

Scrum - Kanban Cage Match

**Scrum and Kanban
Cage Match**

Ben Day
Benjamin Day Consulting, Inc.

David Starr
Pluralsight

Level: Intermediate

The Tale of the Tape

<p>Scrum Ben Day</p> <ul style="list-style-type: none"> • benday@benday.com • @benday • Blog: blog.benday.com • 175 lbs. • 5'8" 	<p>Kanban David Starr</p> <ul style="list-style-type: none"> • david@pluralsight.com • @elegantcoder • Blog: elegantcode.com • More than Ben • 6'0"
--	--

Both

- Test Driven
- Agile Trainer and Coach
- Scrum.org Trainers
- Microsoft MVP in ALM
- Actual Developer

Professional Scrum at Scrum.org

<p>Professional Scrum Product Owner</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 5px auto; width: 60%;"> <p>Product Owners Executives</p> </div>	<p>Professional Scrum Master</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 5px auto; width: 60%;"> <p>Scrum Masters</p> </div>	<p>Professional Scrum Developer .NET or Java</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 5px auto; width: 60%;"> <p>Architects Business Analysts DB Specialists Designers Developers Testers</p> </div>
<p>Professional Scrum Foundations</p>		<p>Everyone</p>

s1i

Session Agenda

- What is Scrum
- What is Kanban
- How the 2 models address specific scenarios
 - It all comes down to complexity

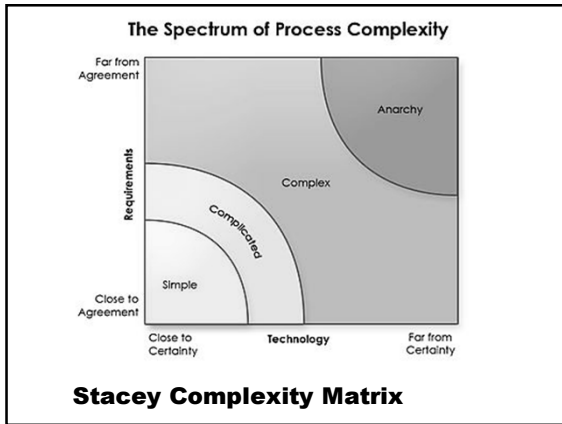
<p>Scrum The Current Champion</p> <ul style="list-style-type: none"> • Used by 88% of Agile organizations * • Iterative / Incremental • Time boxes everything • Introduces a prescribed framework 	<p>Kanban The Challenger</p> <ul style="list-style-type: none"> • Used by hippies and trendy Tweeters • Continuous Flow • Optimizes any process • Starts with what we do today
--	---


* 2010 Forrester Research

Which one?

It all comes down to complexity.

Scrum - Kanban Cage Match



Scrum in a Nut Shell 

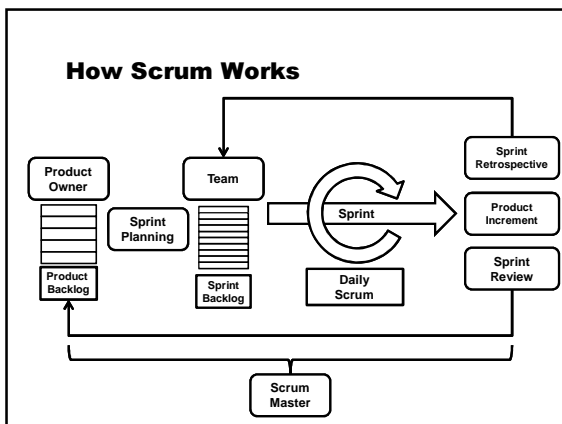
1. A Team commits to delivering working software in 30 days or less
2. A time is scheduled to show that software
3. The Team creates the software
4. Feedback is given to the Team when the software is shown

Kanban in a Nutshell

1. Visualize your current workflow
2. Limit Work in Progress
3. Enable pulling work instead of pushing it
4. Make policies explicit
5. Improve collaboratively

Scrum Language

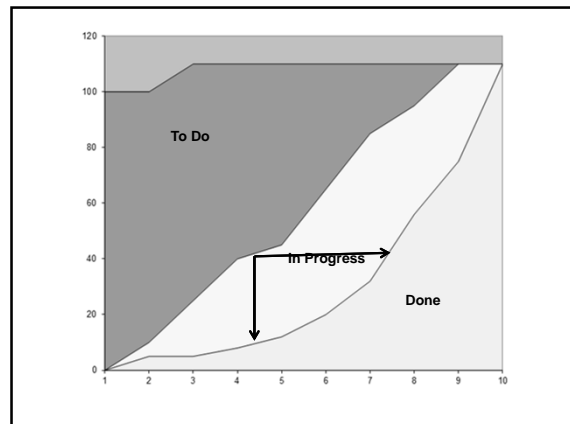
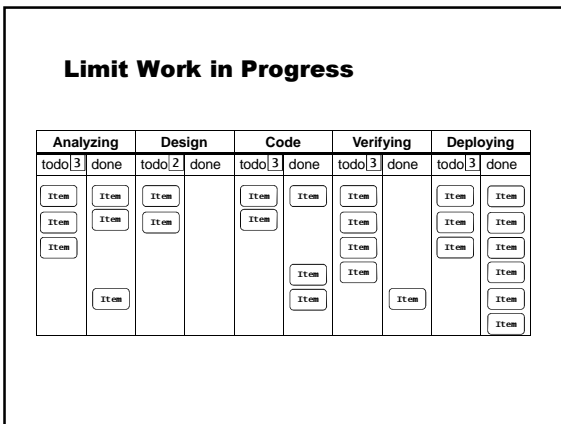
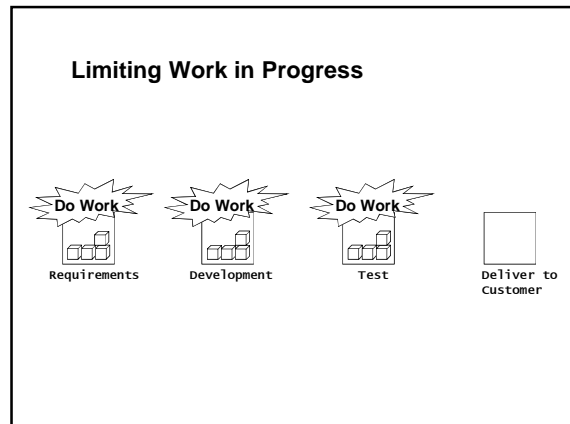
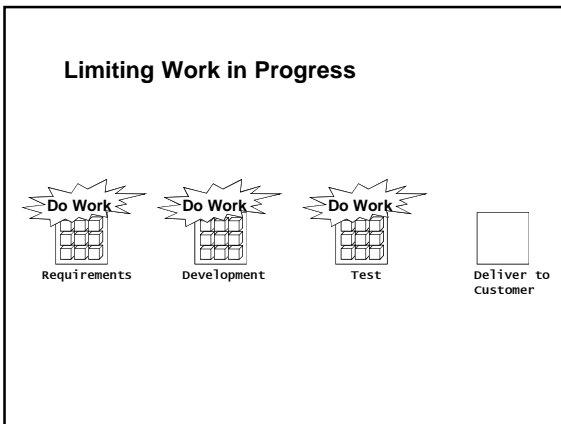
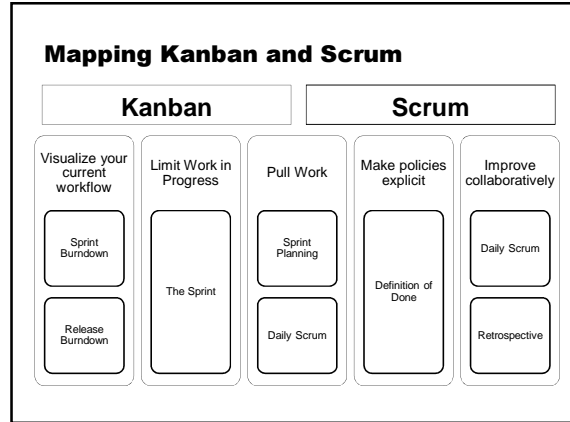
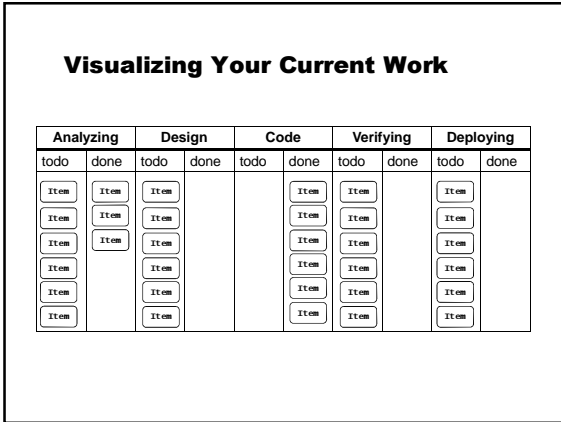
Roles	Artifacts	Events
<ul style="list-style-type: none"> • Product Owner • Team • Scrum Master 	<ul style="list-style-type: none"> • Product Backlog • Sprint Backlog • Increment 	<ul style="list-style-type: none"> • Sprint • Sprint Planning • Daily Scrum • Sprint Review • Retrospective



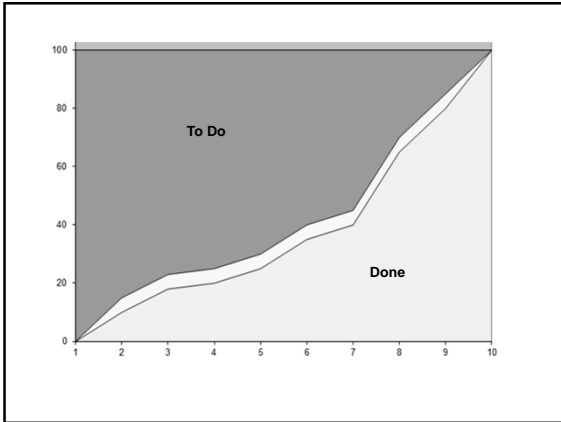
Visualizing Your Current Work

	Analyzing		Design		Code		Verifying		Deploying	
	todo	done	todo	done	todo	done	todo	done	todo	done
Item										

Scrum - Kanban Cage Match




Scrum - Kanban Cage Match



- ### Introducing Scrum in Your Team
1. Map roles to people
 2. Form the Development Team
 3. Create a Product Backlog
 4. Perform 1st Sprint
 5. Perform Sprint Review
 6. Perform Sprint Retrospective

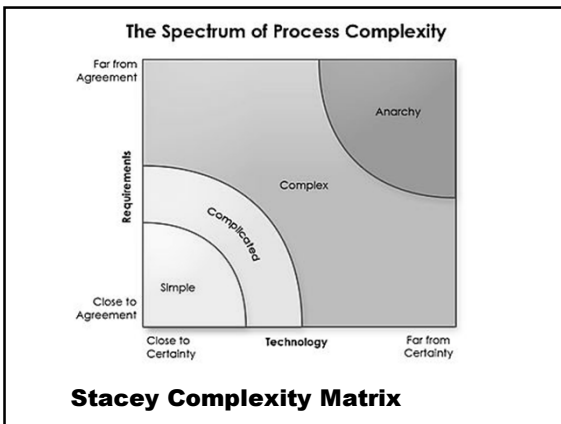
- ### Introducing Kanban in Your Team
1. Create a Kanban board that everyone can see
 2. Help the team understand what it is being shown
 3. Help them see that high WIP is bad

Introducing Change in Your Team

Kaizen Event 

vs.

Quietly



- ### Where is Scrum Appropriate
- Teams moving away from waterfall
 - Product Development environments
 - Organizations needing:
 - To reset culture
 - To change direction quickly
 - To improve software development practices

Scrum - Kanban Cage Match

Where is Kanban Appropriate

- When work is closer to complicated than complex
- Operational, support, and sustained engineering teams
- Places looking to improve without resetting the company in the process

How would the different techniques handle ...

SCENARIOS

**Many defects in
the current
product**

**Lack of
Continuous
Integration**

**We need to change
what the Team is
working on.**

Right now.

**Planning a
release that is
6 months out**

**Requirements are
not understood by
the Team**

**Introducing a new
development
technique,
like TDD**

What are yours?

Thank you

Ben Day	David Starr
<ul style="list-style-type: none">• benday@benday.com• @benday• Blog: blog.benday.com	<ul style="list-style-type: none">• david@elegantcode.com• @elegantcoder• Blog: elegantcode.com
benday.com	pluralsight.com elegantcode.com