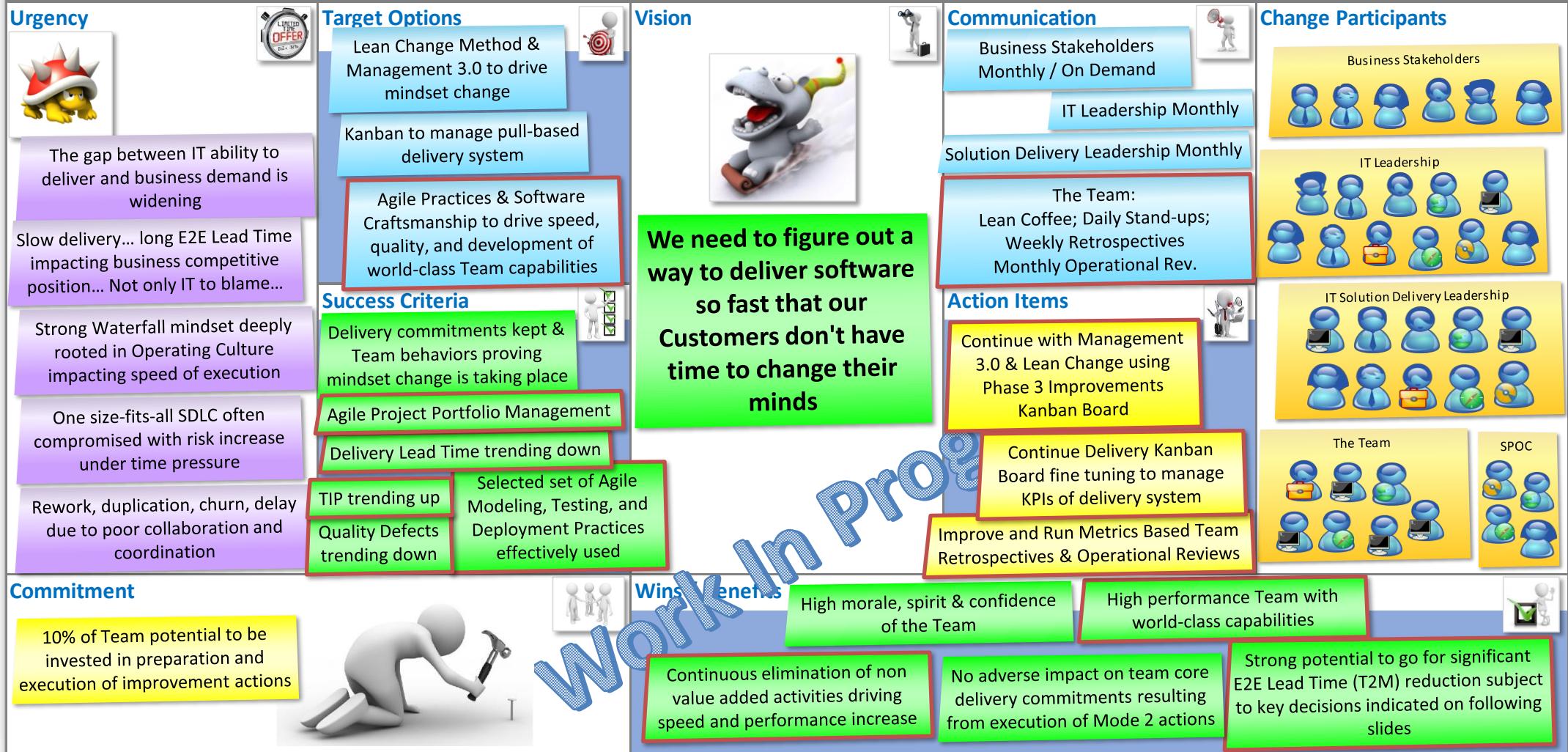
Improvements Backlog	Next [2]	Prepare [2]		Introduce [2]		Learn	
		Doing	Done	Doing	Done	Doing	Done
Delivery Kanban Board – tuning WIP limits							<a href="#">Pursue &amp; Scale</a>
Run Metric Based Team Retrospective							
Run Metric Based Operational Review							
(Acceptance) Test Driven Development							
Behavior Driven Design							
Domain Driven Design							
Design Patterns							<a href="#">Pivot &amp; Adjust</a>
Automated code quality inspections							
Automated Unit Testing							
Stubbing Dependencies							
Automated Builds							
Continuous Integration							
Defect Class of Service							
Expedite Class of Service							
Fixed Date Class of Service							
Standard Class of Service							
Intangible Class of Service							
Policies: - Definition of Done - Pull priority - Other...							

Work In Progress



## The Lean Change Canvas

## Phase 3 – Team and Data Driven Continuous Improvement



## Phase 3 – Improvements Kanban Board

Improvements Backlog	Next [2]	Prepare [2]		Introduce [2]		Learn	
		Doing	Done	Doing	Done	Doing	Done
Agile Project Portfolio Management							<a href="#">Pursue &amp; Scale</a>
Architecture Envisioning							
Requirements Envisioning							
Model Storming							
Agile Data Modeling							
Class Responsibility Collaboration (CRC) Modeling							
Story Mapping for MMF Set definition							
Integrated Development & QA							
Automated Acceptance Testing							
Continuous Deployment							
DevOps for Agile Applications Portfolio							
							<a href="#">Pivot &amp; Adjust</a>
		<i>Work In Progress</i>					
<a href="#">Defect Class of Service</a>		<a href="#">Expedite Class of Service</a>	<a href="#">Fixed Date Class of Service</a>	<a href="#">Standard Class of Service</a>	<a href="#">Intangible Class of Service</a>	Policies: - Definition of Done - Pull priority - Other...	

???



# Let's Do IT Wimba Way



# Acknowledgements

While working on this material, I have been inspired and heavily influenced by the thinking of the following thought leaders of the agile movement, and their respective publications:

- Jeff Anderson - The Lean Change Method, Transforming Your Technology Business through Co-Creation and Validated Learning
- Jason Little - Lean Change Management
- Jurgen Appelo – Management 3.0, Leading Agile Developers, Developing Agile Leaders
- David J. Anderson - Kanban, Successful Evolutionary Change for Your Technology Business
- Mike Burrows - Kanban from the Inside
- Michael Sahota – numerous posts on agilitrix.com
- Eric Ries - The Lean Startup
- Mary Poppendieck and Tom Poppendieck - Lean Software Development, An Agile Toolkit
- Dean Leffingwell - Agile Software Requirements
- Dean Leffingwell - Scaling Software Agility
- Ellen Gottesdiener and Mary Gorman - Discover to Deliver: Agile Product Planning and Analysis
- Mike Cohn - Agile Estimating and Planning
- Mike Cohn – User Stories Applied for Agile Software Development
- Michael Nir - Agile Project Management
- Donald G. Reinertsen - The Principles of Product Flow
- Gene Kim and Kevin Behr and George Soafford - The Phoenix Project
- Craig Larman and Bas Vodde - Scaling Lean & Agile Development
- John P. Kotter - Leading Change
- Eliyahu M. Goldratt and Jeff Cox - The Goal: A Process of Ongoing Improvement
- Eliyahu M. Goldratt - Critical Chain: A Business Novel
- Scott W. Ambler and Mark Lines - Disciplined Agile Delivery
- Mary Mesaglio and Simon Mingay, Gartner - Bimodal IT: How to Be Digitally Agile Without Making a Mess