Adopting an Agile Culture

IN TODAY'S BRICKS & MORTAR COMPANIES

Written and Presented By: Perry J. McLeod, PMP, CBAP, ACP, PBA, ACP
Perry J. McLeod

Senior Management Consultant, Facilitator, and Educator with over twenty-five years of experience in business analysis, workshops and facilitation, agile adoption, process re-engineering, project management, business and systems modeling, strategic alignment, mentoring and training.

In his forthcoming book, Perry discusses his communication framework REPAC; an in-depth treatise designed to help project teams center their efforts on stakeholder’s needs and what is required to fulfill them. Seen through the lenses of poetic naturalism, organizational and behavioral psychology, propositional logic, and communication theory, Perry uniquely argues that all discussions require the right context, focus, perspective, and depth of analysis to build a point-of-view language-set from which atomic, unambiguous imperatives emerge.
“All discussions require the right context, focus, perspective, and depth of analysis to build a point-of-view language-set from which atomic, unambiguous imperatives emerge.”

Tonight’s Learning Points

The History of Waterfall

The History of Agile

The Agile Mindset

Creating an Agile Culture
The History of Waterfall
Way Back in 1970...

Computer scientist Winston Walker Royce published a paper titled: *Managing the development of large software systems*

Royce discussed the use of a “stage-wise waterfall” model for software development.
A 7-Step Method

1. Systems requirements
2. Software requirements
3. Analysis
4. Program design
5. Coding
6. Testing, and
7. Operation
Ah! But There Was a Rub

Dr. Royce saw a problem with this design.

“The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from analyzed. These phenomena are not precisely analyzable.”
Dr. Royce Goes on to Write...

“required design changes are likely to be so disruptive that the software requirements upon which the design is based and which provides the rationale for everything are violated.”
So... Dr. Royce Proposed An Alternative

“Design iterations are never confined to the successive step”

According to Royce,

“This is risky and invites failure.”
Iterate Through...(Stepwise)
What does it mean to “Stepwise?”

“A gradual progression — as apposed to a predictive step by step approach — of regressive, adjacent or progressive steps through a process, based on the analysis and measure of predictive variables.”

It was *Never* a Waterfall

Royce proposed a number of methods to iterate through his design.

However, he never used the term “Waterfall” or advocated it as a workflow
Wait! What’s This You Speak?

“You mean to tell us that we’ve been doing it wrong this whole time?”
Wait! What’s This You Speak?

“So, we’ve been repeating the same mistaken method for the last 57-years expecting each attempt will somehow turn out differently?”
Wait! What’s This You Speak?

We are truly “walking on a tightrope of insanity!\(^{[1]}\)”

\(^{[1]}\)The Box - L’Affaire Dumoutier
Wait! What’s This You Speak?

“Yes!” “Your Waterfalls are saddened, lost, alone, and afraid.”

It's time for a change.

We can, you can, do better.
So...now what?
Barriers to Agile Adoption

Design your approach with your barriers in mind
As time passes and knowledge increases, risk diminishes.

The Cone of Uncertainty
The History of Agile
Way Back in 2001...

At The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen experts in the disciplines of SCRUM, Extreme, Crystal, among others, sympathetic to the need for an alternative to documentation driven, heavyweight software development processes met to talk.

What emerged was the Agile Software Development Manifesto
Agile Is Emergent and Evolutionary

In 2005, an addendum of project management principles, the Declaration of Interdependence, was developed to guide software project management according to agile software development methods.

In 2009, an extension of software development principles, the Software Craftsmanship Manifesto, was developed to guide agile software development.

In 2011, the Agile Alliance created the Guide to Agile Practices (renamed the Agile Glossary in 2016), as an evolving open-source compendium of the working definitions of agile practices, terms, and elements.
The Agile Principles

1. Individuals and Interactions over processes and tools
2. Working Software over comprehensive documentation
3. Customer Collaboration over contract negotiation
4. Responding to Change over following a plan
The Twelve Principles

1. Customer satisfaction by early and continuous delivery of valuable software

2. Welcome changing requirements, even in late development

3. Working software is delivered frequently (weeks rather than months)

4. Close, daily cooperation between business people and developers

5. Projects are built around motivated individuals, who should be trusted

6. Face-to-face conversation is the best form of communication (co-location)
The Twelve Principles

7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly
So...On What Should We Focus?

Iterative, incremental evolutionary delivery

Efficient and face-to-face communication

Decision through collaboration

Short feedback loops and adaptation cycle

Requirements and design are held to be emergent
The Agile Mindset

- Mindset
- Values
- Principles
- Practices
- Tools & Processes
What is a Mindset?

An attitude, disposition, or mood.

An intention or inclination.

A way of thinking.
Agile is a Way of Being

Positive attitude  Focus on Delivering Value
Thirst for knowledge  Ability to Adapt to Change
Pride of ownership  Respect
Pragmatism  Collaboration
Willingness to fail  Intrinsic Motivation
Fixed Mindset vs. Growth Mindset

Monolithic Worldview

Fixed Mindset

Avoid failure
Desire to look smart
Avoids challenges
Stick to what they know
Feedback and criticism is personal
They don’t change or improve

Growth Mindset

Desire continuous learning
Confront uncertainties.
Embracing challenges
Not afraid to fail
Put lots of effort to learn
Feedback is about current capabilities

Polyolithic Worldview

I believe that my [Intelligence, Personality, Character] is inherent and static. Locked-down or fixed. My potential is determined at birth. It doesn’t change.

I believe that my [Intelligence, Personality, Character] can be continuously developed. My true potential is unknown and unknowable.
Predictive Mindset vs. Adaptive Mindset

**Fixed Mindset**

- **Approach to managing uncertainty**
  - Reducing uncertainty by “nailing things down.”
  - Looking to fix and confirm things.

**Agile Mindset**

- **Approach to managing uncertainty**
  - Reducing uncertainty by discovering and learning.
  - Looking to learn and discover in the most efficient way possible.

This is a Fallacy because project work is stochastic.
Projects Are Stochastic

“Because we can remember what happened yesterday, but we cannot remember what happened tomorrow, projects have random probability distributions or patterns which may be analyzed statistically but may not be predicted precisely.” — me! 😊
How The Agile Mind Produces Value

**Fixed Mindset** approach to delivery (Assembly Line)
Must "nail down" the output in order to start delivery (Linear Thinking)

**Growth Mindset** approach to delivery (Knowledge Work)
Discover and learn through valuable output and welcoming change (Circular Thinking – IKIWISI)
How The Agile Mind Releases Value
Creating an Agile Culture
Organizational Agility

*Culture* based on the *values* and *principles* of Agile, supported by the *organizational ecosystem* and manifested through *personal and organizational habits*. 
The Organizational Ecosystem
Sustainable Agility
Create a Common Educational Journey
The Human Element

Common education journey (not training) to change how people work and illustrate how to live the Agile Mindset in their job

Leadership Coaching (inspire performance not mandate it)

Mentoring and Coaching on an individual and team level.
Non-Human Elements

Designing and implementing a multi-stage roadmap to agility that changes all three of these element in synergy and harmony

A combination of consulting, mentoring, organizational coaching, business process re-engineering and organizational change management to roll-out the changes across the organization
Measurement Elements

Establishing a measurement system that is consistently monitoring the alignment of the culture

Primary measure of progress is the mindset shift and the transformation of personal and organizational work habits

Reporting progress, as a function of culture change not process change, nor structure change.
The Requirement Element

Requirements represent needs. Designs represent solutions.

Ask not, “what is the requirement?” Ask, “what is required to fulfill the need the design the solution.”
The Agile Maturity Model

<table>
<thead>
<tr>
<th>Adhoc</th>
<th>Defined</th>
<th>Managed</th>
<th>Optimize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inconsistency in delivery of quality software code</td>
<td>Setup a software quality governance team</td>
<td>Setup framework for monitoring &amp; measurement software quality</td>
<td>Monitor &amp; measure software quality characteristics (Testability, Reusability, Reliability etc)</td>
</tr>
<tr>
<td>Lack of unit tests coverage</td>
<td>Regular training on software quality, unit testing &amp; agile coding techniques</td>
<td>Setup tool for static (automated) &amp; manual code reviews</td>
<td>Enforce usage of code review tools for static and manual reviews</td>
</tr>
<tr>
<td>Lack of management support for unit tests owing to delivery time</td>
<td>Define Software quality goals for teams</td>
<td>Enforce design &amp; code reviews tasks while planning &amp; estimation</td>
<td>Optimize learning from design &amp; code reviews with regular workshops</td>
</tr>
<tr>
<td>Lack of automated build &amp; deployment process</td>
<td>Encourage unit testing with management support;</td>
<td>Enforce unit testing tasks while planning &amp; estimation</td>
<td></td>
</tr>
<tr>
<td>Lack of team awareness &amp; knowledge regarding agile coding techniques</td>
<td>Setup code review guidelines &amp; best practices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of monitoring &amp; measurement on code quality</td>
<td>Setup automated build &amp; deployment process with tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Agile Maturity Curve

Level 1: Limited
- Low competencies and skill level in Agile
- Undefined roles and responsibilities with no shared ownership or accountability
- Lack of individual incentives for Agile adoption
- No processes in place for identifying, initiating, and mobilizing Agile projects
- Limited use of Agile practices and tools for managing user stories, backlog, and testing
- Intra- and inter-project communication is not explicit
- No formal Agile training
- Limited visibility into team capacity
- Team members are not accountable for outcomes
- Limited business support for Agile
- Limited willingness to make changes

Level 2: Evolving
- Skill level in Agile varies by project team and functional department
- Agile roles and responsibilities are inconsistent
- Some processes are in place for identifying, initiating, and mobilizing Agile projects, but not always followed
- Agile teams leverage tools for build and test automation, managing user stories, and have flexibility in installing new tools
- Some involvement between business and development teams
- Limited business support for Agile
- Limited willingness to make changes

Level 3: Project Excellence
- Consistent level of skill and competency in Agile across project teams
- Well-defined roles and responsibilities, and shared sense of ownership at the team level
- Documented processes are in place for identifying, initiating, estimating, and mobilizing Agile projects
- Agile teams are co-located, can support off-shore team members, adopt key agile practices and leverage common integration, delivery, and testing practices
- Business stakeholders communicate directly and frequently with the development teams
- There is visibility into team capacity across departments
- Enterprise performance measures support and encourage Agile collaboration

Level 4: Integrated Excellence
- Documented processes in place to measure quality and performance of Agile projects
- Adoption of Agile practices such as Continuous Delivery is pervasive in Agile projects
- Organizational policies and procedures support Agile adoption
- Key stakeholders are actively involved in the success of Agile
- The organization is adaptable and collaborative towards Agile and embraces change
- People, Process, and Tools & Practices
We must **be** agile before we can **do** agile.