Agenda

➢ Introductions
➢ YRC Worldwide and Agile Audit
➢ Agile Defined
➢ Agile and Scrum Concepts and Framework
➢ Agile in Practice
➢ Key Learnings
➢ Q&A
YRC Worldwide and Agile Audit

➢ Holding company with four less than truckload “LTL” operating companies and a freight brokerage company, HNRY Logistics

Each operating company has similar facility audit programs
Some audit areas were being missed consistently across the network
YRCW IA sought out to find fresh way to address recurring themes
  ▪ “Agile approach”
    ▪ Iterative
    ▪ Collaborative
    ▪ Root Cause and Solutions Oriented
What does it mean to be agile?

➢ **Webster's online definition of agile:**
  - Marked by ready ability to move with quick easy grace
  - Having a quick resourceful and adaptable character

➢ **Wikipedia definition of agile software development:**

An approach to software development under which requirements and solutions evolve through the **collaborative effort of self-organizing and cross-functional teams** and their customer(s)/end user(s). It advocates **adaptive planning**, evolutionary development, early delivery, and **continual improvement**, and it encourages **rapid** and **flexible** response to change.

➢ **Association for Project Management’s definition of agile project management:**

An **iterative** and **incremental** approach to delivering requirements throughout the project life cycle. At the core, agile projects should exhibit central values and **behaviours of trust**, flexibility, empowerment and collaboration.

➢ **Other mentions of agile:**

A set of guiding principles, a mindset, a revolution, a fad
The Need for Frequent, Effective Communication and Collaboration

How the customer explained it.

How the project leader understood it.

How the analyst designed it.

How the programmer wrote it.

What the customer really wanted.
Making the IA process better by reducing inefficiencies in order to be more “agile” with the goal of addressing dynamic risk landscapes in a nimble way
• Process improvement methods
• Rationalization of documentation
• Flexible and nimble audit plan and risk assessment

An innovative IA approach that transforms the IA process using Agile software development values, principles, frameworks, and methods with the goal of addressing dynamic risk landscapes in a nimble and collaborative way
• Common frameworks = Scrum, Kanban
• Iterative and incremental planning and reporting
• Trusted, empowered, and self-organizing audit teams
• Mindset shift
What does it mean to be agile?

➢ Who uses agile / Agile?
  ▪ IBM, Microsoft, Cisco, AT&T, Fidelity, Walmart

➢ Why?
  ▪ Rapid adaptation to change - the current pace of tech innovation (or more broadly, business) can change an industry quickly, or even revolutionize it
  ▪ Increased collaboration and creativity
  ▪ Removal of Silos
  ▪ Increased speed to market
  ▪ Frequent checkpoints to ensure value is provided

Source: [https://artplusmarketing.com/this-is-why-fortune-500-companies-use-an-agile-approach-to-process-improvement-d4095d410a61](https://artplusmarketing.com/this-is-why-fortune-500-companies-use-an-agile-approach-to-process-improvement-d4095d410a61)
Agile and Scrum

➢ *Difference between waterfall, agile and scrum*

➢ **Waterfall Model** methodology which is also known as Liner Sequential Life Cycle Model. The Waterfall Model follows in *sequential* order, so the model develops systematically from one phase to the next in a downward trajectory.

➢ **Agile** describes a set of *guiding principles* that uses iterative approach for software development
  
  ▪ Agile is a practice that helps *continuous iteration* of development and testing in the software development process. In this model, development and testing activities are *concurrent*, unlike the Waterfall model. This mindset allows more *communication* between customers, developers, managers, and testers.

➢ **Scrum** is a *frame work of specific set of rules* that are to be followed while practicing Agile with rigid timeboxes and specified ceremonies.
Brief history of Scrum and Agile

➢ History of Scrum

- In 1986 Japanese business experts Hirotaka Takeuchi and Ikujiro Nonaka published the article, “New New Product Development Game” noting scrum as a new approach to commercial product development
- Result was a framework that focuses on what needs to be done, not how to do it
- 3 pillars were identified:
  - Transparency - visibility to aspects of process responsible for the outcome
  - Inspection - frequent checks on process to detect undesirable variance
  - Adaptation - adjusting to process as timely as possible

➢ History of Agile

- In 2001, 17 people in the programming world got together in the Wasatch Mountains in Utah
- In attendance were Agile pioneers Alistair Cockburn and Ken Schwaber
- Results were the Agile Manifesto and changes in the way of thinking.
Manifesto for Agile Software Development

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

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

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

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

© 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

Twelve Principles of Agile Software

View Signatories
Principles behind the Agile Manifesto

We follow these principles:

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

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

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

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

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

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

Working software is the primary measure of progress.

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

Continuous attention to technical excellence and good design enhances agility.

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

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.
Twelve Principles of Agile Software Development

Guiding principles that affect the thoughts and actions of how work is performed:

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

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

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

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

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

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

7) **Working software** is the primary measure of progress.

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

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

10) **Simplicity**—the art of maximizing the amount of work not done—is essential.

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

12) At **regular intervals**, the team **reflects** on how to become more effective, then tunes and adjusts its behavior accordingly.
Audit: Traditional/Waterfall vs Agile Audit

- Larger scale project planning and execution
- Compile results and report at the end
- Adhere to the established process

- Smaller increments of work
- Show the customer and get feedback
- Work on improving your process
**Attributes of Scrum**

**Sprint Work-cycles (Time-boxes)**
- Consistent duration (2- to 4-week work-cycles)
- New sprint immediately follows the preceding sprint
- Sprint start date and end date are fixed

**Ceremonies**
- Daily stand-up
- Sprint planning
- Backlog refinement
- Sprint review
- Sprint retrospective

**Artifacts**
- Project Canvas
- Backlog
- Sprint backlog
- Release plan / road map

**Roles**
- Product Owner
- Scrum Team
- Scrum Master
Terminology and Tools

- **Epic** – Concludable Area
- **Point of View** – Opinion
- **User Story** – a tool used to capture a description of a desired feature from an individual’s perspective. A user story describes the type of user, what they want and why.

As an ____auditor____, I want to ____perform a RACI analysis____, so that ____I have an understanding of who is responsible, accountable, consulted and informed about relevant processes____.

- **Acceptance Criteria** – specific conditions under which a user story is fulfilled

When I am done, I will ____have documented a complete RACI matrix for the process____.

- Developing acceptance criteria to define done is critical to success.
- Especially since teams are designed to be cross-functional, we have to acknowledge that “done” does not look the same way to everyone.
- Quick exercise - take 30 seconds and think about how you know your laundry is done.
### Terminology and Tools

- **Project Backlog** – collection of user stories and their associated story points, dependencies (if applicable), sprints, etc.

- **Sprint Backlog** – the portion of the project backlog that has been incorporated into a current sprint.

<table>
<thead>
<tr>
<th>STORY #</th>
<th>SPRINT #</th>
<th>EPIC</th>
<th>USER STORY</th>
<th>DEPENDENCIES</th>
<th>STORY POINTS</th>
<th>BACKLOG STATUS</th>
<th>AUDITOR(S)</th>
<th>ASSIGNED</th>
<th>ACCEPTANCE CRITERIA (WE WILL KNOW WE ARE DONE WHEN...)</th>
<th>TASKS</th>
<th>WORK PERFORMED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>As a ______, I want to ________, so that ________</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>As a ______, I want to ________, so that ________</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Planning Poker** – technique for estimating story points (the effort or relative size of a user story).

- **Kanban Board** – visualization tool that typically uses sticky notes on a whiteboard to communicate workflow status and progress.
Scrum Overview

Elaboration / Discovery
- Inputs from CAE, Executive team, Stakeholders, Customers, Users
- Identify concludable areas and confirm stakeholder buy-in
- Audit project canvas

Daily Standup
- Audit Scrum Master
  - What I did yesterday
  - What I’m doing today
  - What are my impediments?
- Audit Scrum team
- Key Stakeholders
- Burndown chart

Sprint Planning
- Audit Scrum Master
  - Sprint Goals / Objectives
  - Define / Clarify
  - Risks / Controls
  - Plan / Procedures
- Audit Scrum team
- Audit Product Owner
- Project Backlog
- Demo
- Point of view (EPIC)

Sprint Review and Demo
- Audit Scrum Master
  - Review completed stories
  - Discuss points of view
  - Gather feedback
  - Update backlog
- Audit Scrum team
- Audit Product Owner
- Key Stakeholders
- Project Report

Roadmap / Release plan
- Audit Product Owner
  - Define project epics
  - Identify features
  - Risk assessment at the epic level
  - Prioritize epics
  - Build the roadmap
  - Prioritize features
- Project backlog
- Project roadmap

Sprint Retrospective
- Audit Scrum Master
  - Actionable (Internal) Process Enhancements
- Audit Scrum Team
  - What I liked
  - What I learned
  - What was lacking
  - What I long for
  - Improvements

2-Week Sprint
Key Learnings – Agile and Scrum in Internal Audit

➢ Socialization and buy-in is key – senior management, audit clients and the scrum team
➢ Benefits of Internal Audit Scrum Master Training
➢ Agile is not one-size-fits-all – variations will occur
  ▪ Company to company
  ▪ Audit client to audit client
  ▪ Audit to audit
  ▪ Scrum team to scrum team
➢ Pilot project for adapting to the framework
➢ Benefits require time
Key Learnings – Agile and Scrum in Internal Audit

Advantages identified from the YRCW implementation

➢ Frequent communication and solicitation of client feedback
  ▪ Stay on task
  ▪ Report more frequently, potential for earlier action plan implementation
  ▪ Increased flexibility and better overall client experience

➢ Rigid ceremonial procedure
  ▪ Check and balance on work performed
  ▪ Removal of impediments
  ▪ Continuous improvement of process

➢ Defined timelines
  ▪ All stakeholders in the audit have a clear understanding of what is to be expected and when

➢ Planning dedication
  ▪ After planning, all team members have a clear understanding of what needs and accomplished and when that requirement must be met
Hurdles to overcome from the YRCW implementation

- **Agile for audit concerns**
  - No perfect parallel audit
    - Some aspects have a “scrummerfall” approach
      - Especially true when approaching the demonstration / reporting phase
  - Inherent difficulty when working remotely
    - Agile embraces face to face interactions
    - Remote teammates have expressed some frustration with physical proximity
  - Right-sizing client involvement
    - The right level of engagement and involvement can be difficult to find
    - Too little involvement leads to lack of commitment to action plan
    - Too much involvement can lead to inefficiencies
  - Organizing Agile process
    - Agile software or Build your own
Resources

Agile Auditing: Transforming the Internal Audit Process,
by Rick Wright, October 2019

Agile Manifesto: https://agilemanifesto.org/
Scrum Alliance: https://www.scrumalliance.org/
caleb.blanding@yrcw.com, mitch.child@yrcw.com

Articles:

➢ Understanding The Agile Mindset, by Steve Denning, August 2019

➢ This Is Why Fortune 500 Companies Use An Agile Approach To Process Improvement,
  by Maria Materelli, November 2017
  https://artplusmarketing.com/this-is-why-fortune-500-companies-use-an-agile-approach-to-process-improvement-d4095d410a61
Questions?