iNNolution
agile innovative solutions

Agile Principles:
The Foundation Underlying Successful Agile Development
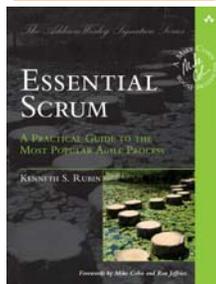# March 25, 2015
by Ken Rubin

☀ Essential Scrum in Six Languages

English	French	German

Chinese	Japanese	Polish
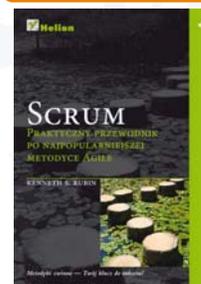
☀ Available on Safari Books Online
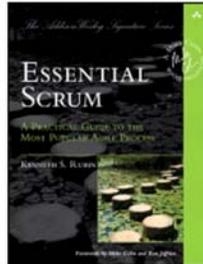
< Return to Search Results ☒

Essential Scrum: A Practical Guide to the Most Popular Agile Process
By: Kenneth S. Rubin
Publisher: Addison-Wesley Professional
Pub. Date: July 26, 2012
Print ISBN-10: 0-13-704329-5
Print ISBN-13: 978-0-13-704329-3
Web ISBN-10: 0-321-70040-6
Web ISBN-13: 978-0-321-70040-7
Pages in Print Edition: 504
Subscriber Rating: ☆☆☆☆☆ [24 Ratings] Subscriber Reviews
My Rating: ☆☆☆☆☆ REVIEW THIS BOOK ★
Popular Tags: agile, scrum   ADD NOTE / TAGS

| Overview | Table of Contents | Notes & Tags | Bookmarks | Search This Book |

3

☀ US Airdrops and Cargo Staging

4

Cargo Cult

Cargo Cult Issues…

Imitating actions alone produces desired results

Correlation implies causation

Process is more important than core principles

No need to know the foundational "**why**"

# Agile Principles Provide Context for Inspecting & Adapting

People who apply Scrum without understanding its underlying principles lack the necessary context to understand why they are doing things and when and how best to inspect and adapt their approaches
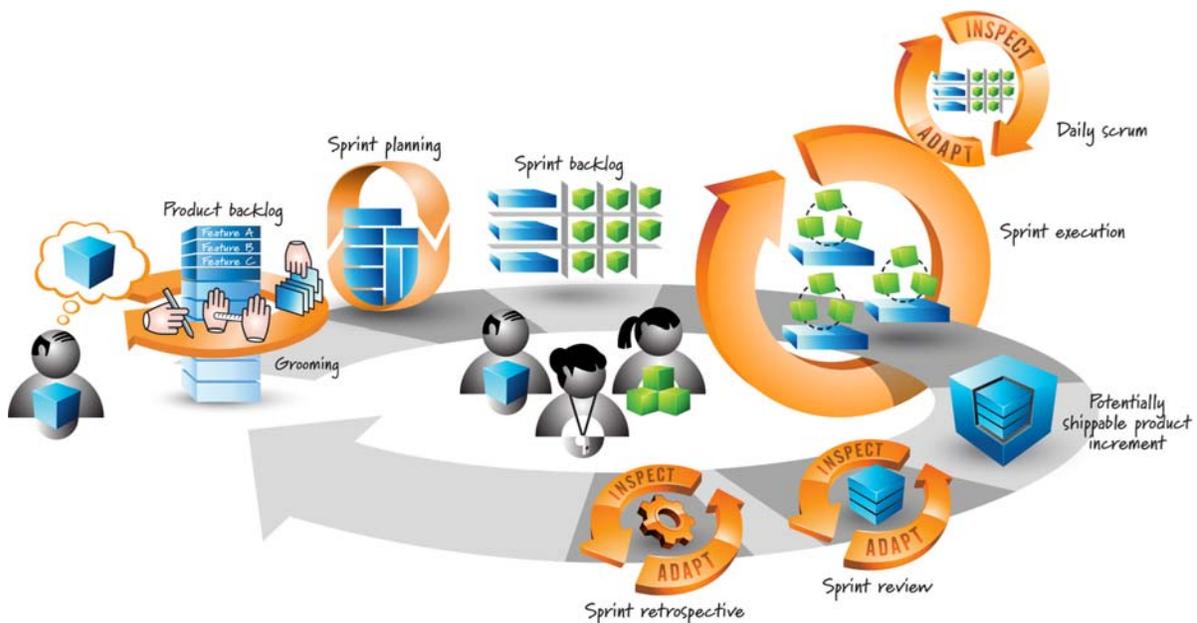

Variability & uncertainty | Prediction & adaptation | Validated learning | Work in Process (WIP) | Progress | Performance

Waterfall and Agile – Two Tools in the Toolbox

Traditional, Phased-based, Plan-driven Development (aka Waterfall)
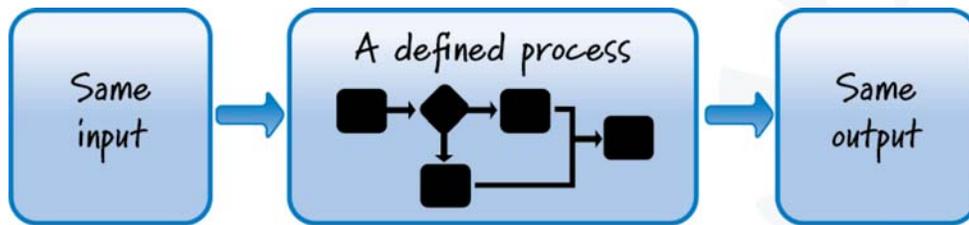
- Deliverable/Milestone
- Review/Approval
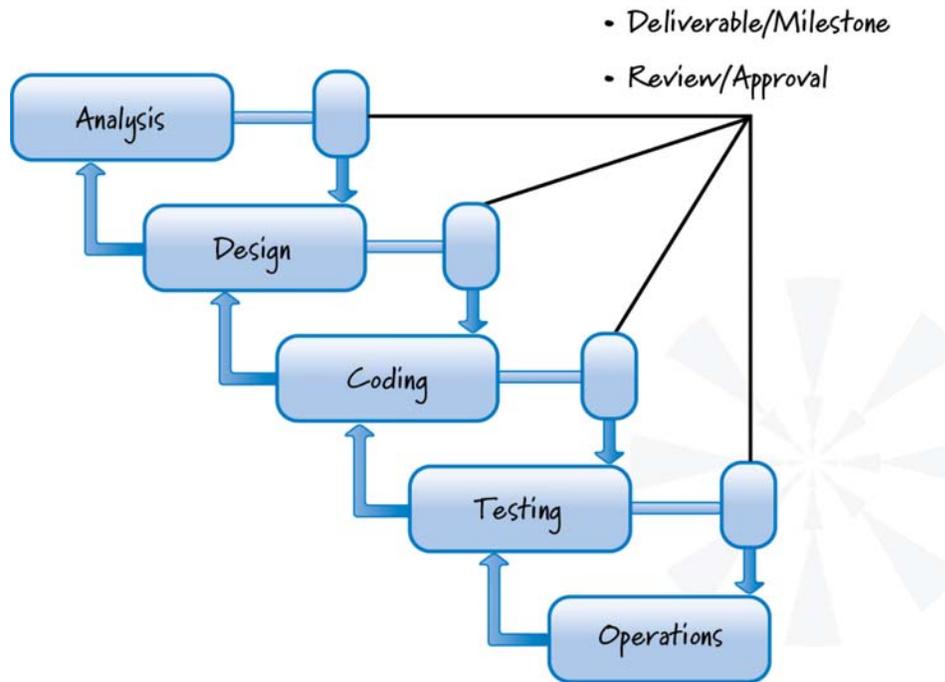
Agile Development — The Scrum Framework

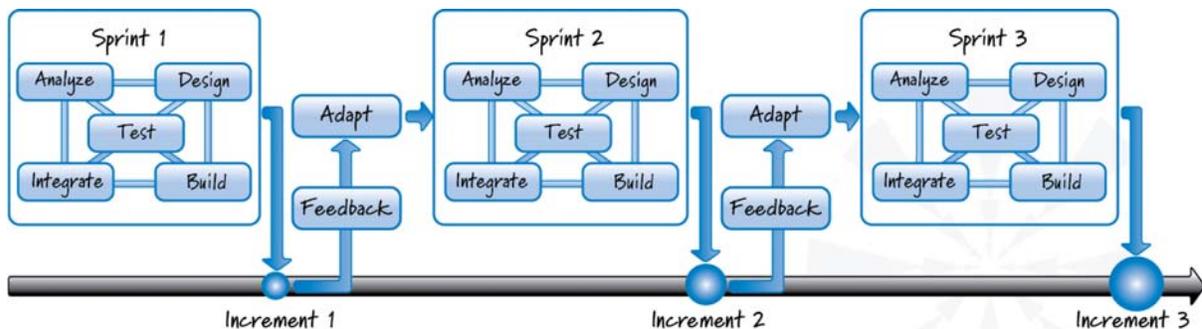# Development Isn't Manufacturing
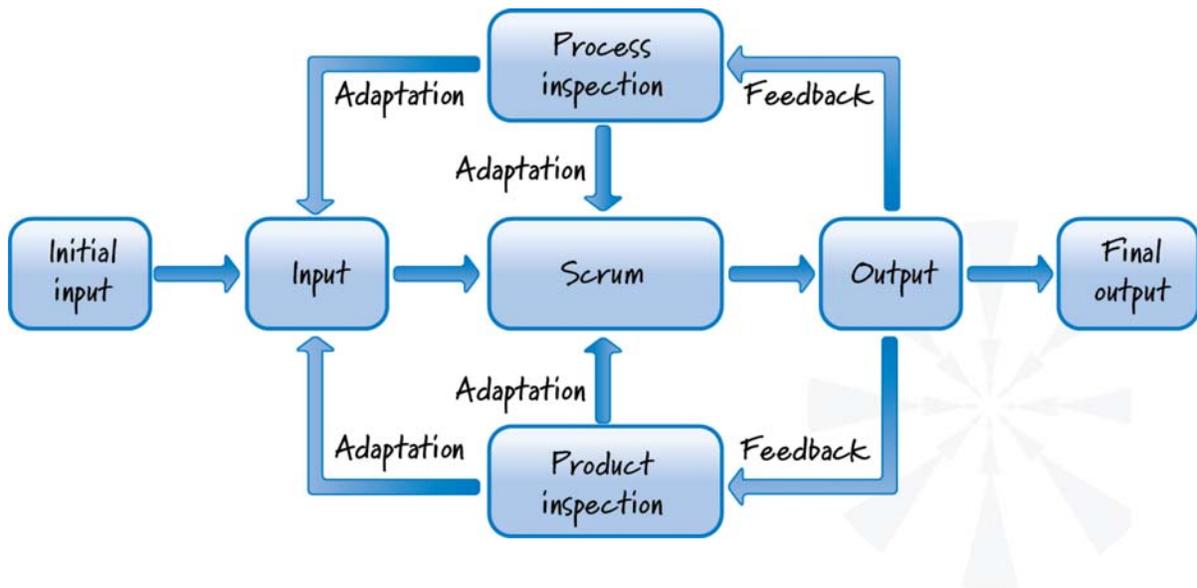
- In manufacturing our goal is:
  - Take a fixed set of requirements
  - Follow a sequential set of well-understood steps
  - Manufacture a finished product that is the same every time

Agile is Iterative & Incremental

Agile is an Empirical Process Model

Comparison of Plan-Driven and Agile Processes

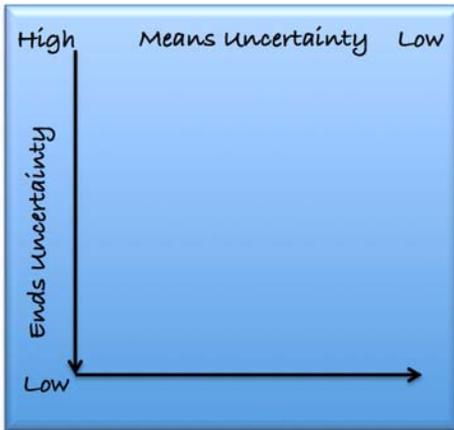| Dimension	Plan-driven	Agile
Degree of process definition	Well-defined set of sequential steps	Complex process that would defy a complete up-front definition
Randomness of output	Little or no output variability	Expect variability because we are not trying to build the same thing over and over
Amount of feedback used	Little and late	Frequent and early

Uncertainty Management

Waterfall

High   Means Uncertainty   Low

Ends Uncertainty

Low

Defined

Agile

High   Means Uncertainty   Low

Ends Uncertainty

Low
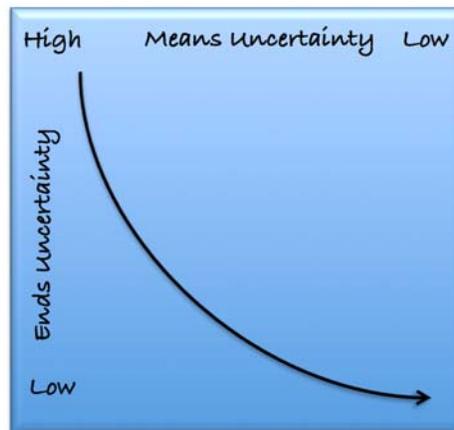
Empirical

Fragile, Robust, Antifragile
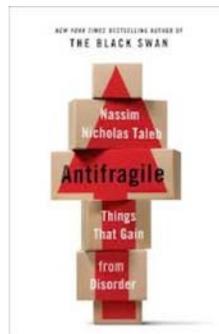
Waterfall

Agile

Fragile	Robust	Antifragile
Harmed by disorder	Resilient to disorder	Benefits from disorder

Asymmetric Payoffs Create Economic Value or Harm

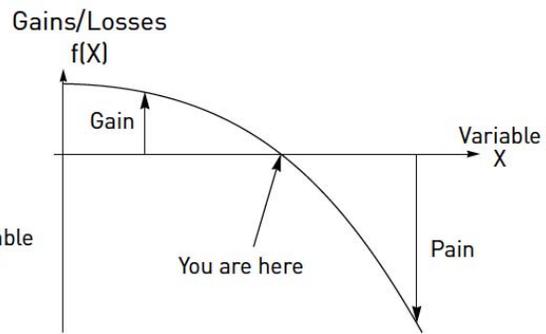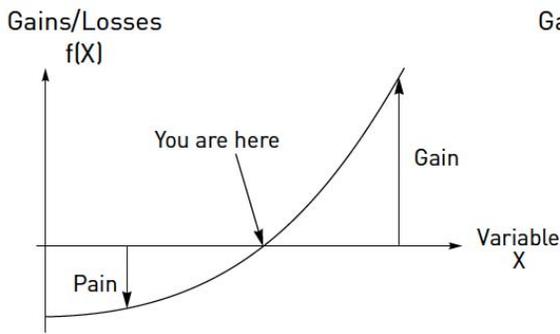| Positive asymmetric payoff (**antifragile**) anything that has more upside than downside from random events (variability) | Negative asymmetric payoff (**fragile**) anything that has more downside than upside from random events (variability) |

Source: Taleb, Nassim, Antifragile: Things That Gain from Disorder, Random House, 2012.

Getting Right Up-front

Danger zone! That's a lot of low-quality requirements specified when we don't have enough knowledge

Quantity of requirements produced at each point in time

Cumulative knowledge

Our cumulative product knowledge grows over time

Analysis   Design   Coding   Testing   Ops
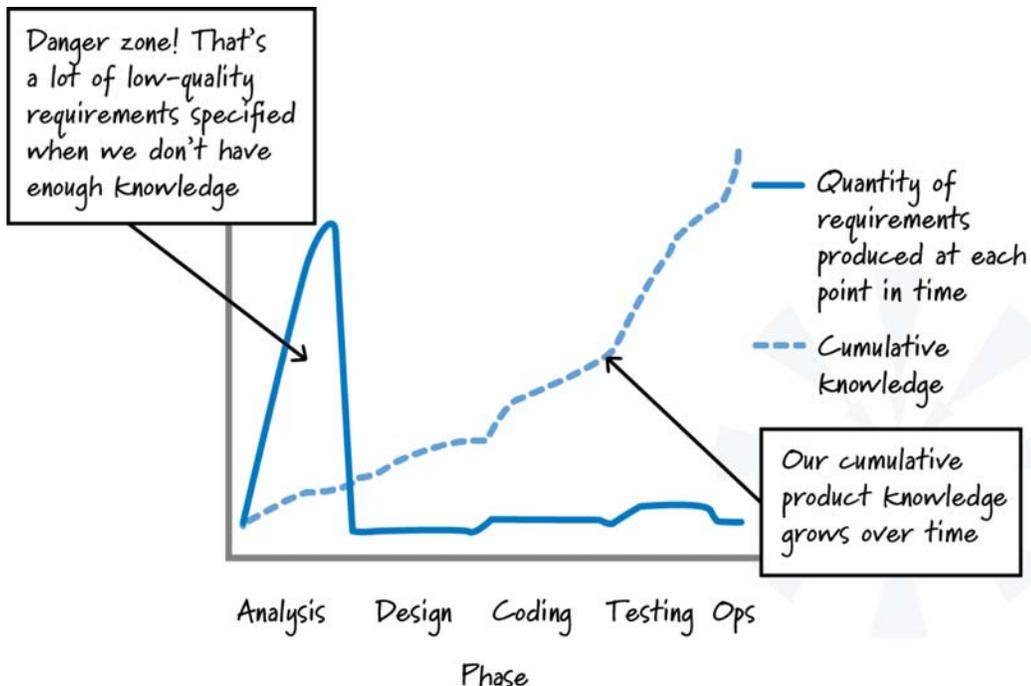
Phase

Decision Making

Make each decision in its proper phase

- Deliverable/Milestone
- Review/Approval

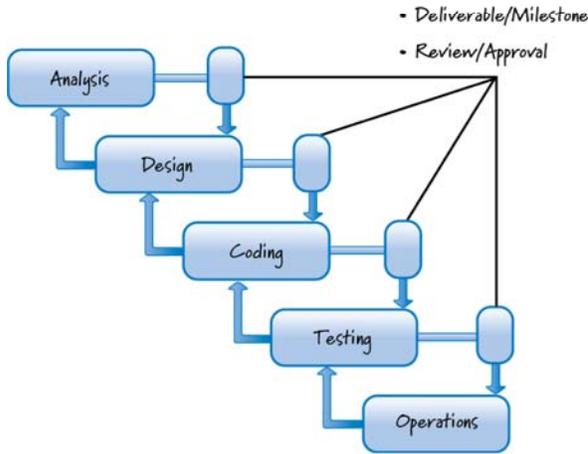Analysis → Design → Coding → Testing → Operations

Make important decisions at the last responsible moment

Cost

Last (economically) responsible moment

Cost of deferring

Cost of deciding

Time
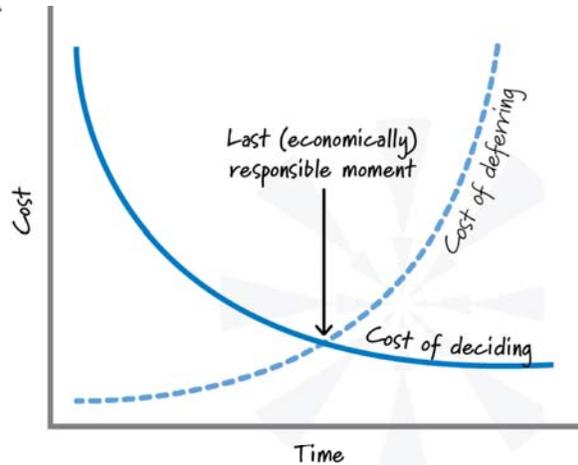
Exploration vs. Exploitation

Adaptive processes

Predictive processes

interleave small-scale

heavily focus on early

Exploration

always a tension between

Exploitation

in the presence of

in the presence of

Uncertainty

Uncertainty

involves

requires

Knowledge acquisition

Predicting

increases

does not increase
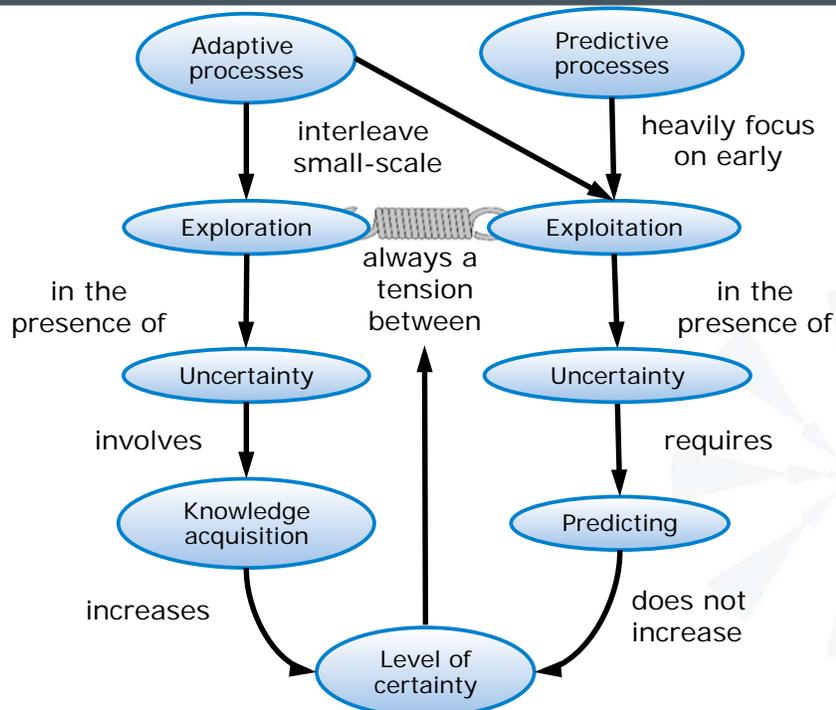
Level of certainty

Balance between Predictive and Adaptive

- Type of product
- Degree of end uncertainty
- Degree of means uncertainty
- Constraints on development
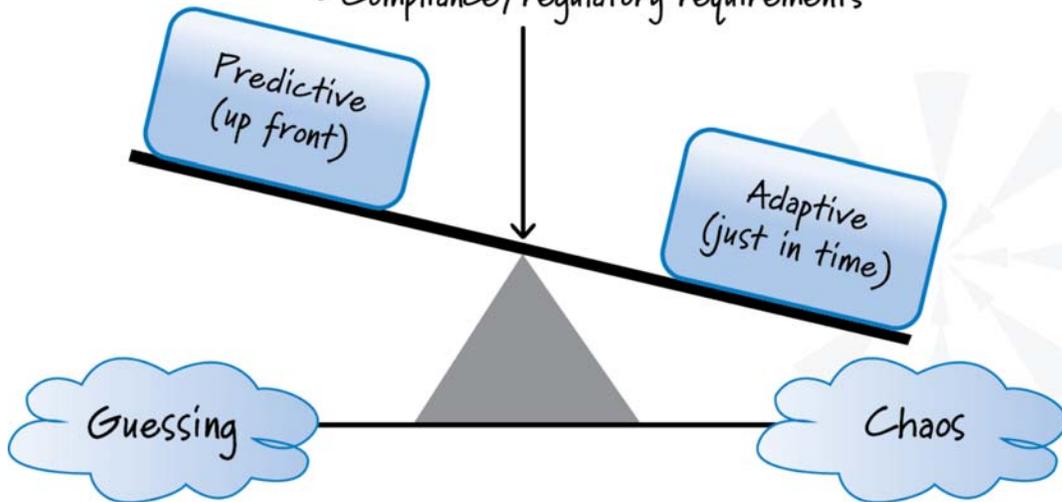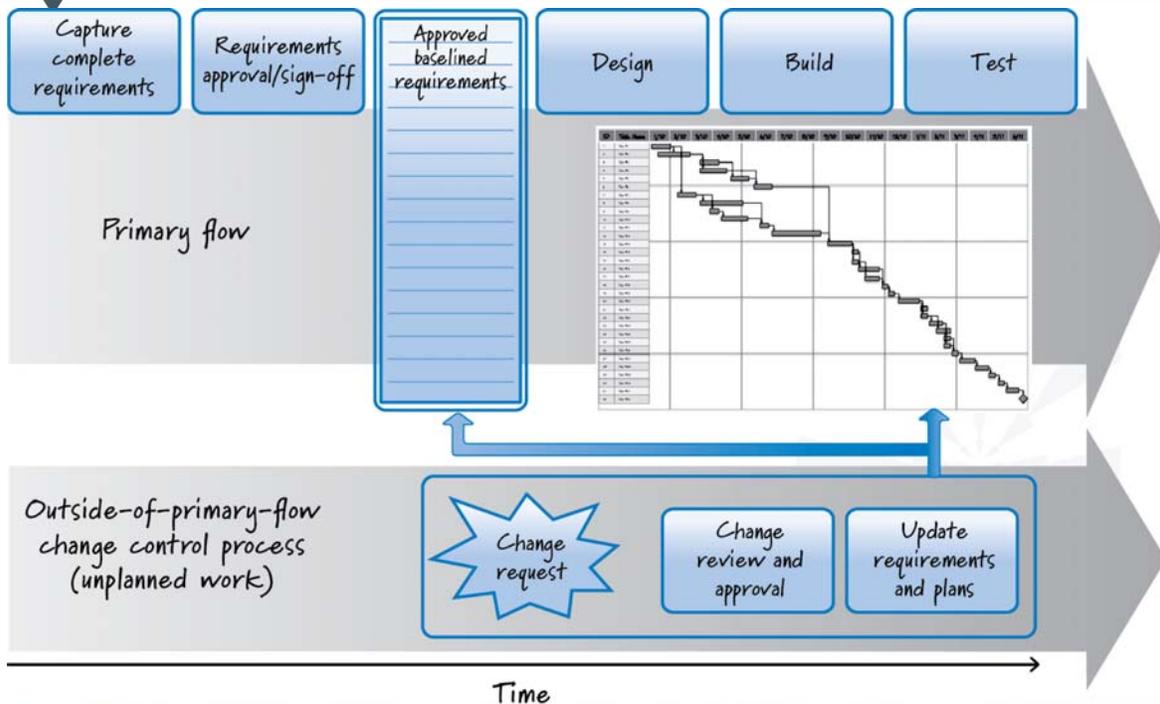- Compliance/regulatory requirements

Predictive (up front)

Adaptive (just in time)

Guessing

Chaos

Managing Change Risk During a Traditional Development Project

Capture complete requirements

Requirements approval/sign-off

Approved baselined requirements

Design

Build

Test

Primary flow

Outside-of-primary-flow change control process (unplanned work)

Change request

Change review and approval

Update requirements and plans

Time

Managing Change Risk Using Scrum
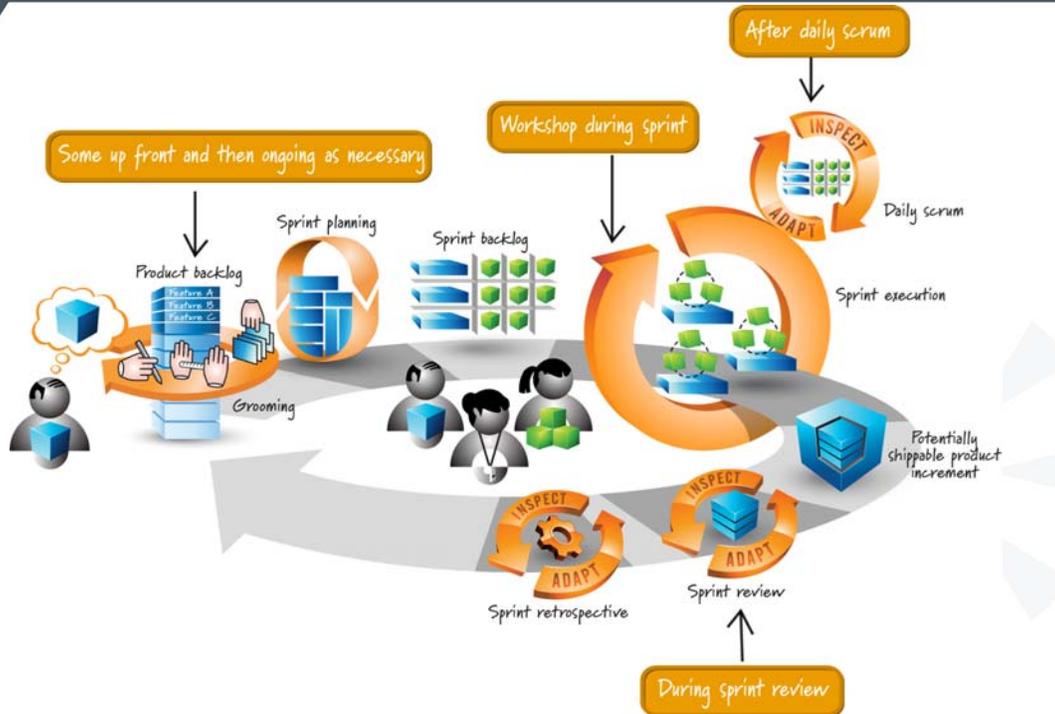
Leverage Multiple Concurrent Learning Loops
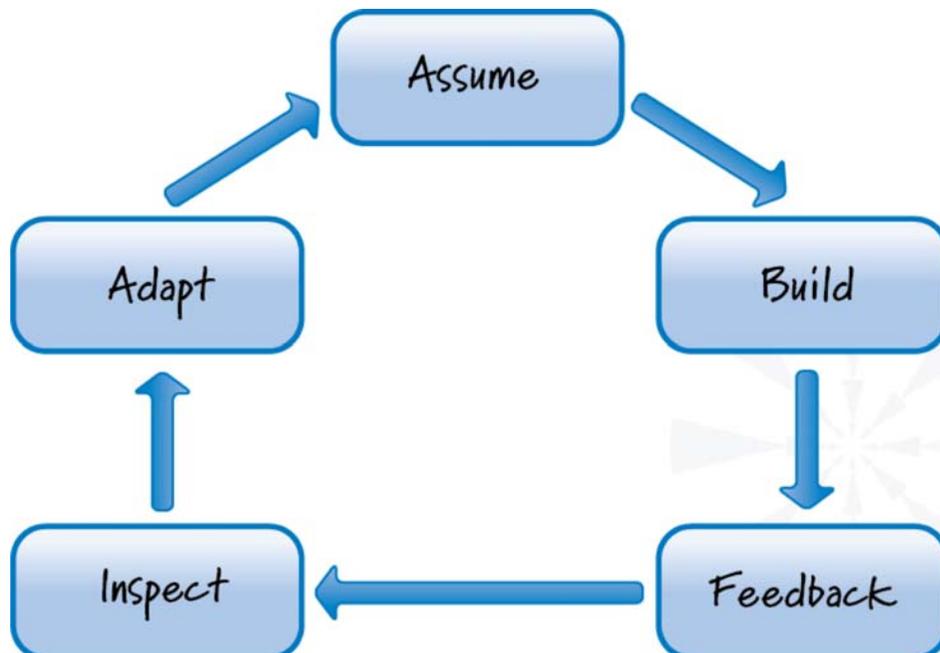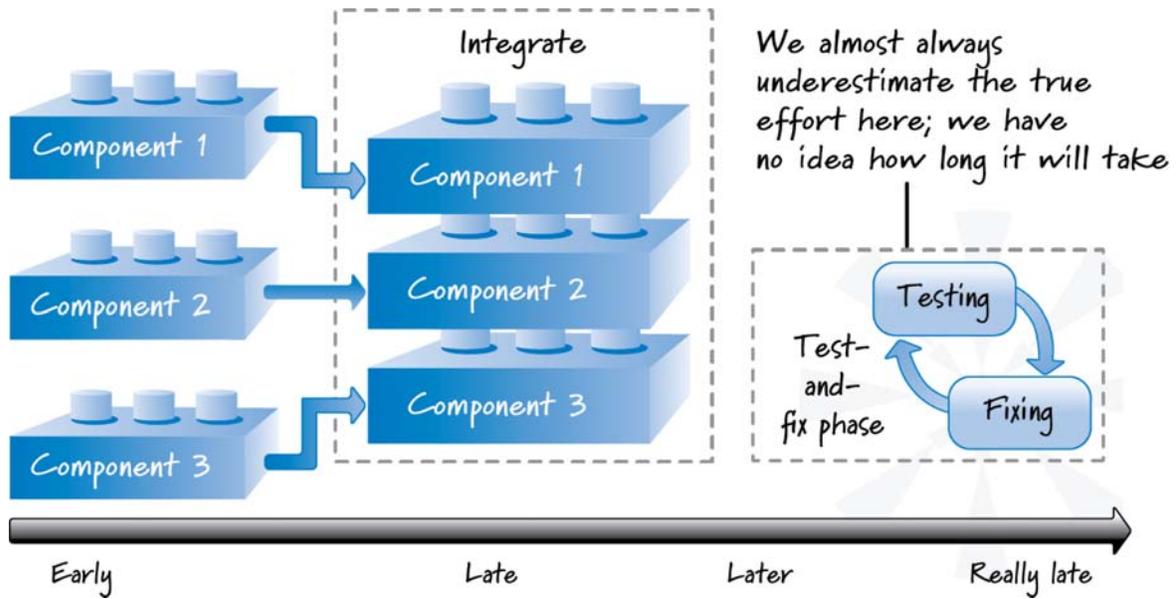
Organize Flow of Work for Fast Feedback

Integrate

We almost always underestimate the true effort here; we have no idea how long it will take

Component 1 → Component 1

Component 2 → Component 2

Component 3 → Component 3

Test-and-fix phase

Testing → Fixing

Early — Late — Later — Really late

Large vs. Small Batch Sizes

Large batches (100%) All before any

Small batches

- Deliverable/Milestone
- Review/Approval

Analysis → Design → Coding → Testing → Operations

Sprint 1: Analyze, Design, Test, Integrate, Build
Sprint 2: Analyze, Design, Test, Integrate, Build
Sprint 3: Analyze, Design, Test, Integrate, Build

Adapt / Feedback

Increment 1 — Increment 2 — Increment 3
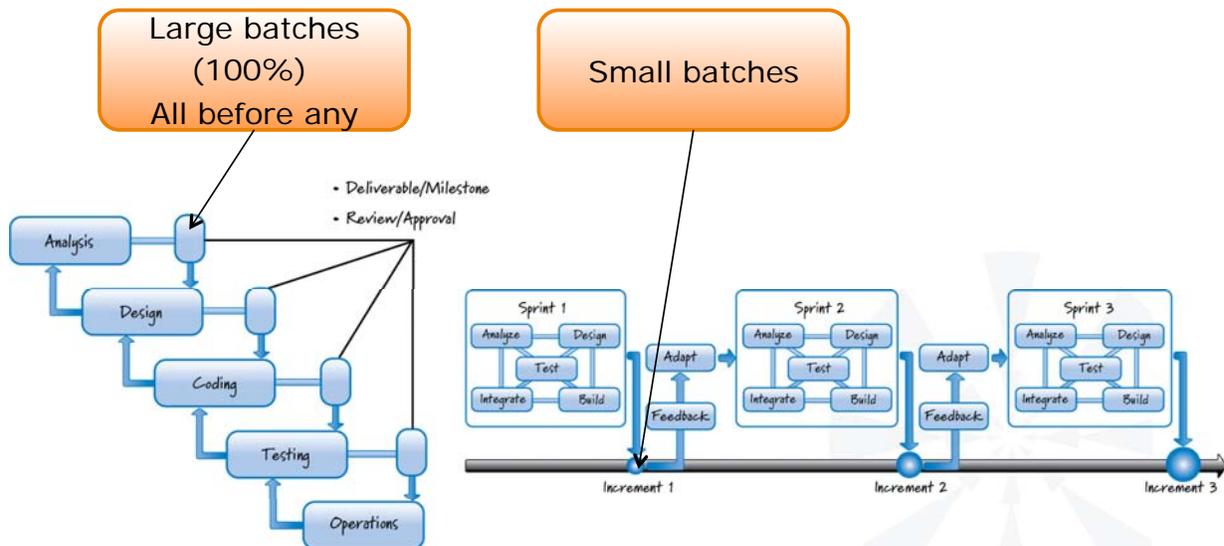
Benefits of Small Batch Sizes in Product Development

Reduced cycle time

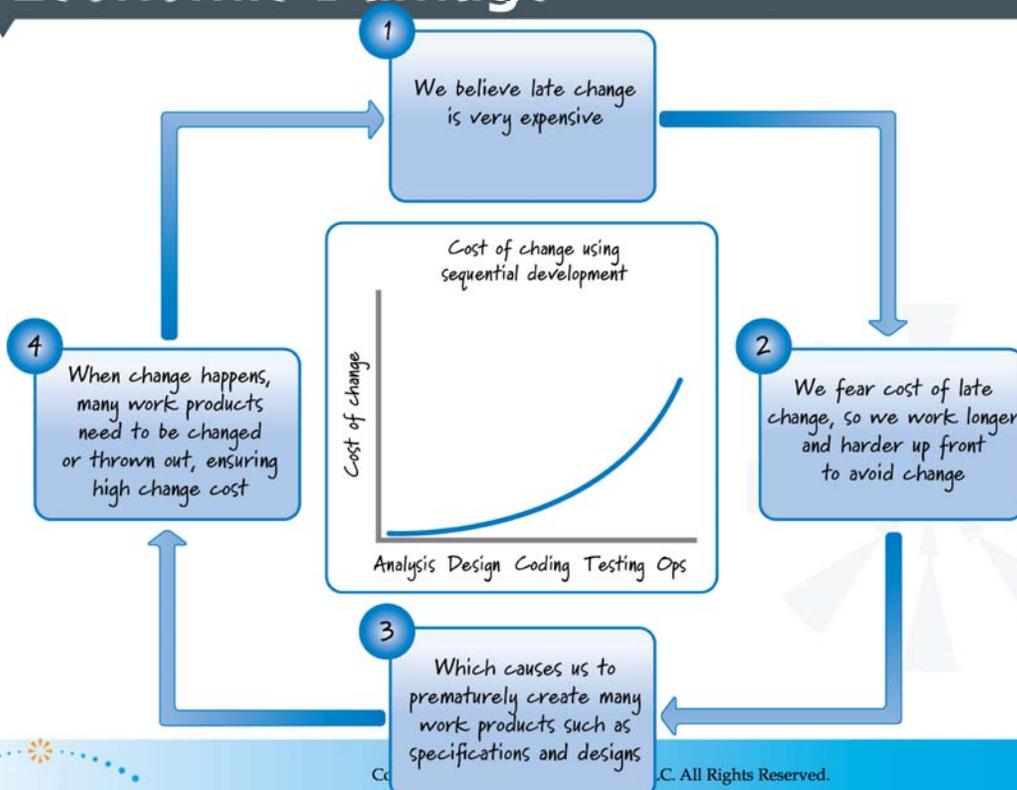Reduced flow variability

Accelerated feedback

Lower risk of failure

Reduced overhead

Increased motivation & urgency

Reduced cost and schedule growth

Poorly Managed Inventory Causes Economic Damage

1 We believe late change is very expensive

2 We fear cost of late change, so we work longer and harder up front to avoid change

3 Which causes us to prematurely create many work products such as specifications and designs

4 When change happens, many work products need to be changed or thrown out, ensuring high change cost

Cost of change using sequential development

Cost of change
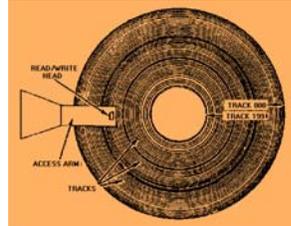
Analysis Design Coding Testing Ops

Recognize Inventory (WIP) and Manage it for Good Flow

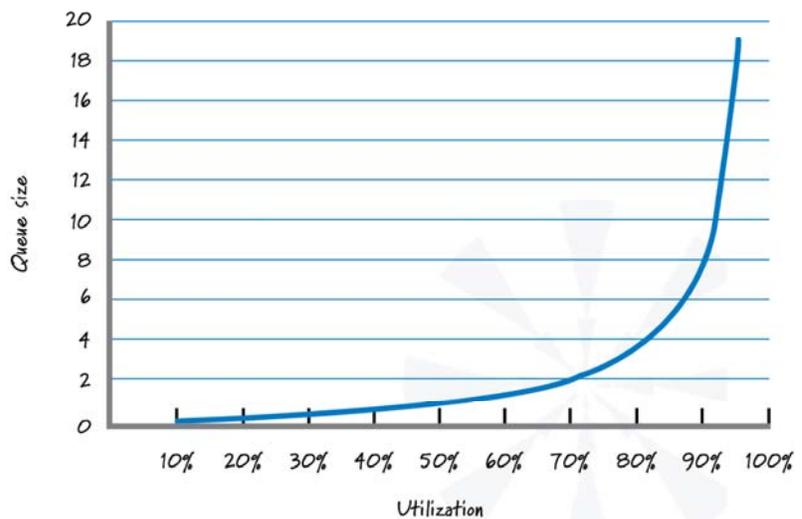Manufacturing inventory is both physically and financially visible

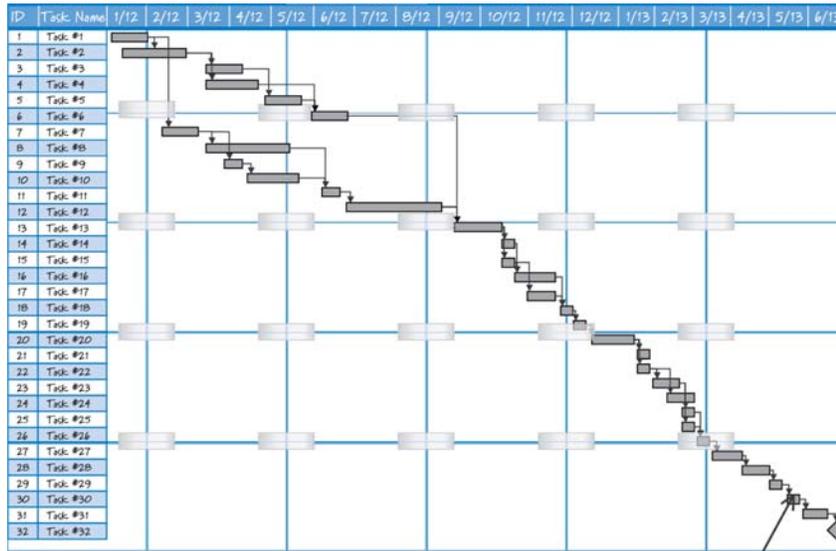Product-development inventory are knowledge assets that aren't visible in the same way as physical parts

Focus on Idle Work Not Idle Workers

Watch the Baton Not the Runners
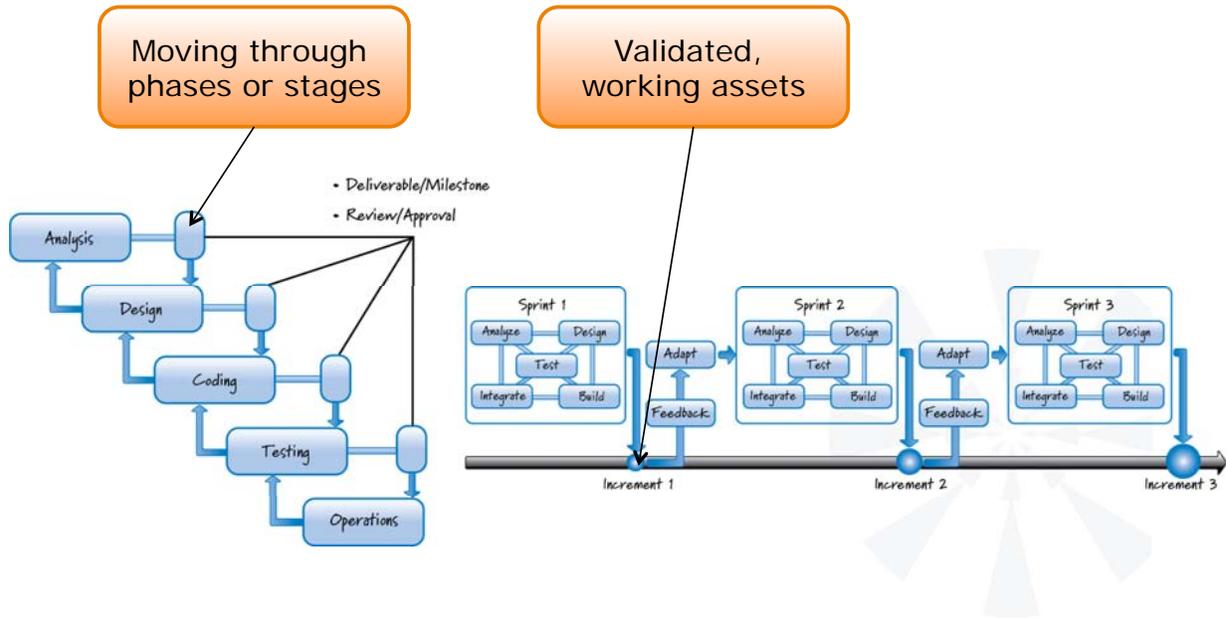
Waterfall is Conformance to Plan

George 18 months from today

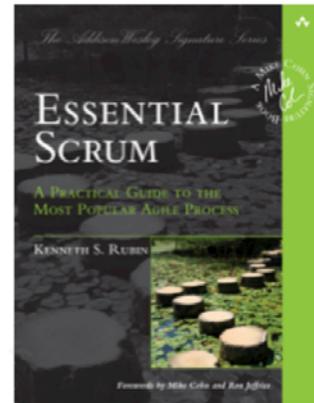# Agile is Replanning and Adapting

This doesn't look the same at all!

Moving through phases or stages

Validated, working assets

- Deliverable/Milestone
- Review/Approval

Analysis

Design

Coding

Testing

Operations

Sprint 1
Analyze Design
Test
Integrate Build

Adapt

Feedback

Sprint 2
Analyze Design
Test
Integrate Build

Adapt

Feedback

Sprint 3
Analyze Design
Test
Integrate Build

Increment 1

Increment 2

Increment 3

Summary: Principles are the Foundational Why

Sprint planning

Sprint backlog

INSPECT
ADAPT
Daily scrum

Product backlog

Grooming

Sprint execution

Potentially shippable product increment

INSPECT
ADAPT

INSPECT
ADAPT

Sprint review

Sprint retrospective

Validated learning

Work in Process (WIP)

Prediction & adaptation

Variability & uncertainty

Lifecycle Profits

Performance

Progress

✳ Visual AGILExicon®

✳ Slides in this presentation contain items from the Visual AGILExicon®, which is a trademark of Innolution, LLC and Kenneth S. Rubin.

✳ The Visual AGILExicon is used and described in the book: ***Essential Scrum: A Practical Guide to the Most Popular Agile Process***.

✳ You can learn more about the Visual AGILExicon and permitted uses at: http://innolution.com/resources/val-home-page

Visual AGILExicon®

✳ www.essentialscrum.com

Contact Info for Ken Rubin

iNNolutioN
agile innovative solutions

Email:	krubin@innolution.com
Website:	www.innolution.com
Phone:	(303) 827-3333
LinkedIn:	www.linkedin.com/in/kennethrubin
Twitter:	www.twitter.com/krubinagile
Facebook:	www.facebook.com/InnolutionLLC
Google+	plus.google.com/+KennyRubin1/
Essential Scrum: A Practical Guide to the Most Popular Agile Process	www.essentialscrum.com