Why Agile?

The Economics, Psychology, and Science of Agile's Success

Matthew Renze

Agile Iowa

Fall 2013

Purpose

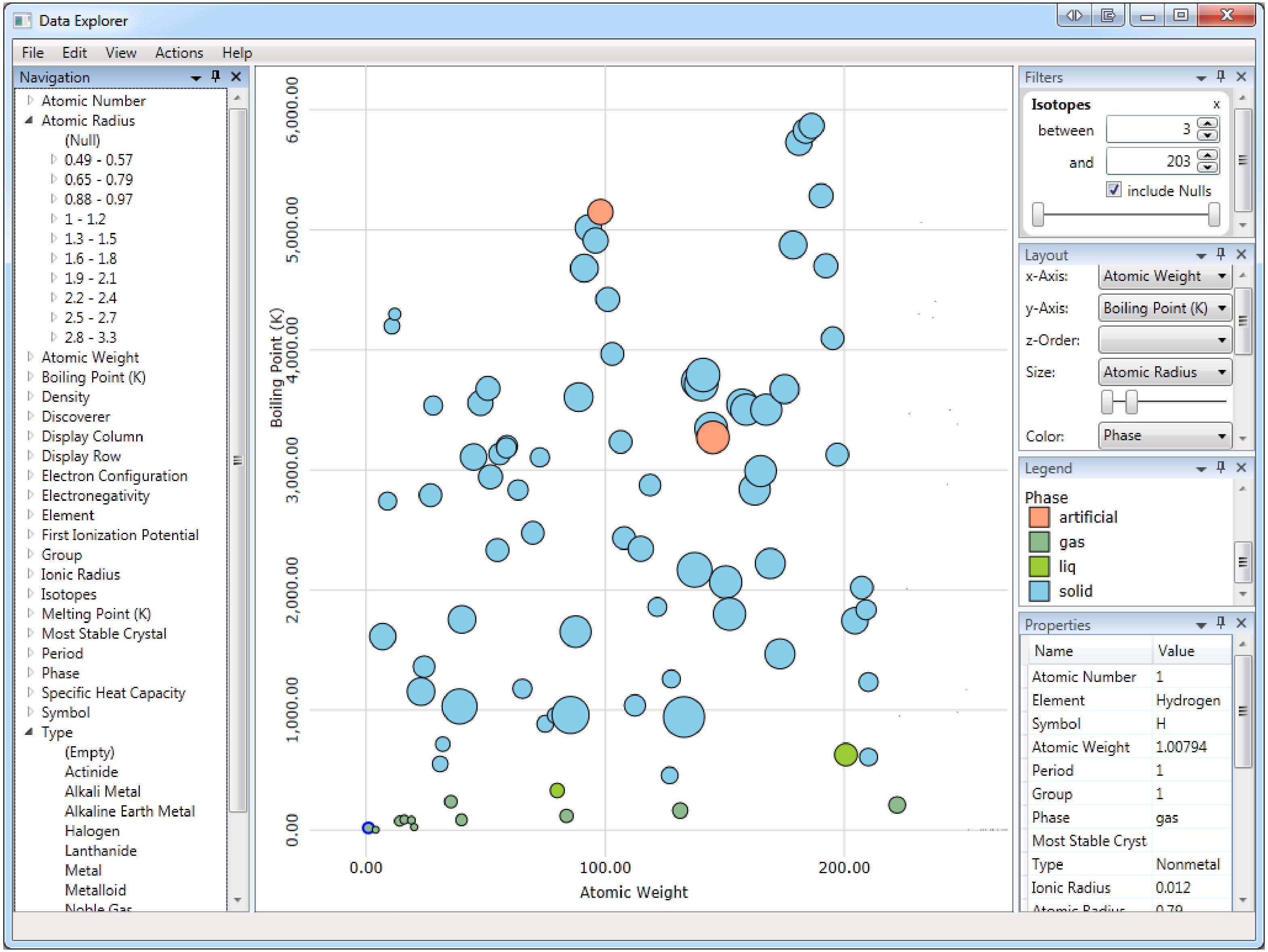
- Explain why Agile practices are so successful
- Insights from:
    - Economics
    - Psychology
    - Science
- Top 7 most important ideas
- Ideas that are not typically covered

Purpose

- The presentation will not provide:
  - Introduction to Agile
  - Basic or obvious explanations for Agile’s success
  - Comprehensive theory of the success of Agile
- I will however provide a quick review of Agile

About Me

- Independent software consultant
- 14 years of Agile software development experience
- Data-driven desktop, server, and web apps
    - Web-based GIS data warehouse
    - Energy data ETL application
    - Global data management system
    - Intelligent lighting control systems
    - Open source data explorer



Isotopes

between 3

and 203

☑ include Nulls

x-Axis: Atomic Weight

y-Axis: Boiling Point (K)

z-Order:

Size: Atomic Radius

Color: Phase

Name	Value
Atomic Number	1
Element	Hydrogen
Symbol	H
Atomic Weight	1.00794
Period	1
Group	1
Phase	gas
Most Stable Cryst	
Type	Nonmetal
Ionic Radius	0.012
Atomic Radius	0.79

www.data-explorer.com

About Me

- Education
  - BS in Computer Science
  - BA in Philosophy
    - Minor in Economics
    - Focus on Artificial Intelligence and Machine Learning
  - AS in MIS
  - AS in Business Administration
- Training
  - Certified Scrum Master

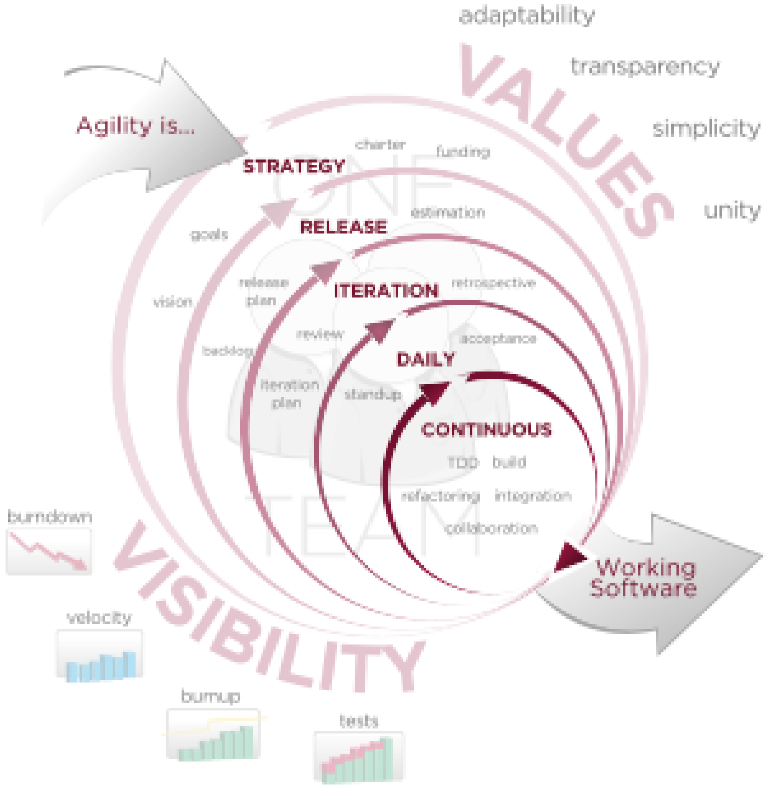
Overview

1. The World after Midnight
2. Inverted Constraints
3. Prioritizing Value
4. Embracing Change
5. Self-Organization
6. Efficient Communication
7. Feedback

A Brief Review of Agile

What is Agile?

- Started with the Agile Manifesto
  - 4 value propositions
  - 12 principles
- Common set of practices across several methodologies



Source: Wikipedia

What is Agile?

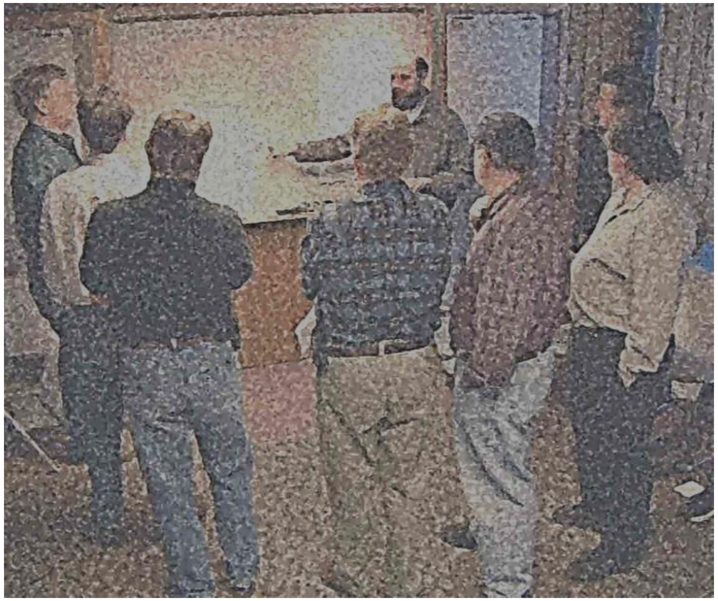
Agile is *not*:

- A software development methodology itself
- A silver bullet for all your software woes

Source: http://www.best-story.net/userfiles/silver-bullets.jpg

Agile Values

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



Source: http://agilemanifesto.org/

12 Principles of Agile

1. Continuous delivery of value
2. Embrace changing requirements
3. Frequent deployment
4. Customer collaboration
5. Motivated individuals
6. Face-to-face conversation

12 Principles of Agile

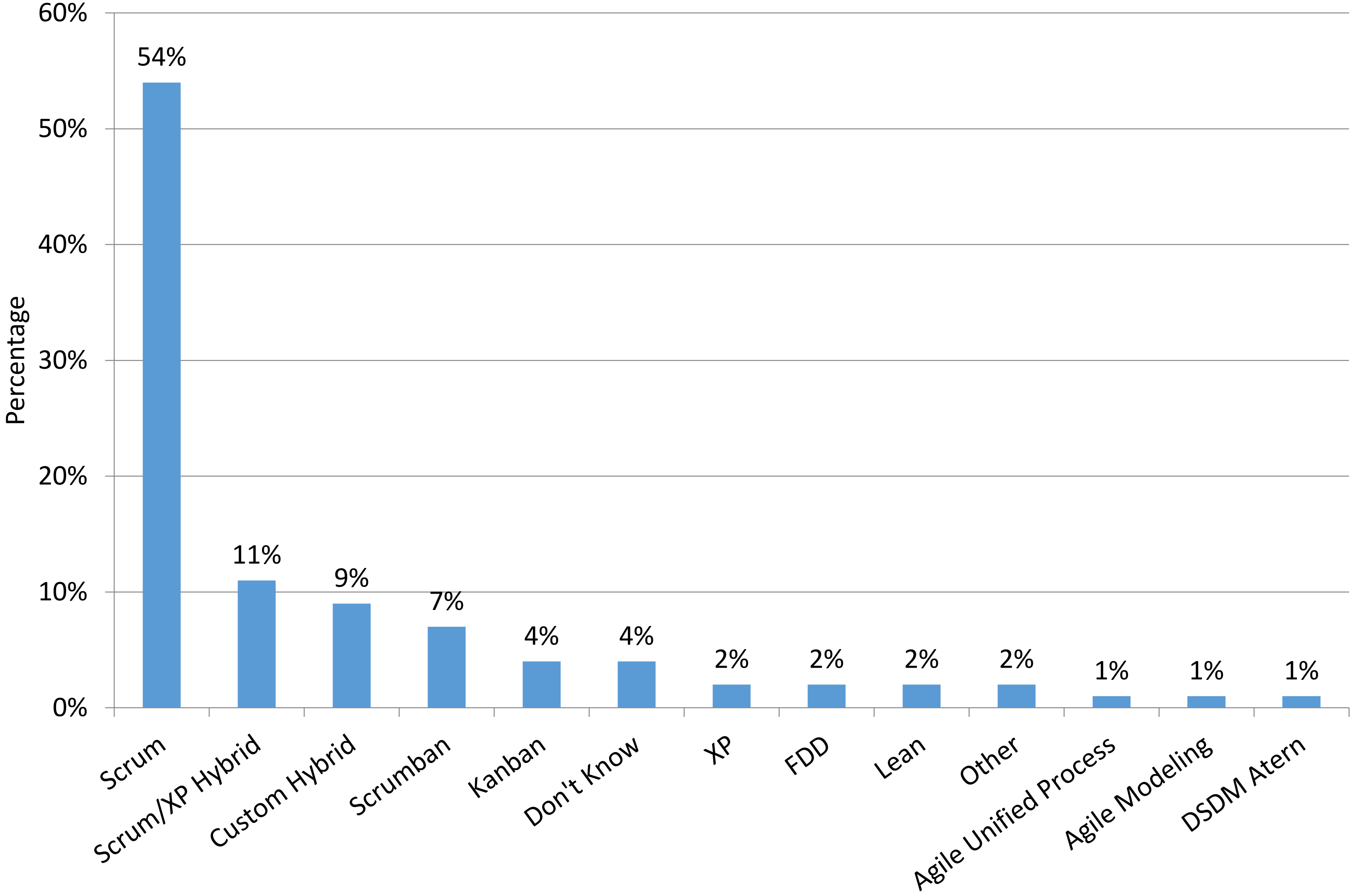
7. Working software as measure of progress
8. Sustainable development
9. Technical excellence
10. Simplicity
11. Self-organization
12. Continuous improvement

Agile Methodologies

- Scrum
- XP
- Kanban
- Lean
- And many more…

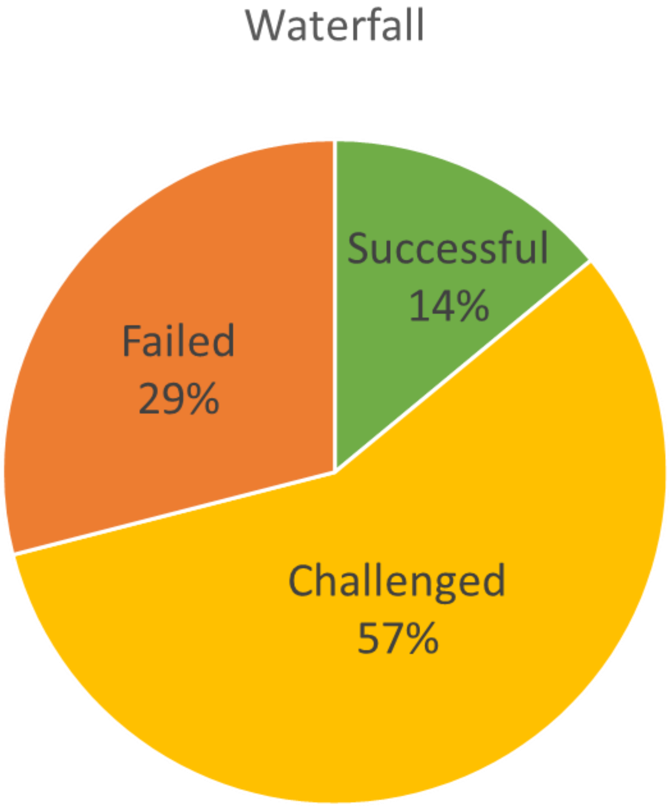
Source: http://parkertoddloesch.files.wordpress.com/2011/09/umbrella.jpg

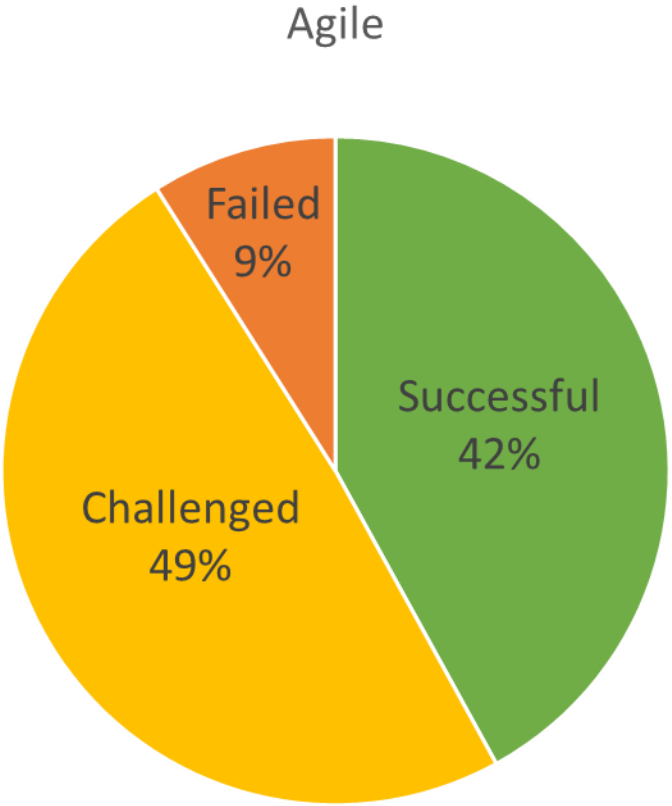
Agile Methods Used



Original Source: Version One - State of Agile Survey 2012

Is Agile More Successful?





Original Source: The Standish Group, The CHAOS Report 2012

1. The World after Midnight

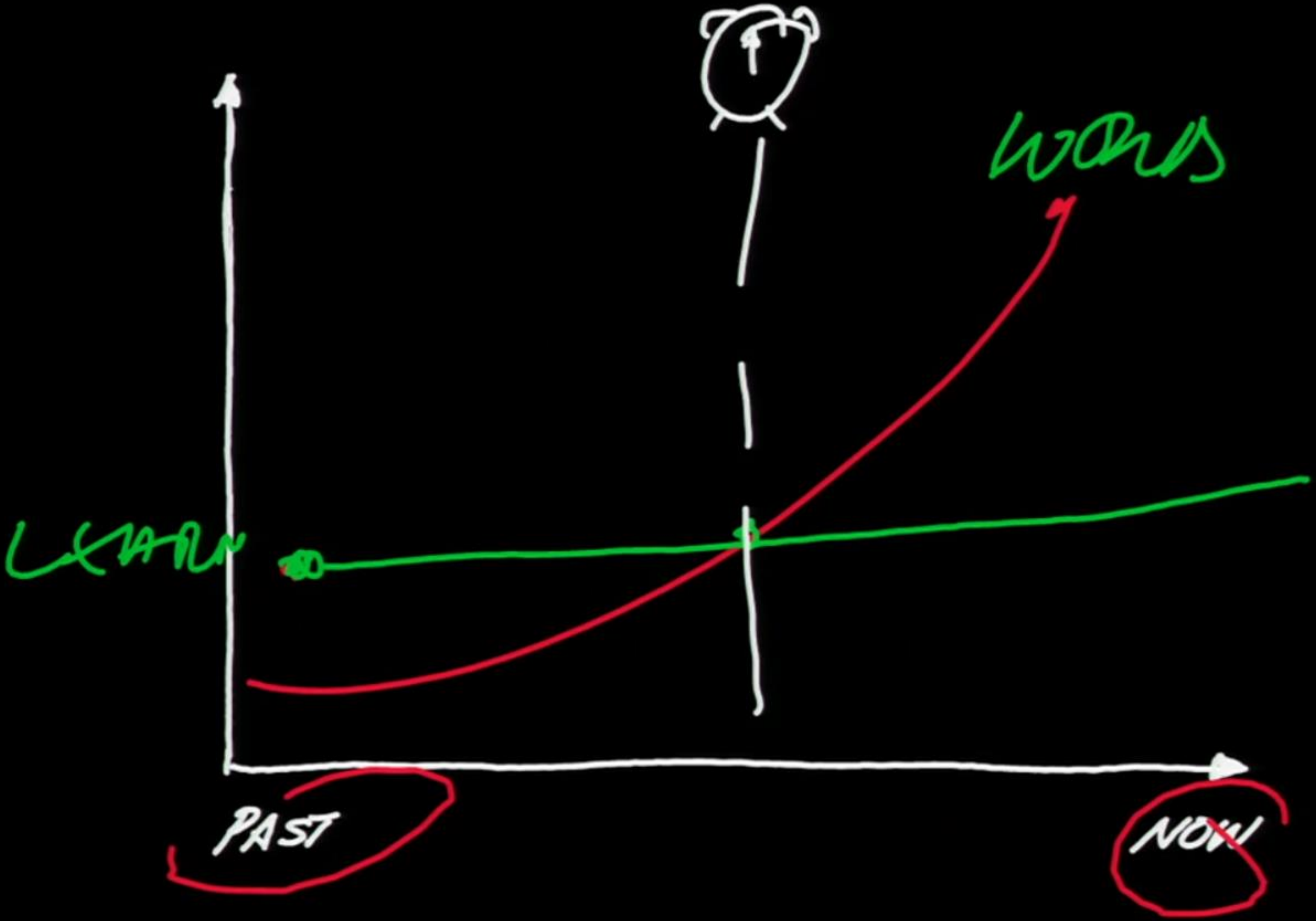
Source: www.ted.com

The World After Midnight

About fifteen years ago all the ‘Rules’ about how to run a business, organization, or government successfully, were changed or deleted and a completely new set of ‘Rules’ has been in operation ever since, which means that ***we keep acting rationally in response to a world we recognize and understand… but which no longer exists!***

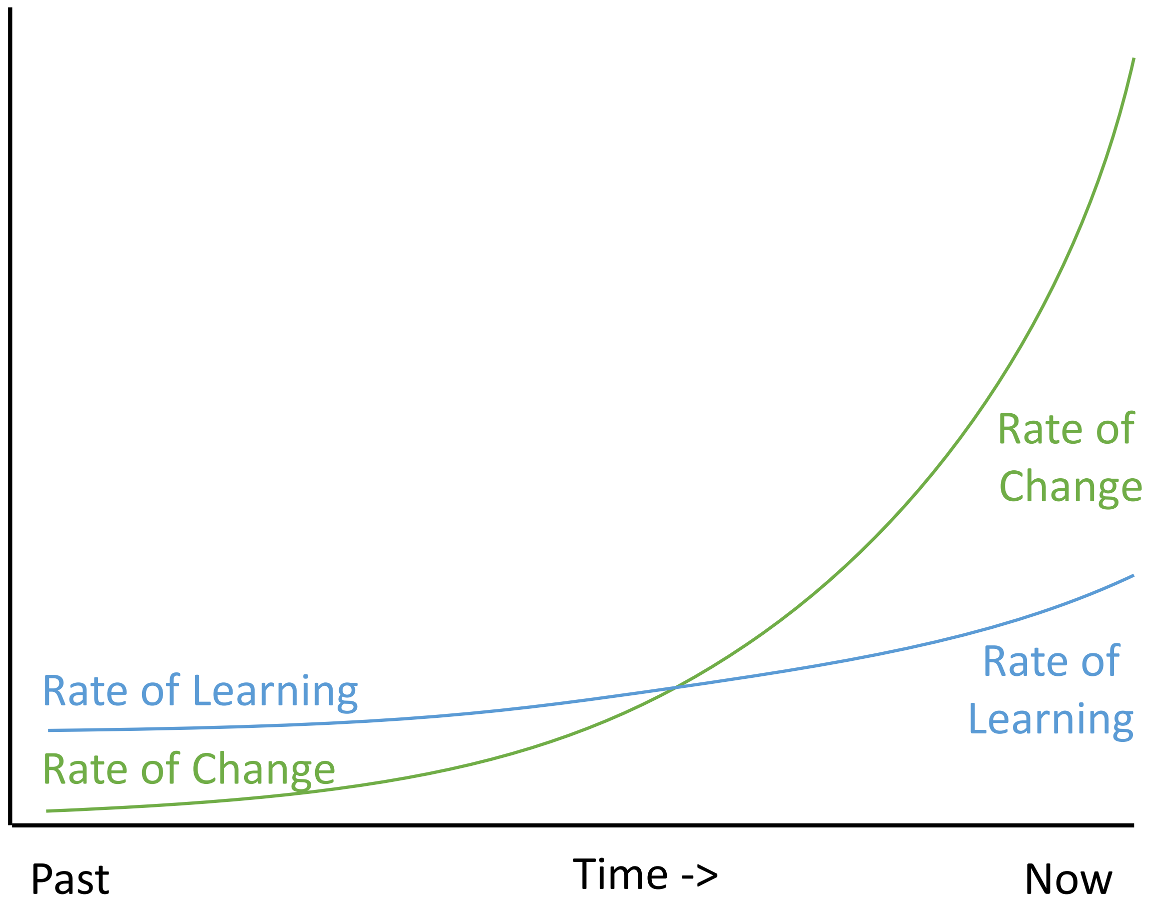
- Eddie Obeng

The World After Midnight



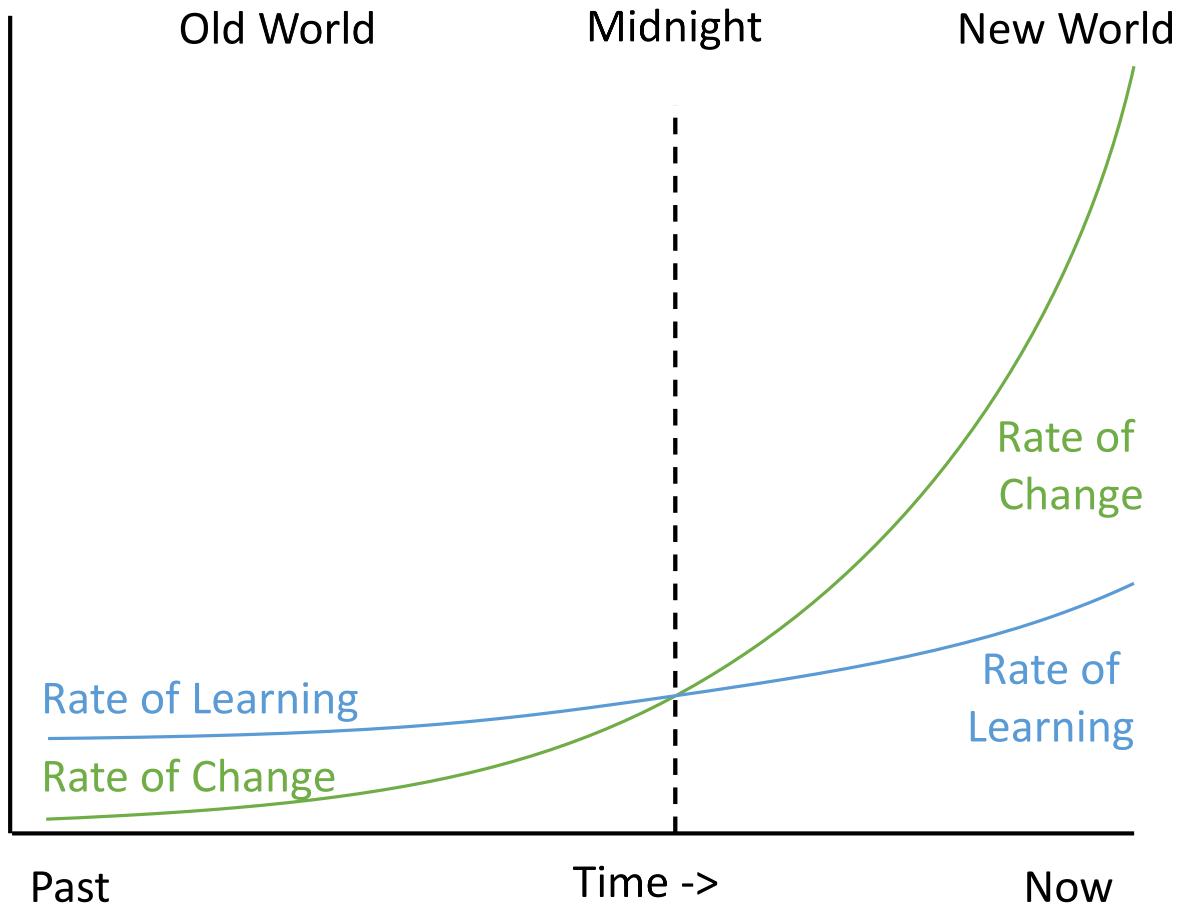
Source: http://www.ted.com/talks/eddie_obeng_smart_failure_for_a_fast_changing_world.html

The World after Midnight



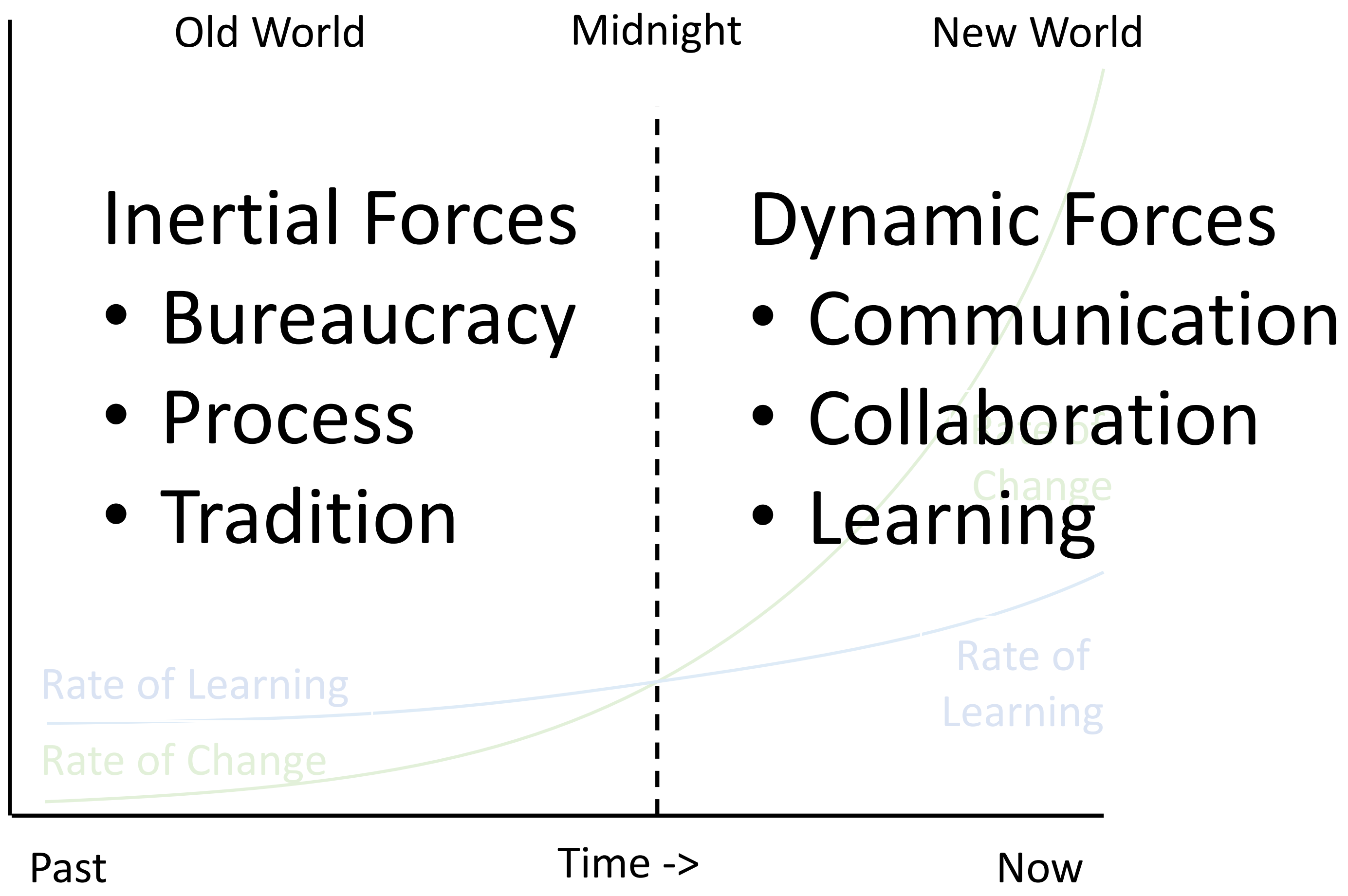
Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg

The World after Midnight



Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg

The World after Midnight



Inertial Forces

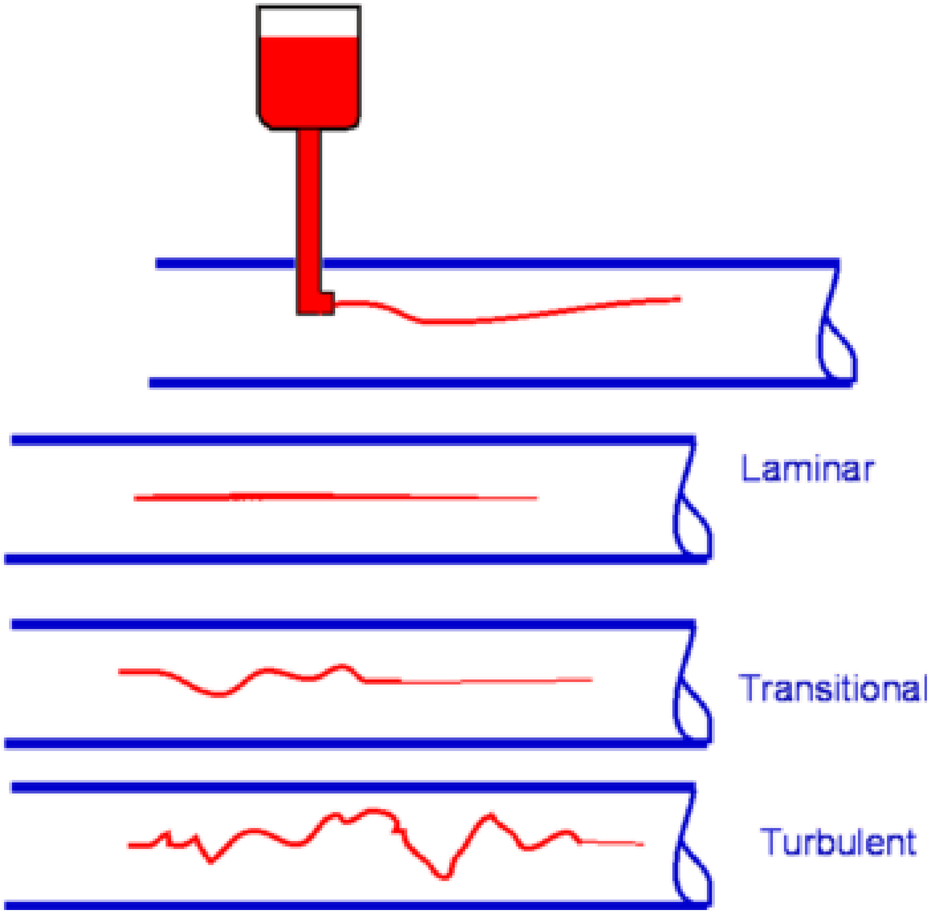
- Bureaucracy
- Process
- Tradition

Dynamic Forces

- Communication
- Collaboration
- Learning

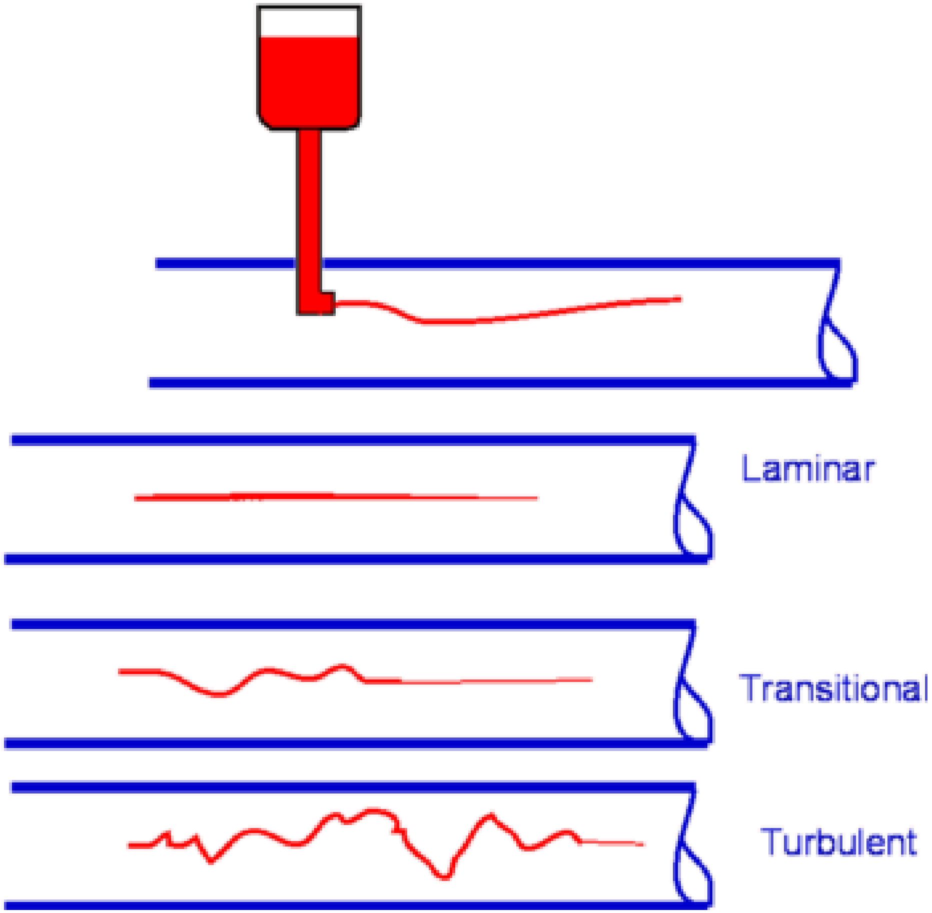
Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg

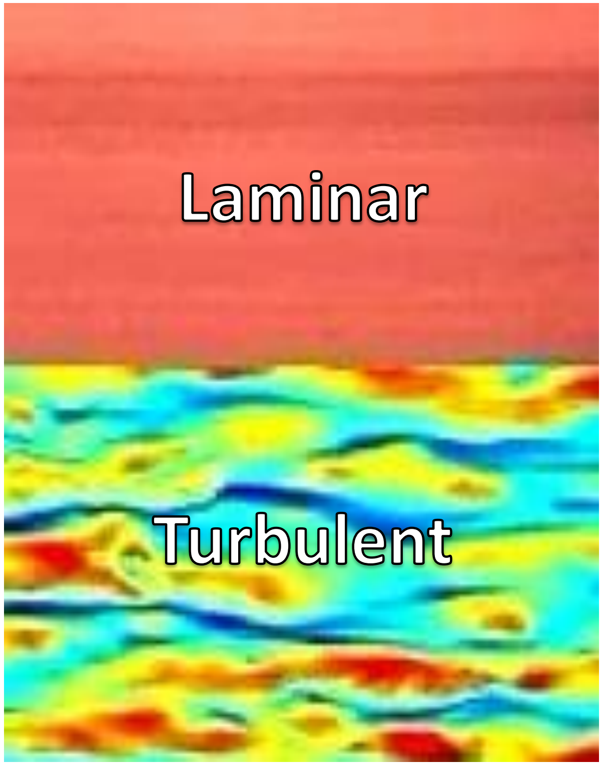
Laminar Flow vs. Turbulent Flow



Source: http://www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/fprops/pipeflow/node8.html

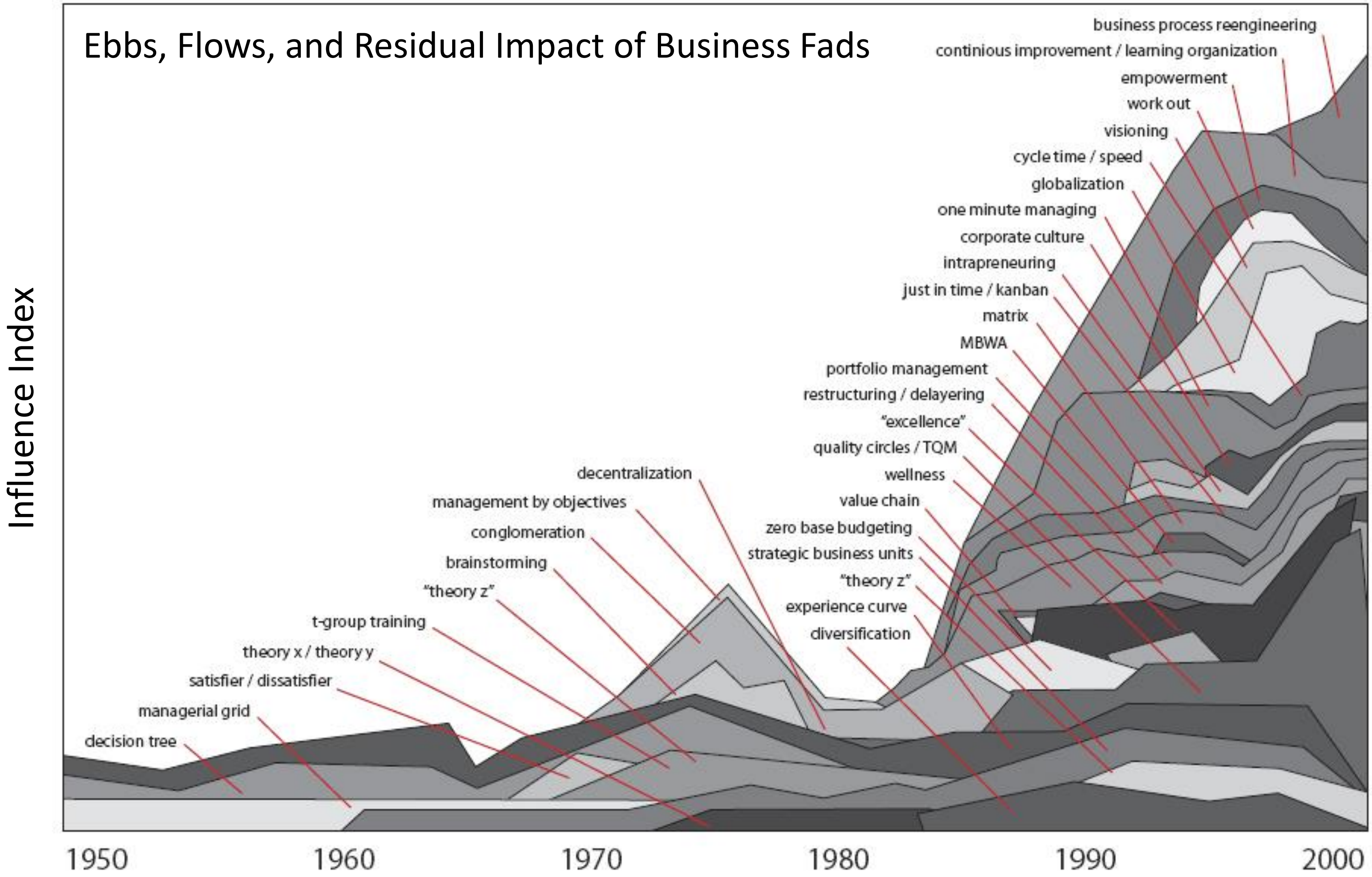
Laminar Flow vs. Turbulent Flow





Source: http://www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/fprops/pipeflow/node8.html

Ebbs, Flows, and Residual Impact of Business Fads



Source: 'The Ebbs, Flows and Residual Impact of Business Fads 1950 – 1995' by R. Pascale

Why is this important?

Problem

- World has changed
- Markets change rapidly
- Requirements change rapidly
- High degree of uncertainty

Solution

- Adapt to new physics
- Faster time-to-market
- Better response to change
- Continuous and rapid feedback, validated learning, and smart failure

Agile is very well suited to operate in the physics of this new world!

2. Inverted Constraints

Four Levers of Software Development

- Levers
  - Scope
  - Resources
  - Schedule
  - Quality
- Goal
  - Working software
  - Max value
  - Min cost

Source: http://farm6.staticflickr.com/5300/5521479079_36815225e4_z.jpg

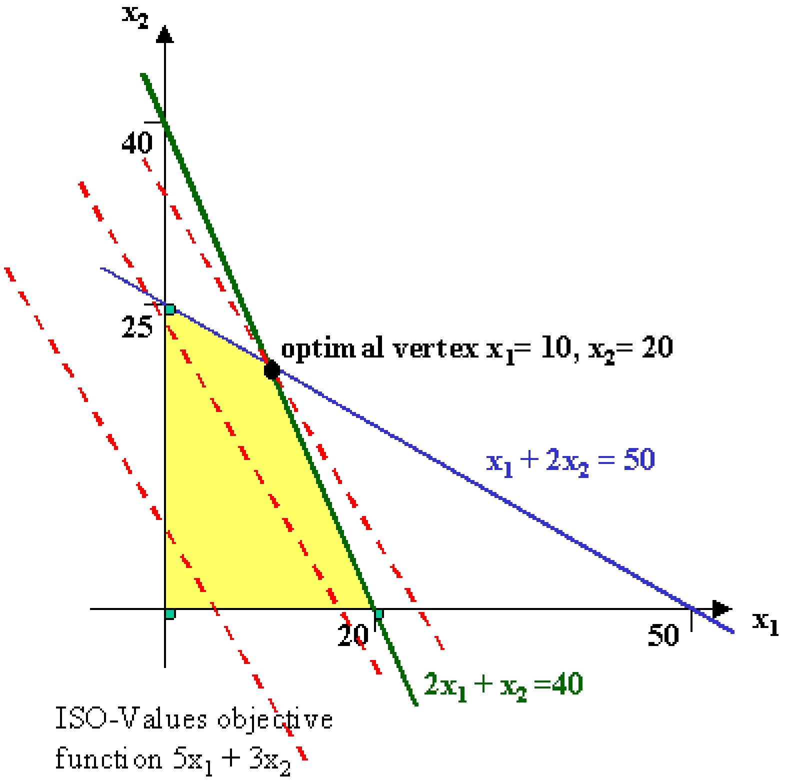
Constraints

- Restriction on a degree of freedom
- Prevent the system from achieving its goal
- Examples
  - Time
  - Money
  - Talent

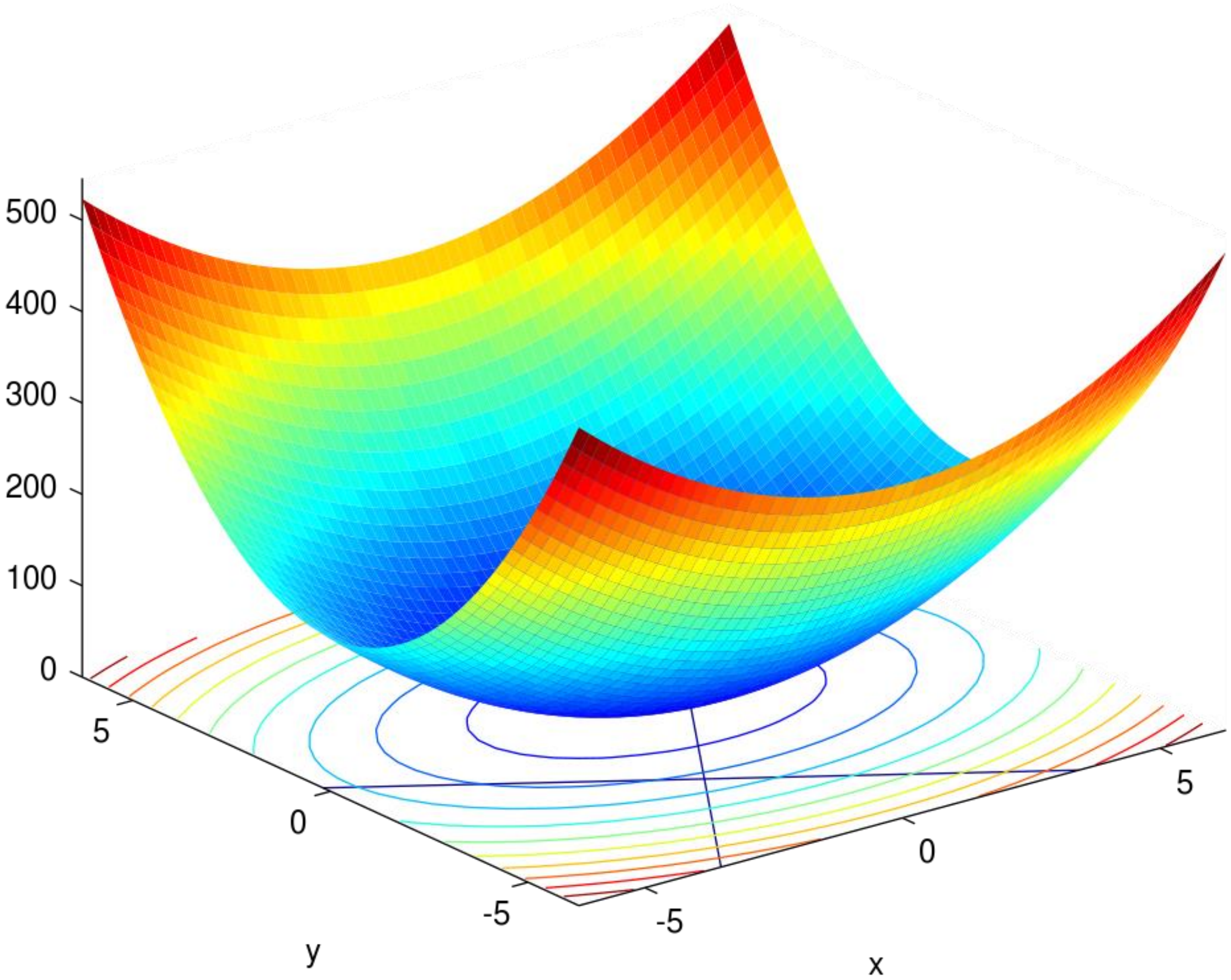
Source: http://www.myspaceantics.com/image-myspace-graphic/funny-pictures/outfielder-wall-collision.jpg.html

Constrained Optimization

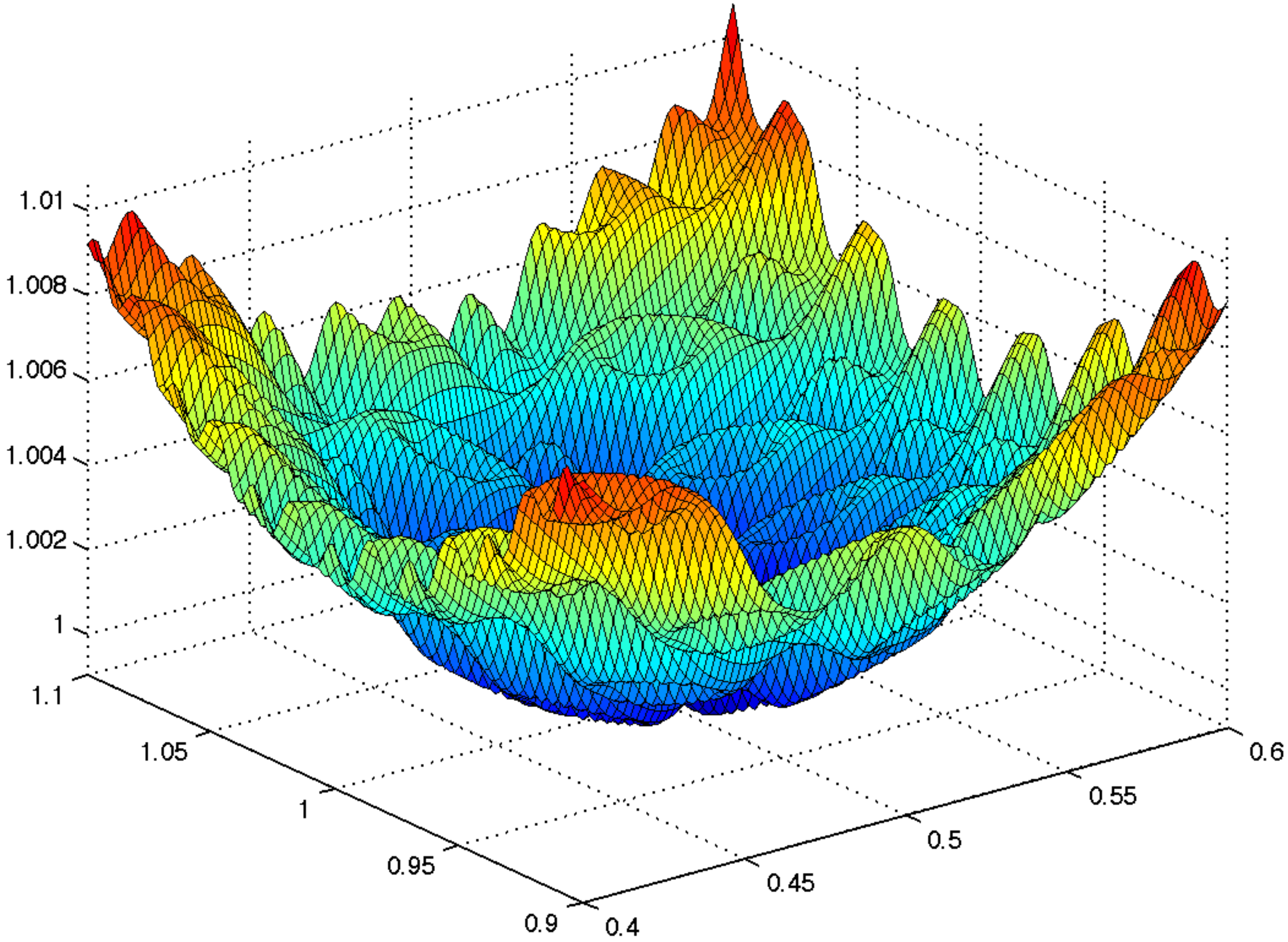
- Linear Programming
  - Dimensions = levers
  - Solid lines = constraints
  - Dash lines = value
  - Shaded = feasible
  - Solution = optimum



Source: http://home.ubalt.edu/ntsbarsh/business-stat/opre/partVIII.htm

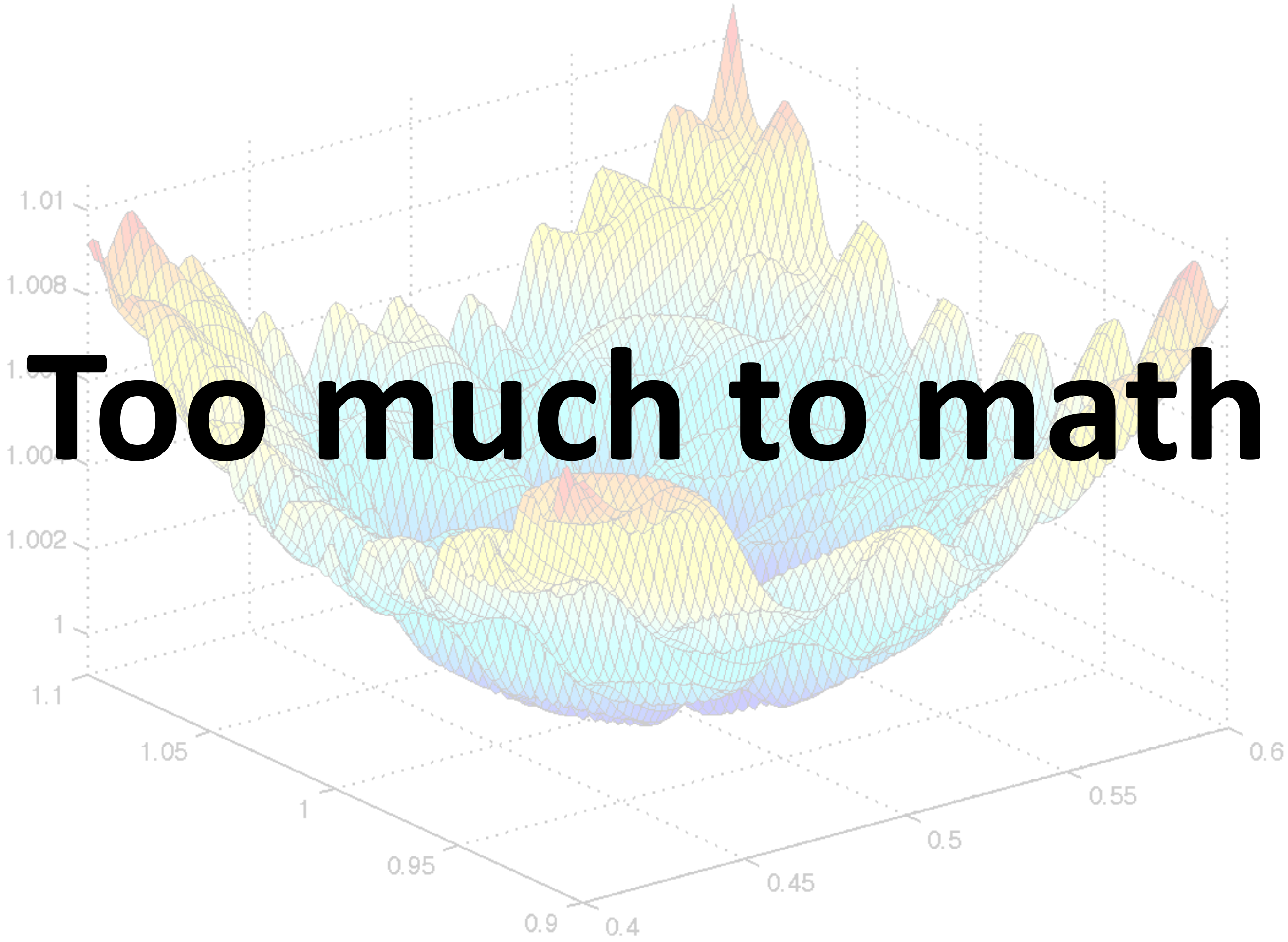


Source: http://en.wikipedia.org/wiki/File:As_wiki_lgm_parab.svg

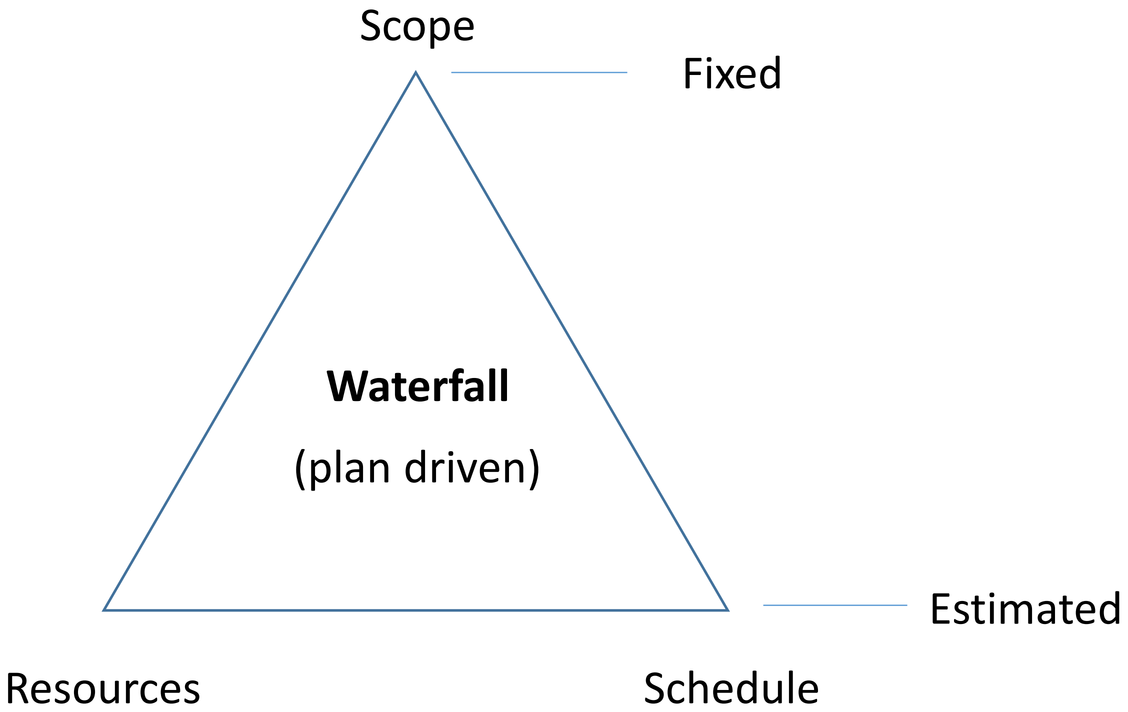


Source: http://www.oceanmedia.in/ebooks/wp-content/uploads/2012/09/9788132336426.jpg

Too much to math!



Waterfall Constraints



Scope

Fixed

Waterfall

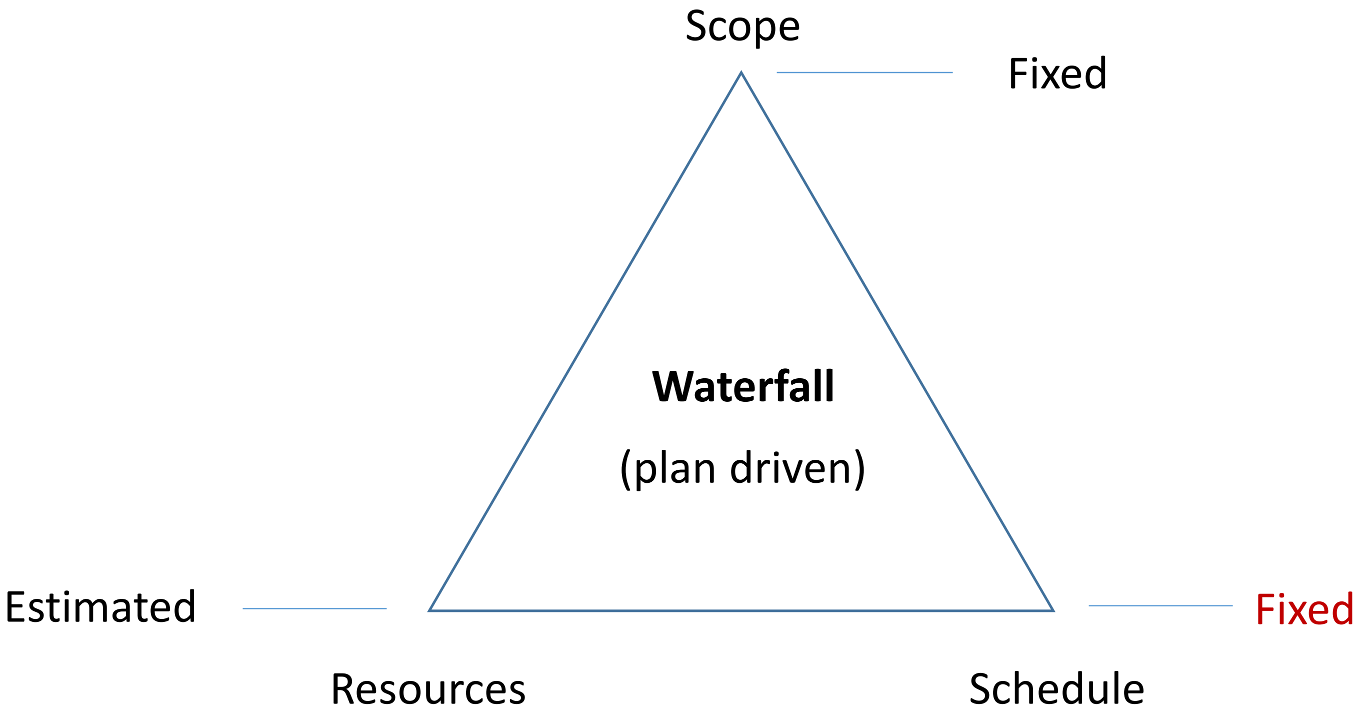
(plan driven)

Estimated

Resources

Schedule

Waterfall Constraints



Scope

Fixed

Waterfall

(plan driven)

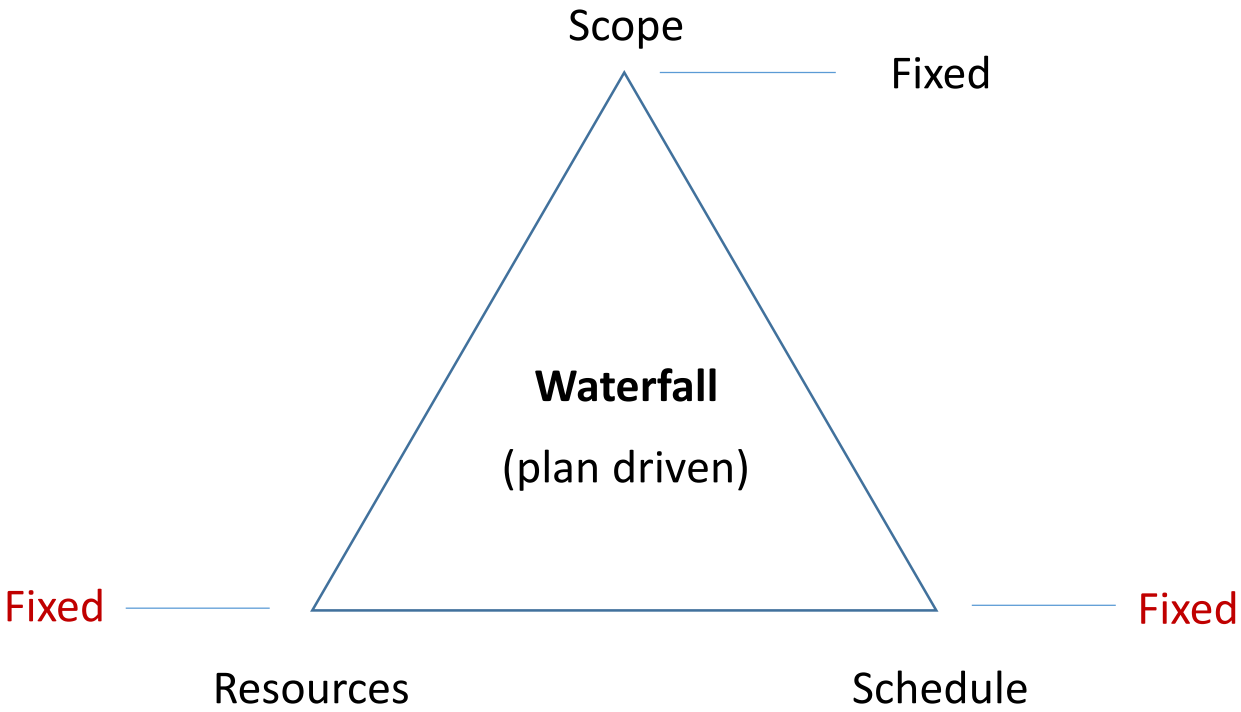
Estimated

Fixed

Resources

Schedule

Waterfall Constraints



Scope

Fixed

Waterfall

(plan driven)

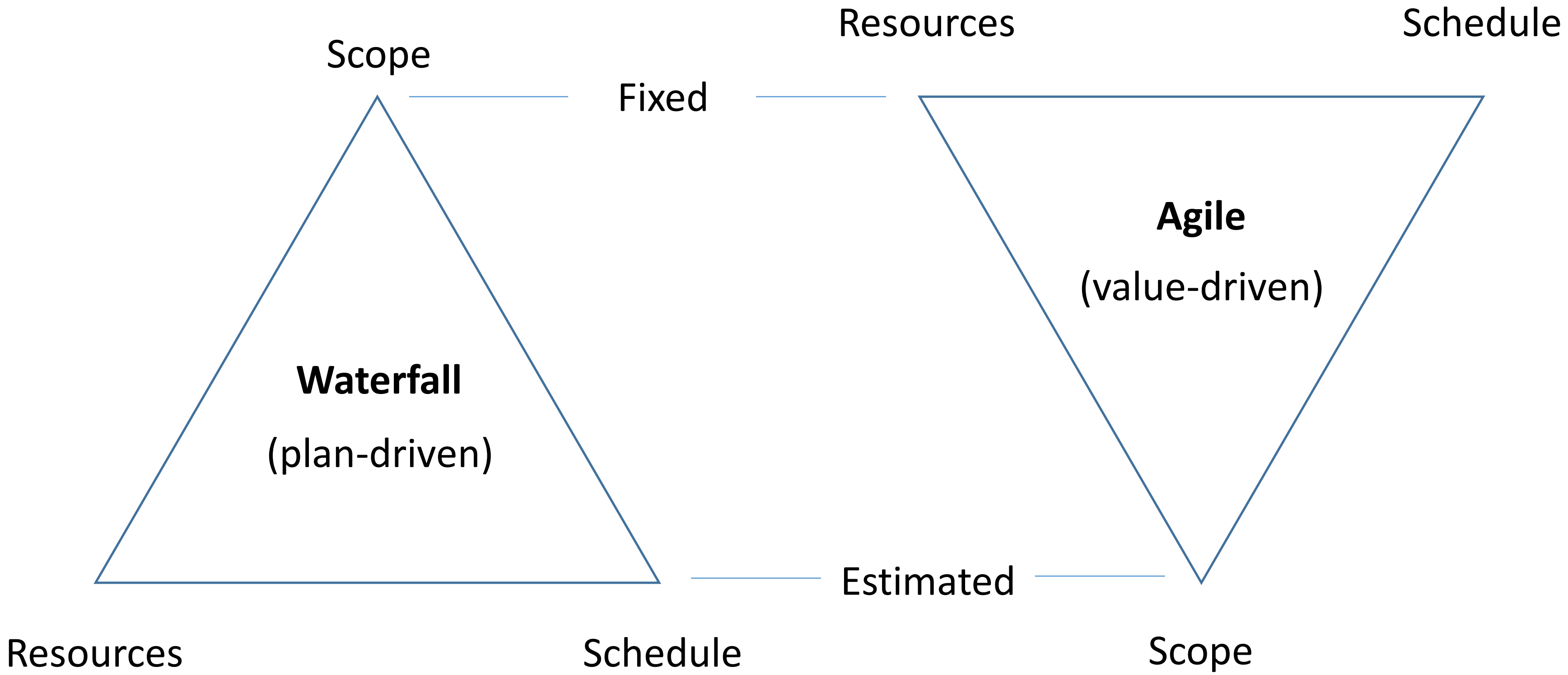
Fixed

Fixed

Resources

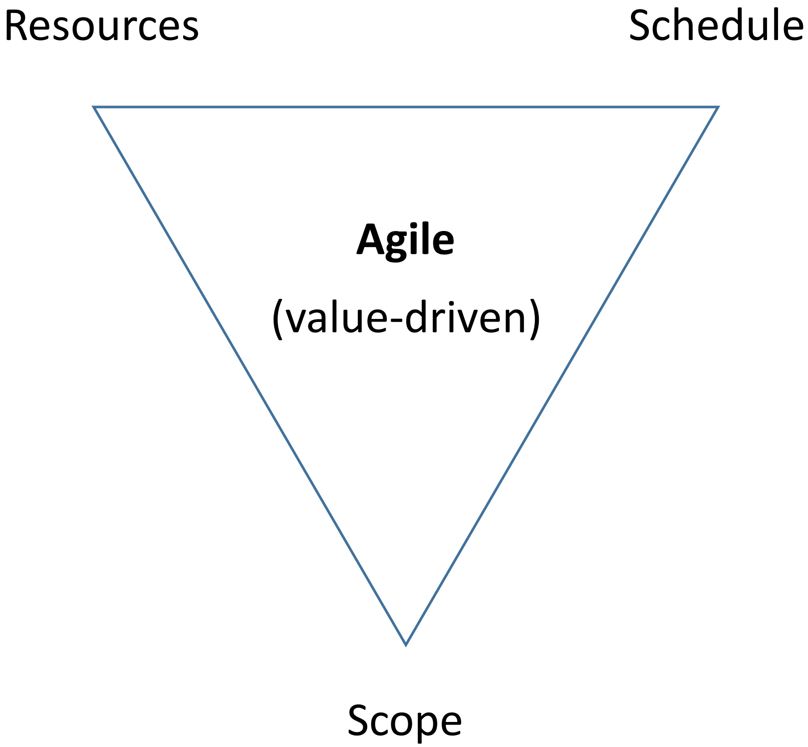
Schedule

Agile Constraints



Agile Constraints

- Avoids the Mythical Man Month
- Protects against slipping deadlines
- Prevents scope creep
- Protects quality



Agile Constraints Protect Quality

- Allowing line workers to halt production for quality issues
- If we sacrifice quality we accumulate technical debt
- Technical debt is costly and can stall a project

If you want go fast…
you must have quality!

3. Prioritizing Value

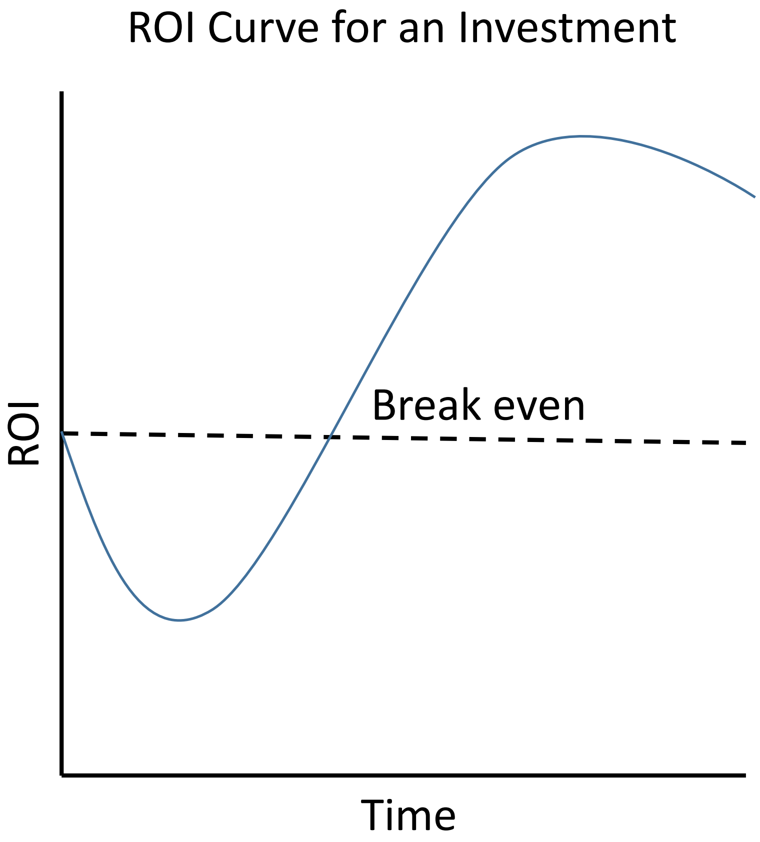
Quick Lesson in Economics

1. Return on Investment
2. Pareto Principle
3. Opportunity Cost

Source: http://myhomeworkhelp.com/economics-homework-help/

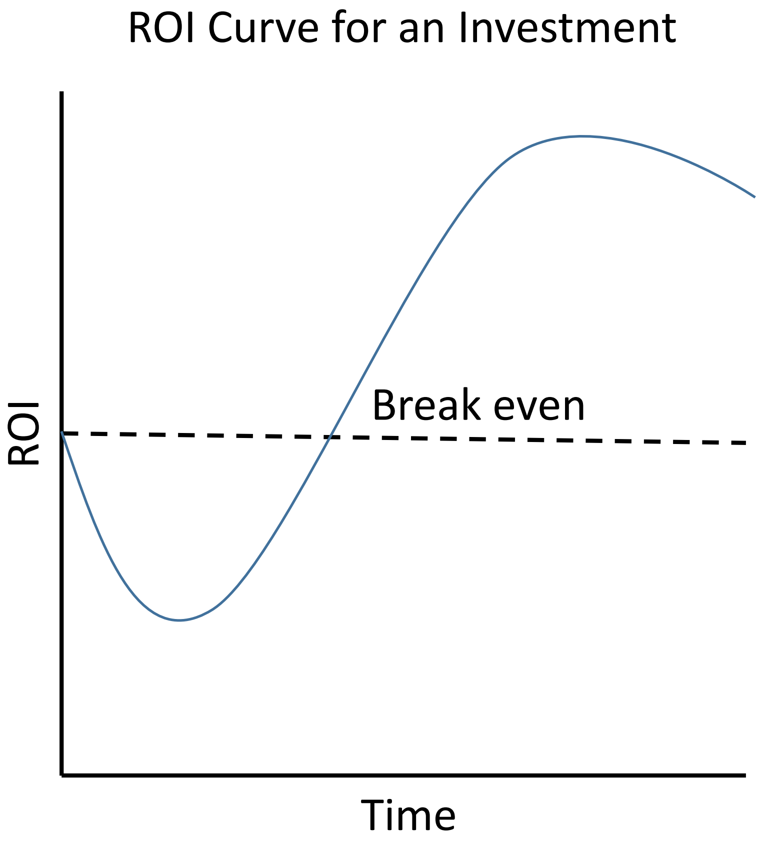
Return on Investment

- $ROI = \frac{Value - Cost}{Cost}$
- High ROI => lots of value
- Low ROI => some value
- Neg. ROI => lost value



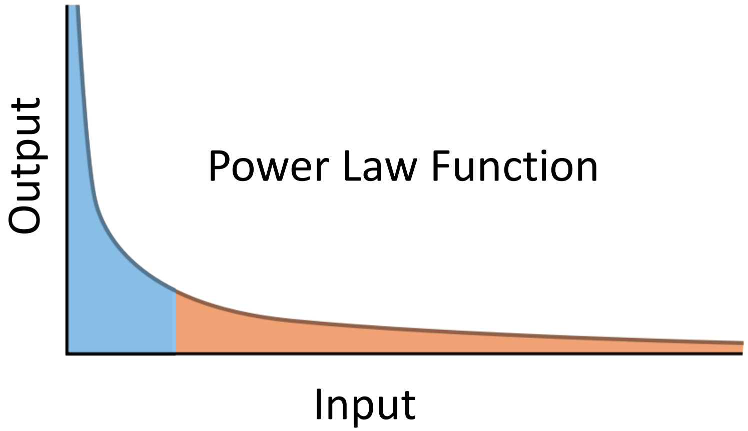
Return on Investment

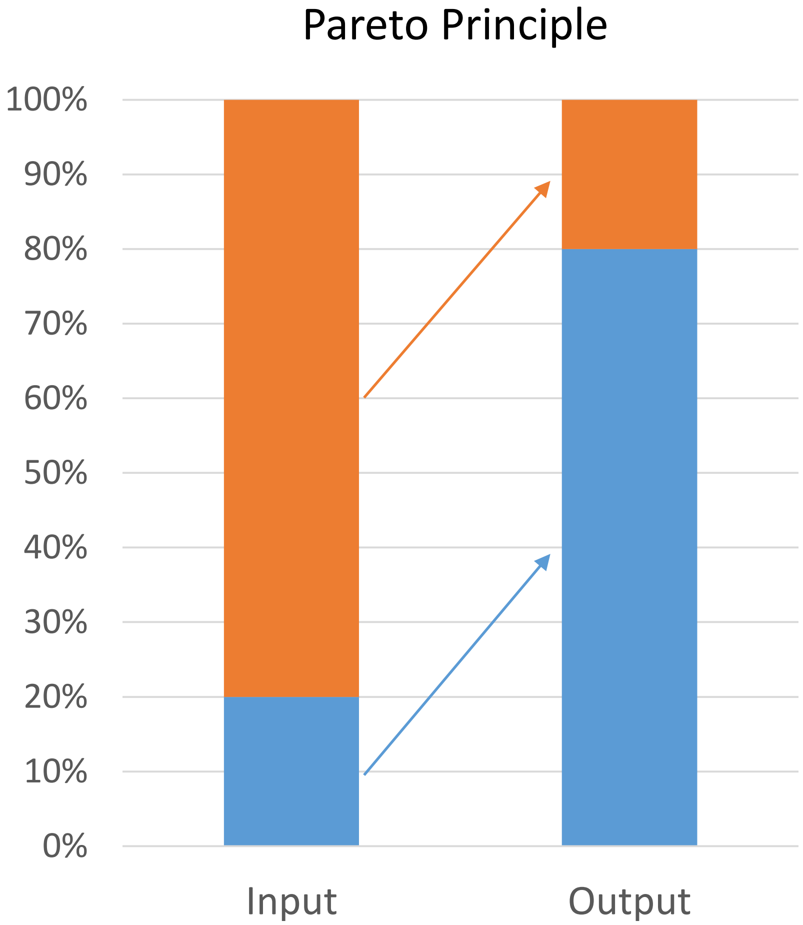
- Each feature has ROI
  - Cost to develop
  - Value to business
- Project ROI is sum of all feature ROIs
- Goal is to maximize ROI of the project



Pareto Principle

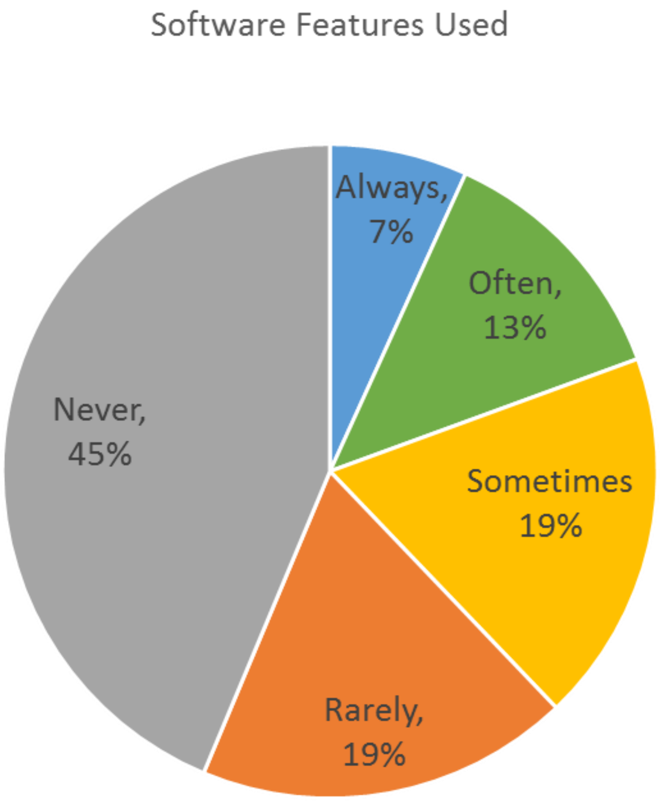
- 80/20 rule
- 80% of effects
- 20% of causes
- Power law function
- Diminishing marginal returns





Pareto Principle of Software Feature Usage

- Features:
  - 20% of features
  - 80% of value
- Traditional software is:
  - 20% high-value features
  - 80% low-value features



Source: Standish Group

Opportunity Cost

Source: http://www.ethicurean.com/2009/03/03/free-lunch-program-in-new-england/

Opportunity Cost

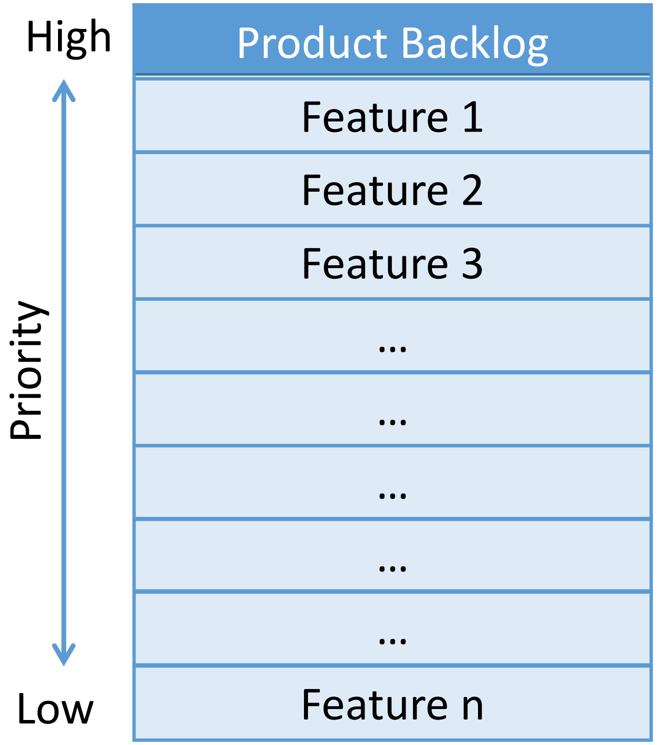
- Cost of foregone alternative options
- True cost = explicit cost + implicit cost
- Must be included in cost-benefit analysis

Stu's Views
© 2004 Stu All Rights Reserved www.STUS.com
S U
I hope you appreciate that each "walk" costs $175 of my billable time.
I hope you appreciate that I'm your only friend.

Source: http://www.stus.com/

Prioritizing Features by Business Value

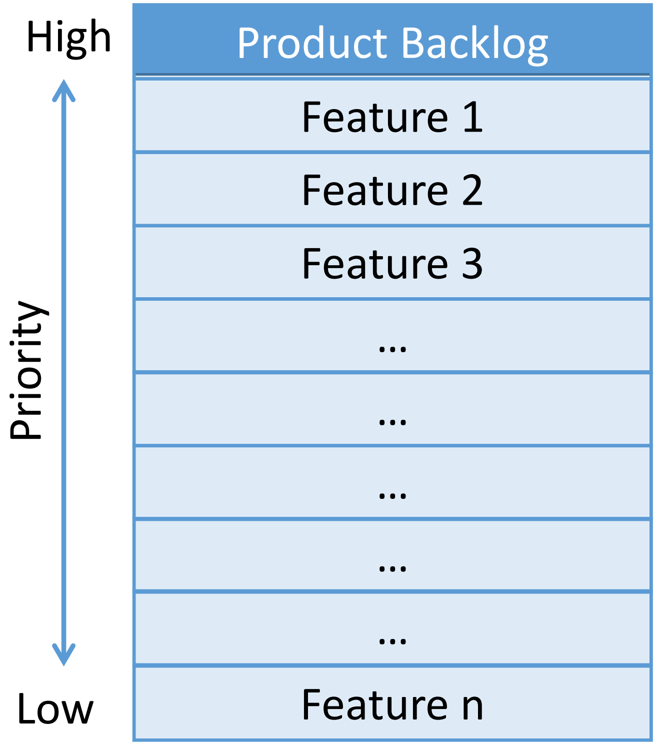
- Product backlog
- List of features
- Ordered by business value
- Highest priority on top
- Create features in order
- Deliver features in order



Product Backlog
Feature 1
Feature 2
Feature 3
…
…
…
…
…
Feature n

Prioritizing Features by Business Value

- Maximizes ROI at each step
- Follows 80/20 rule
- Recognizes opportunity cost



High

Priority

Low

Product Backlog

Feature 1

Feature 2

Feature 3

…

…

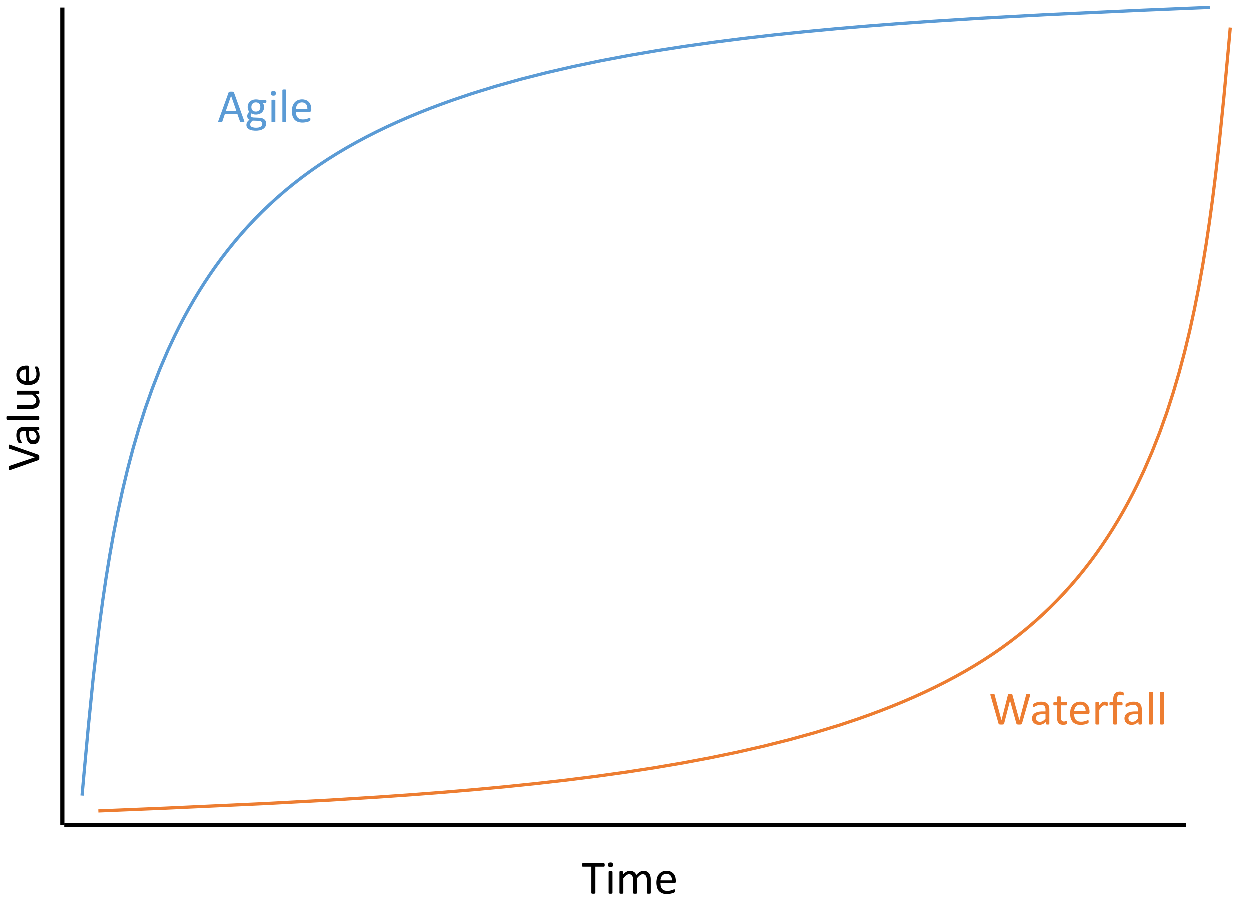
…

…

…

Feature n

Value over Time



Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

4. Embracing Change

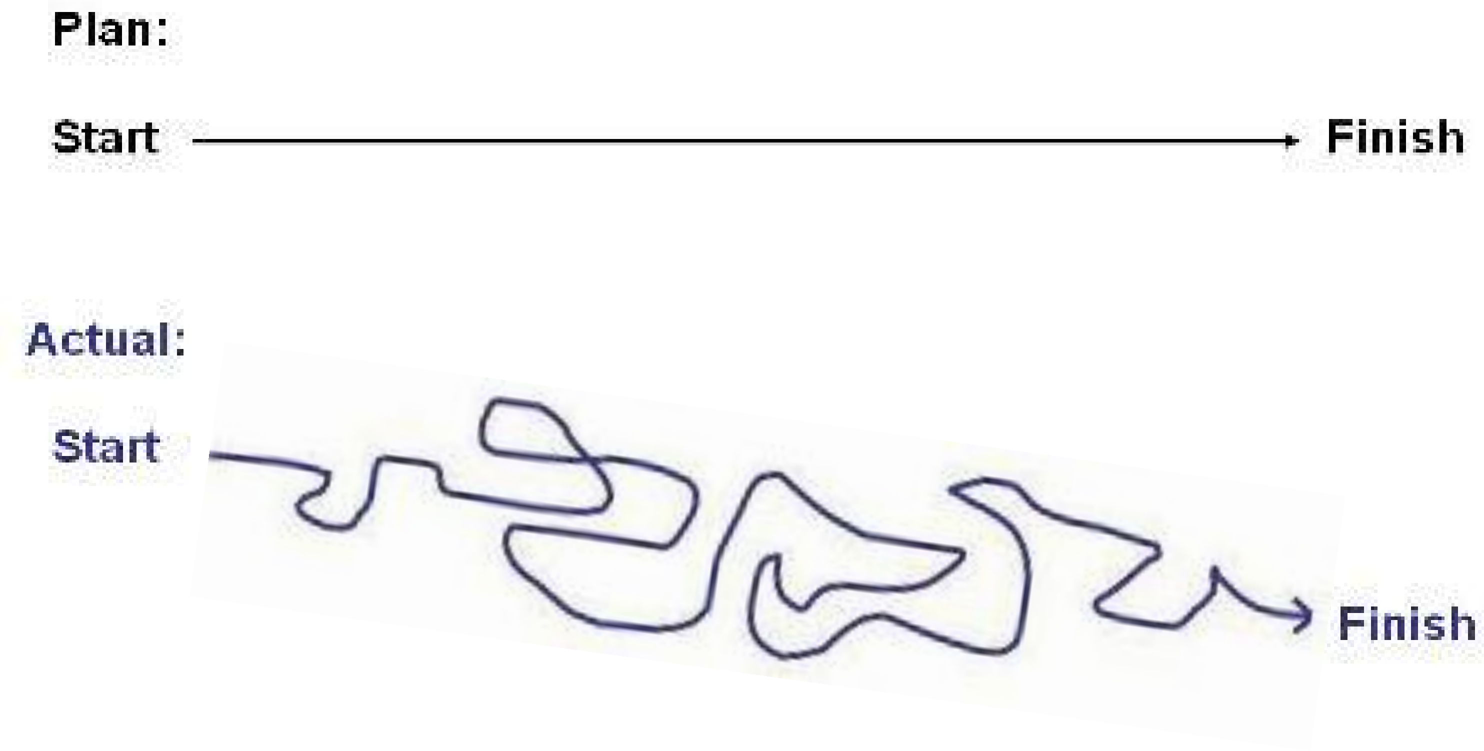
Waterfall Assumes that Things Will Go According to a Plan

Plan:

Start → Finish

Source: Doug DeCarlo – eXtreme Project Management

Waterfall assumes that everything will go according to plan



Source: Doug DeCarlo – eXtreme Project Management

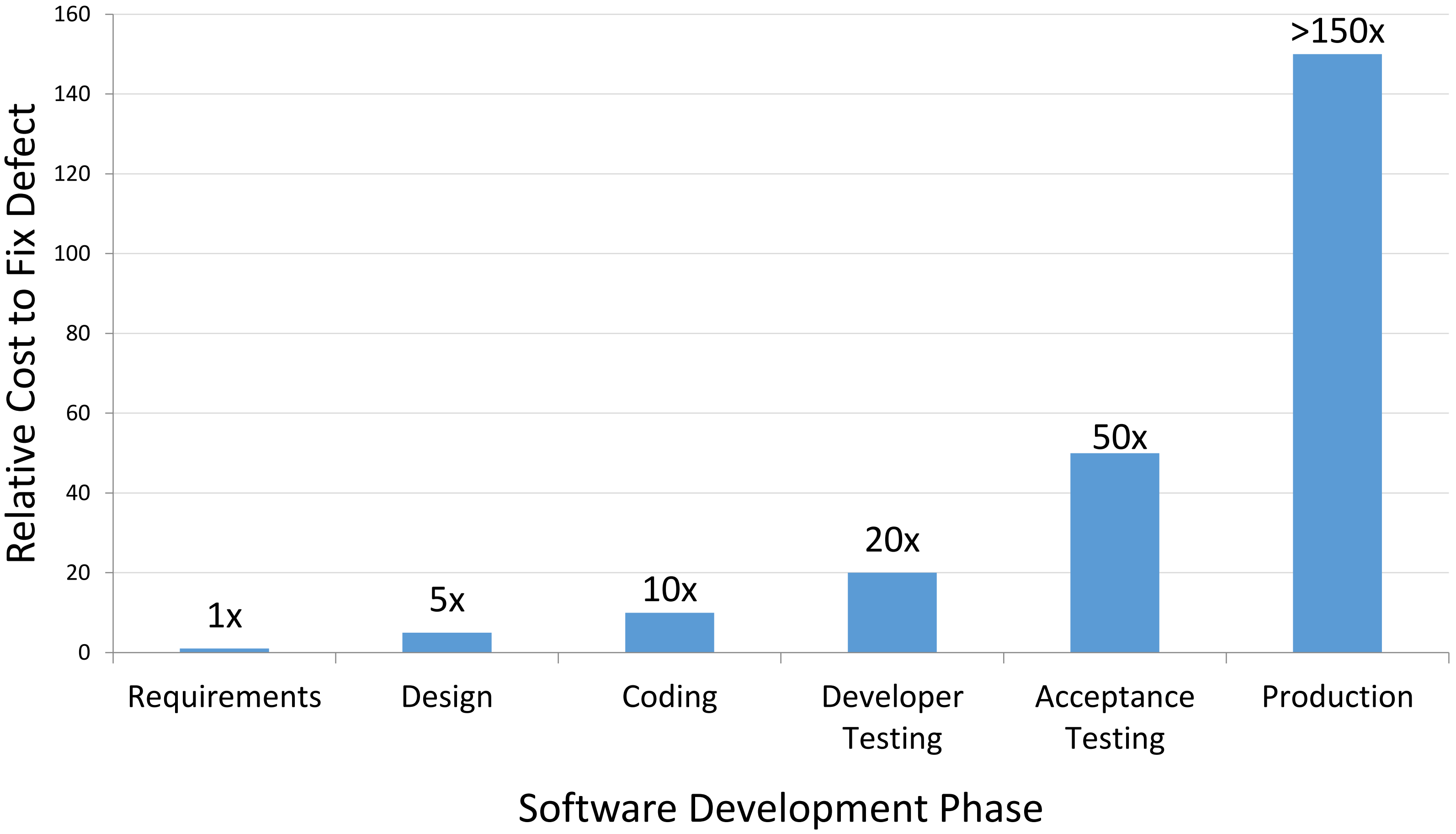
Waterfall

- Waterfall assumes:
  - Users actually know what they want
  - Markets will not change during development
  - There is nothing new or unknown
  - Technology is stable and mature
  - All of the pieces will fit together in the end

Waterfall

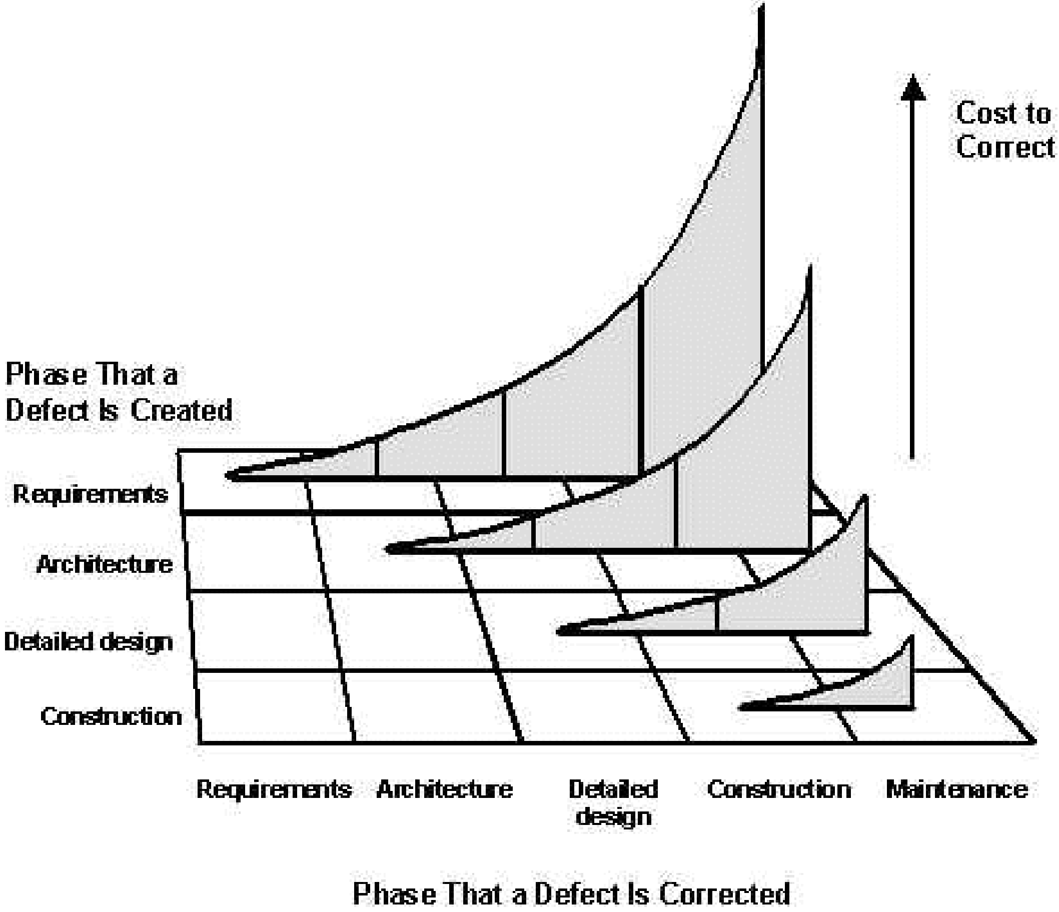
- Reality for most projects is:
    - Requirements are not stable
    - Requirements are just assumptions

Cost of Fixing Defects in Waterfall



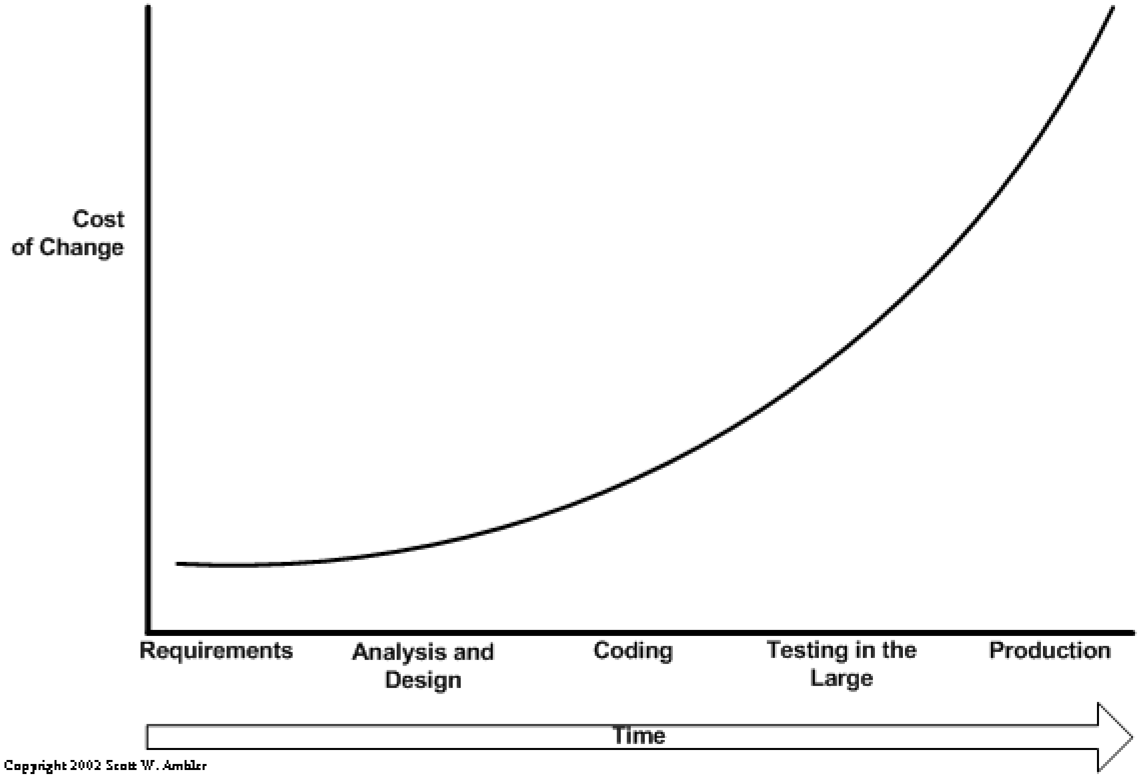
Original Source: Barry Boehm, "Equity Keynote Address" March 19, 2007

Cost of Fixing Defects in Waterfall



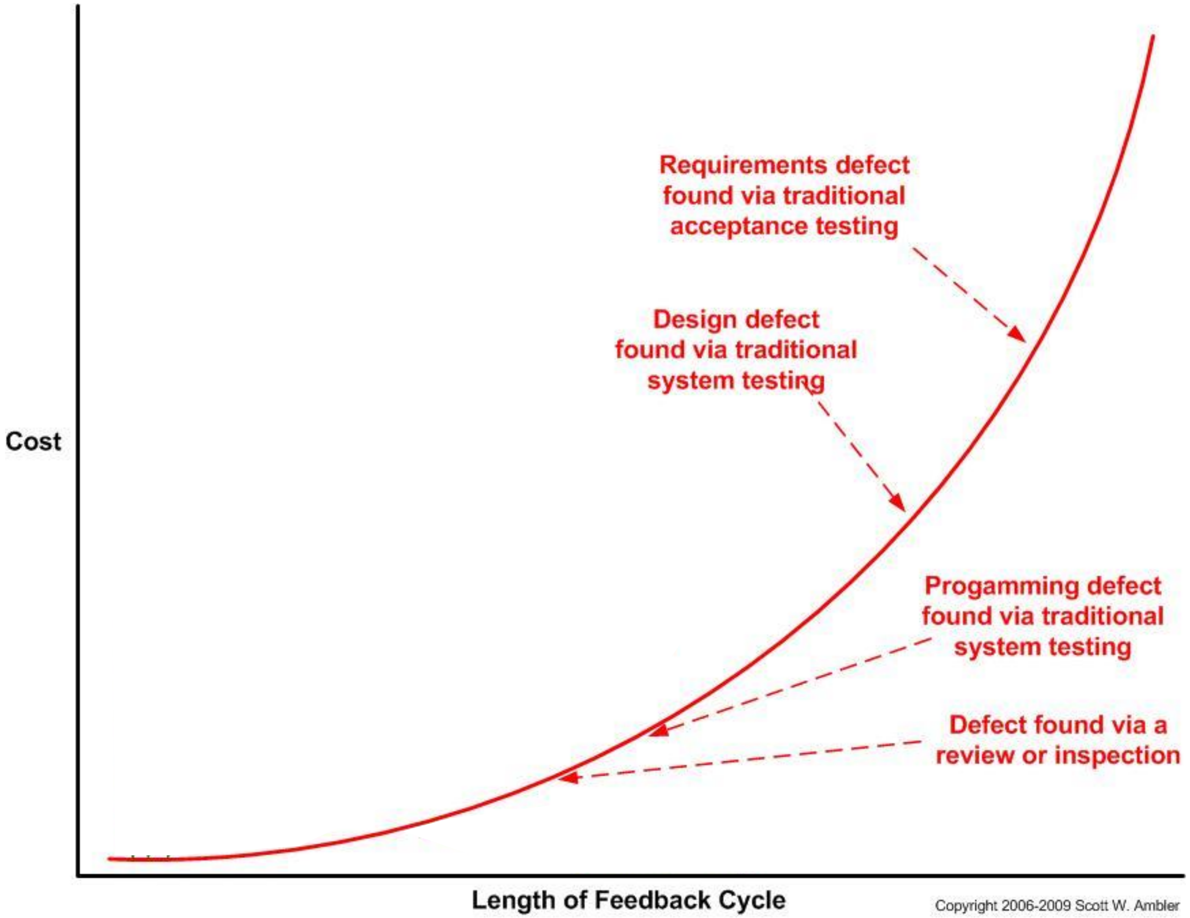
Source: http://www.stevemcconnell.com/articles/art04.htm

Cost of Change in Waterfall



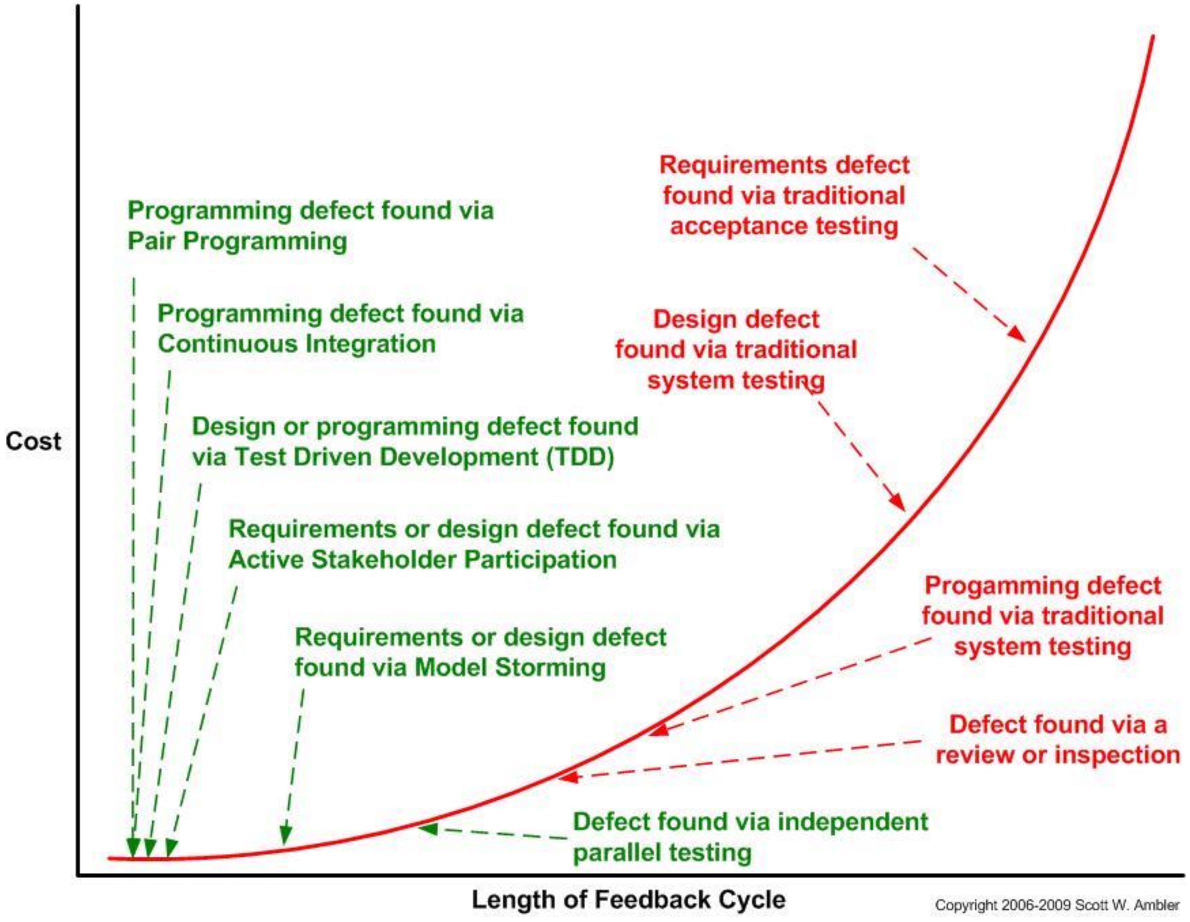
http://www.agilemodeling.com/essays/costofchange.htm

Finding Defects in Waterfall



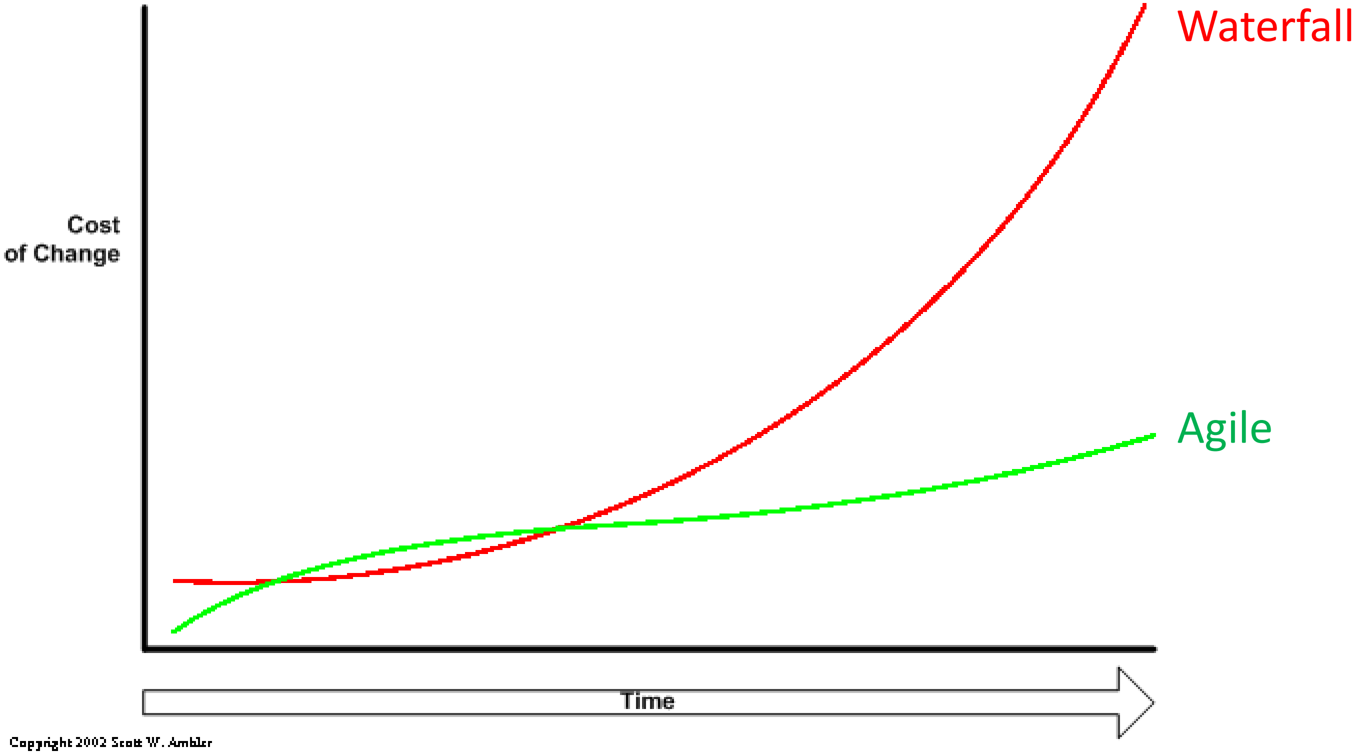
http://www.agilemodeling.com/essays/costofchange.htm

Finding Defects in Agile



http://www.agilemodeling.com/essays/costofchange.htm

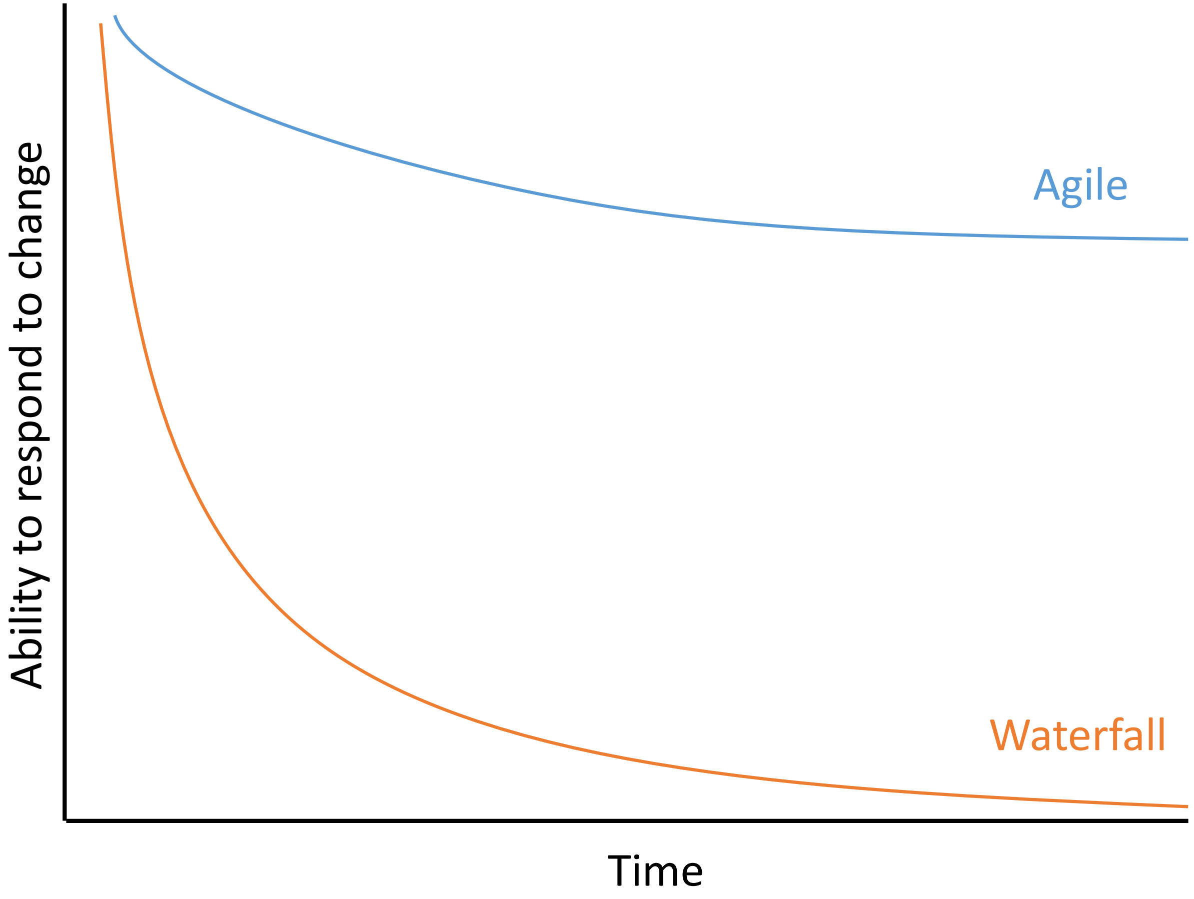
Cost of Change in Agile



Copyright 2002 Scott W. Ambler

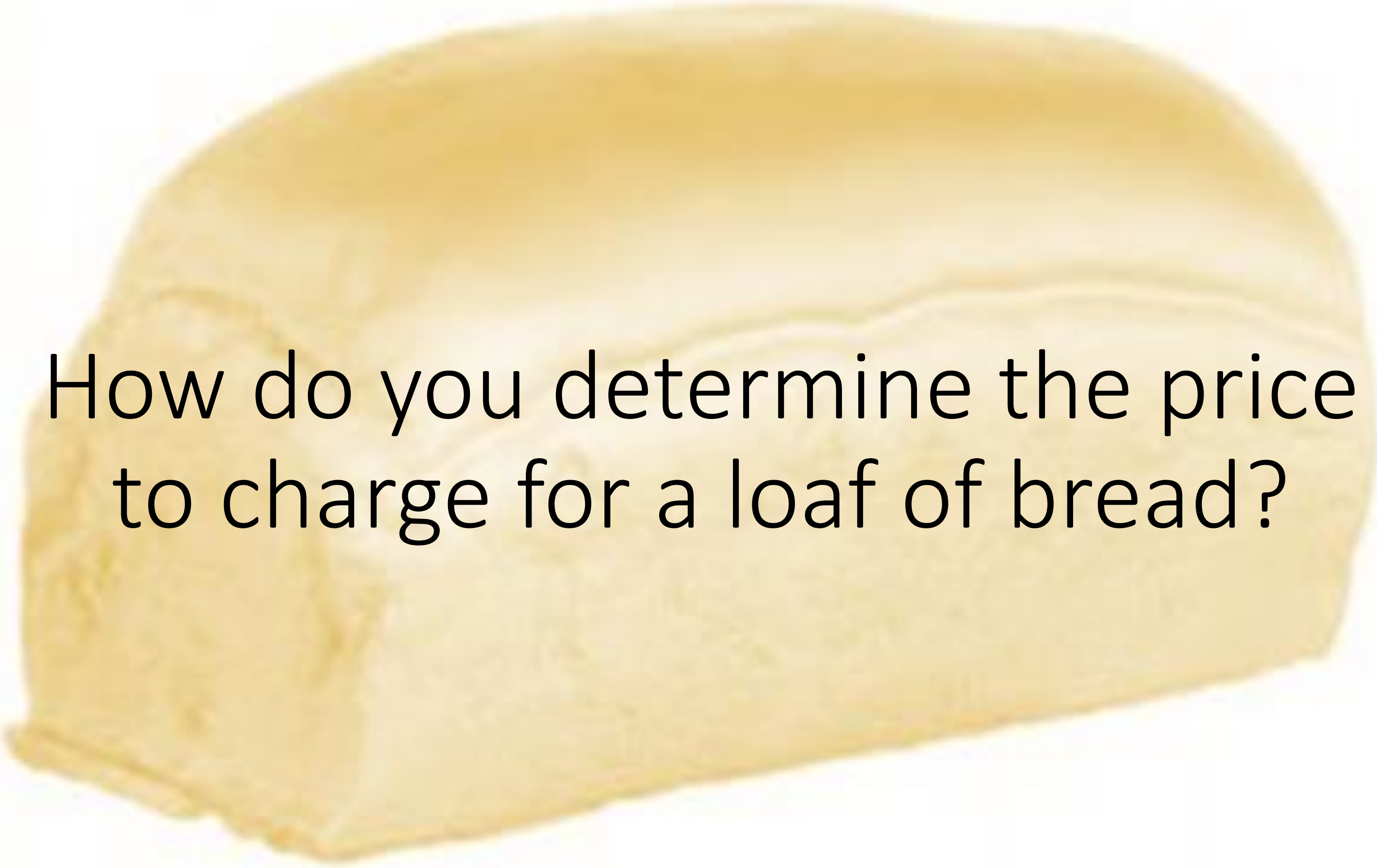
Source: http://www.agilemodeling.com/essays/costOfChange.htm

Adaptability



Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

5. Self-Organization



How do you determine the price to charge for a loaf of bread?

Command vs. Market Economies

- Tale of two economies:
  - Command economy
  - Market economy

Source: Wikipedia

Source: Britannica

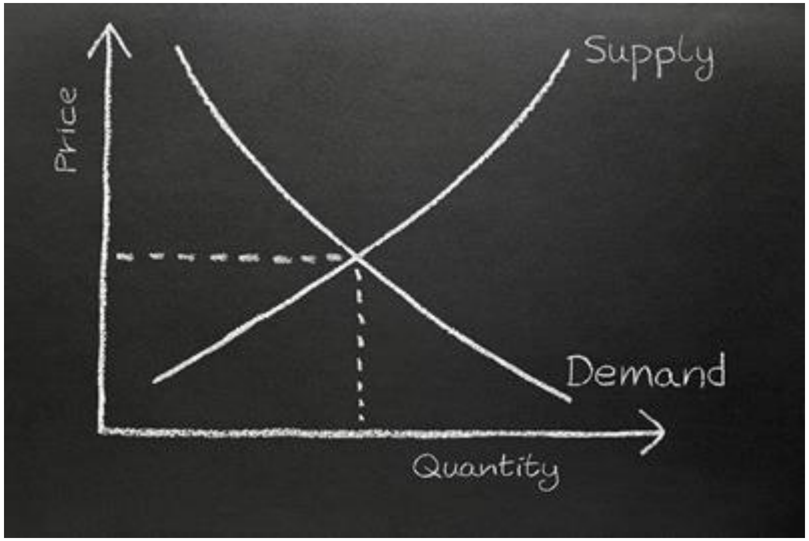
Command Economy

- Central Planning Agency makes decisions
- Controlled production
    - Too high => surplus
    - Too low => shortfall
- Controlled prices
    - Too high => no sale
    - Too low => deficit
- Millions of decisions

Source: Wikipedia

Command Economy

- Goal: Maximize Social Welfare
- Competitive Market Equilibrium
- Produces market inefficiencies
- “Controlled failure”



Source: https://content.dodea.edu/
VS/HS/DVHS_Courses/Economics/syllabus.html

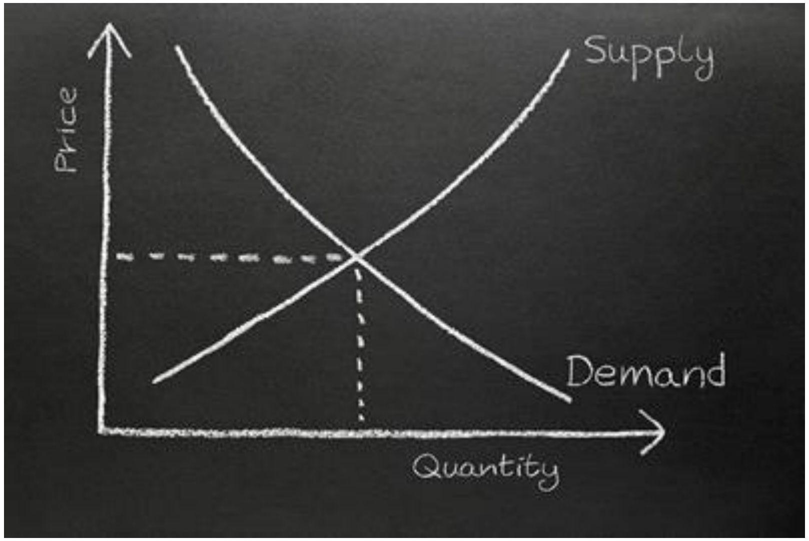
Market Economy

- Market makes decisions
  - Individuals
  - Interactions
- Produces & Consumers
  - Supply
  - Demand
- Millions of decisions

Source: Britannica

Market Economy

- Goal: Maximize Social Welfare
- Competitive Market Equilibrium
- Extremely efficient
- “Chaotic success”

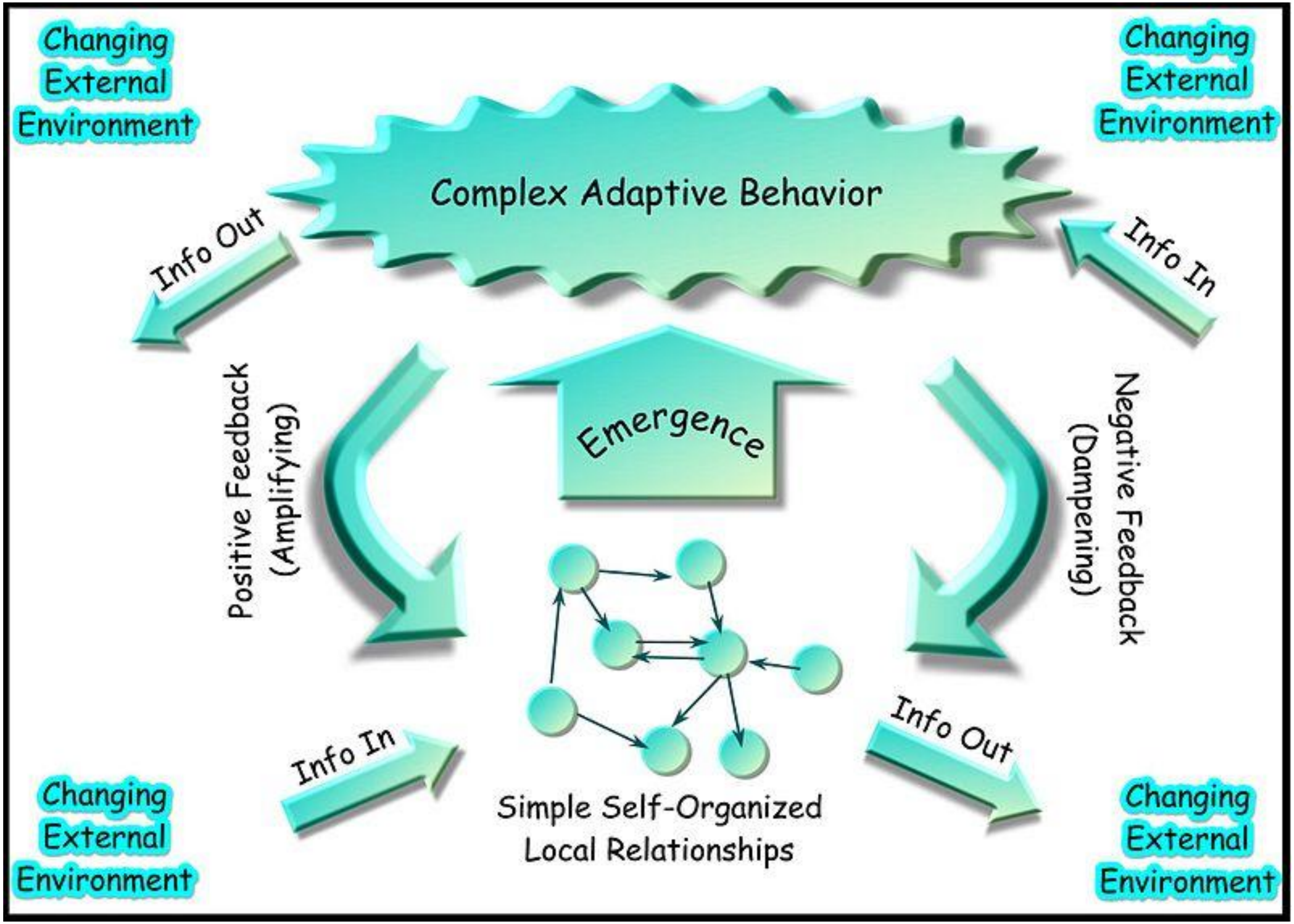


Source: https://content.dodea.edu/VS/HS/DVHS_Courses/Economics/syllabus.html

Complex Adaptive Systems

- System
  - collection of interconnected things
- Complex
  - dynamic network of interactions
- Adaptive
  - changes in response to environment
  - to increase survivability

Source: http://integral-options.blogspot.com/2013/03/peter-fryer-brief-description-of.html


Changing External Environment
Changing External Environment
Complex Adaptive Behavior
Info Out
Info In
Positive Feedback (Amplifying)
Emergence
Negative Feedback (Dampening)
Info Out
Info In
Simple Self-Organized Local Relationships
Changing External Environment
Changing External Environment

Source: Wikipedia

Inversion of Control

- Top-down
- Command and Control
- Bureaucracy

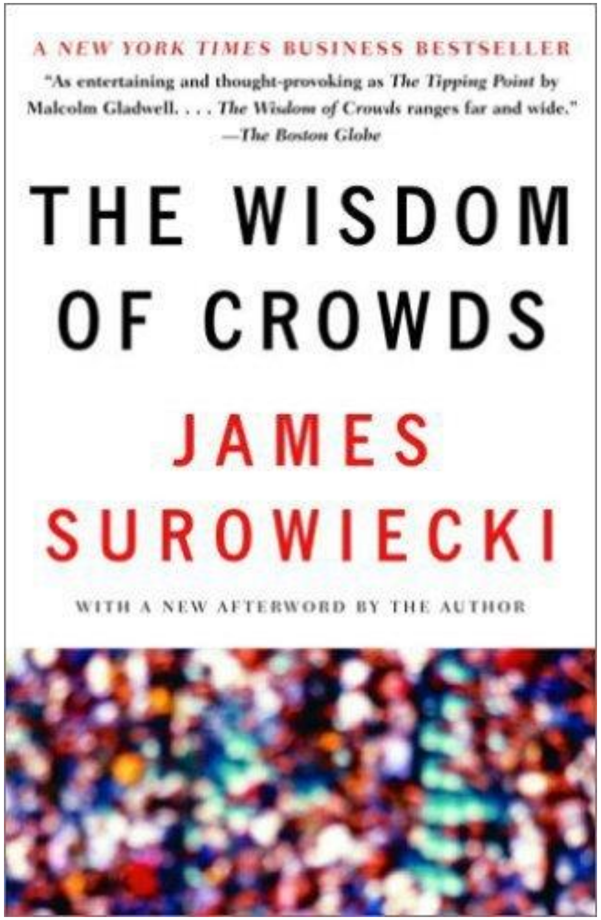
vs.

- Bottom-up
- Self-organization
- Adhocracy

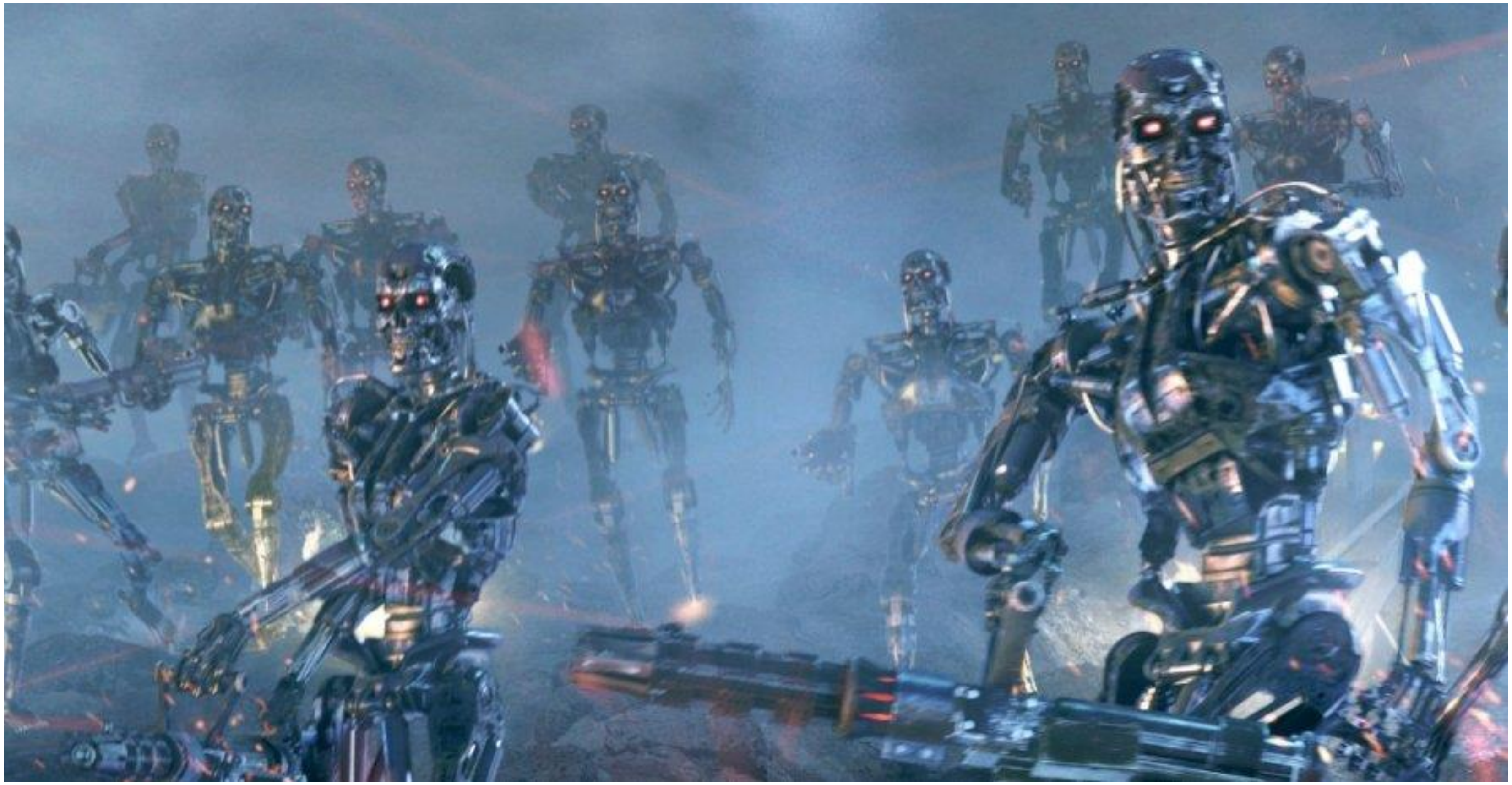
Source: http://funnyasduck.net/post/10458

Wisdom of the Crowd

- Take collective guesses of the crowd
- Aggregated answer is better than expert in many cases
- Works well for:
    - Quantity estimation
    - General knowledge
    - Spatial reasoning
- Not all crowds are wise!


A NEW YORK TIMES BUSINESS BESTSELLER
"As entertaining and thought-provoking as The Tipping Point by Malcolm Gladwell. . . . The Wisdom of Crowds ranges far and wide."
—The Boston Globe
THE WISDOM OF CROWDS
JAMES SUROWIECKI
WITH A NEW AFTERWORD BY THE AUTHOR

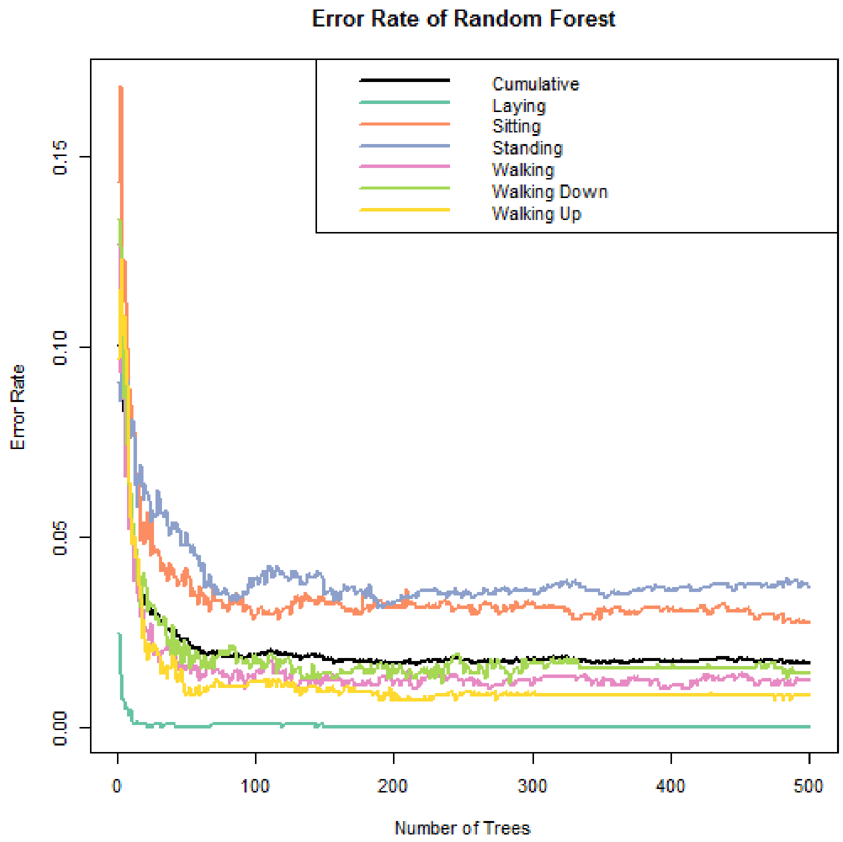
Wisdom of the Machines



Source: http://www.internationalhero.co.uk/t/terminators.htm

Wisdom of the Machines

- Machine learning:
- Ensemble learning
  - Many weak learners vs. one strong learner
  - If better than random chance → aggregate
- Random-forest classifiers



Agile Team Self-Organization

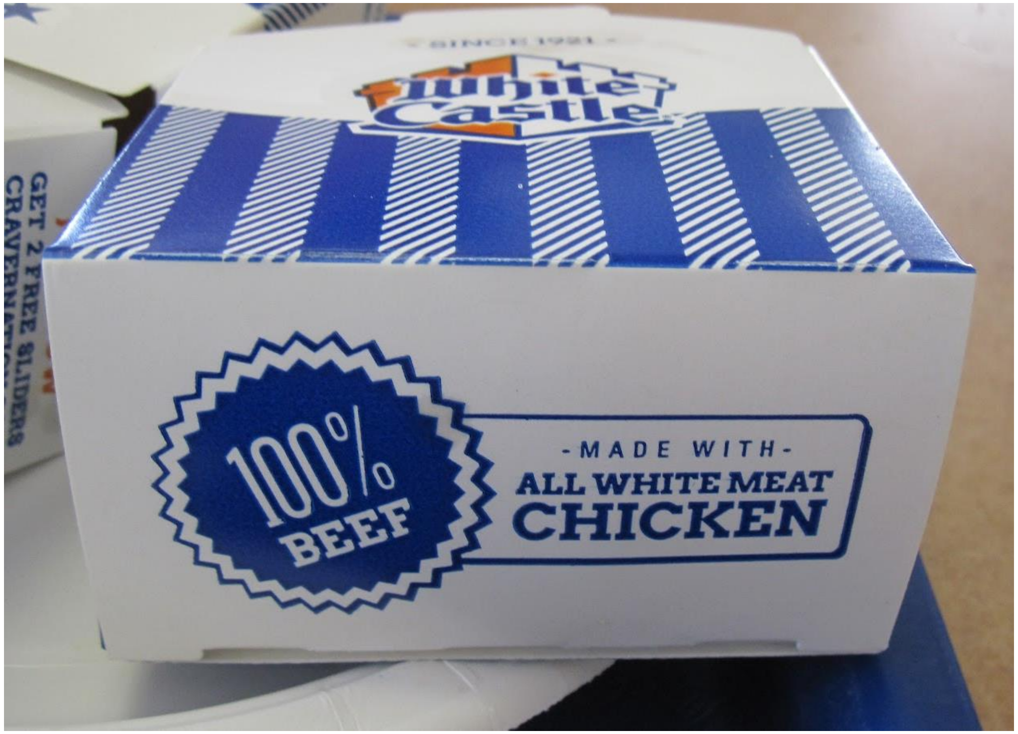
- Self-Organizing
  - vs. command and control
- Inversion of Control
  - Bottom-up vs. Top-down
- Complex Adaptive Systems
  - More efficient than top down control systems
- Wisdom of the Crowds
  - Better decisions in aggregate than individuals

Source: http://www.telegraph.co.uk

6. Efficient Communication

Cost of Poor Communication

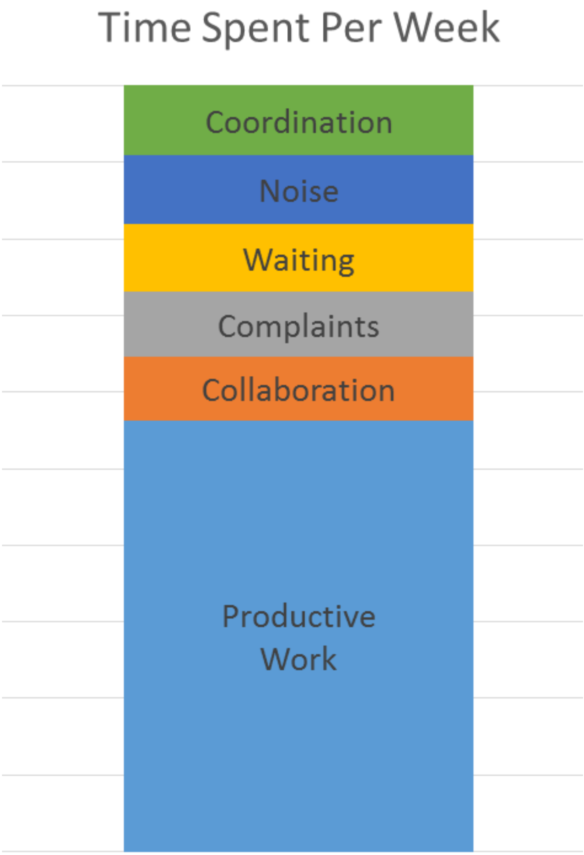
- Cost is enormous
- Hard to quantify
- Hidden cost
- Expense is real



Source: http://www.cathy.willman.com/2012/06/what-boys-need.html

Cost of Poor Communication

- 17.5 hrs / person / week
- Top 5 issues identified:
  1. Waiting for information
  2. Unwanted communication (noise)
  3. Inefficient coordination among team members
  4. Barriers to collaboration
  5. Customer complaints



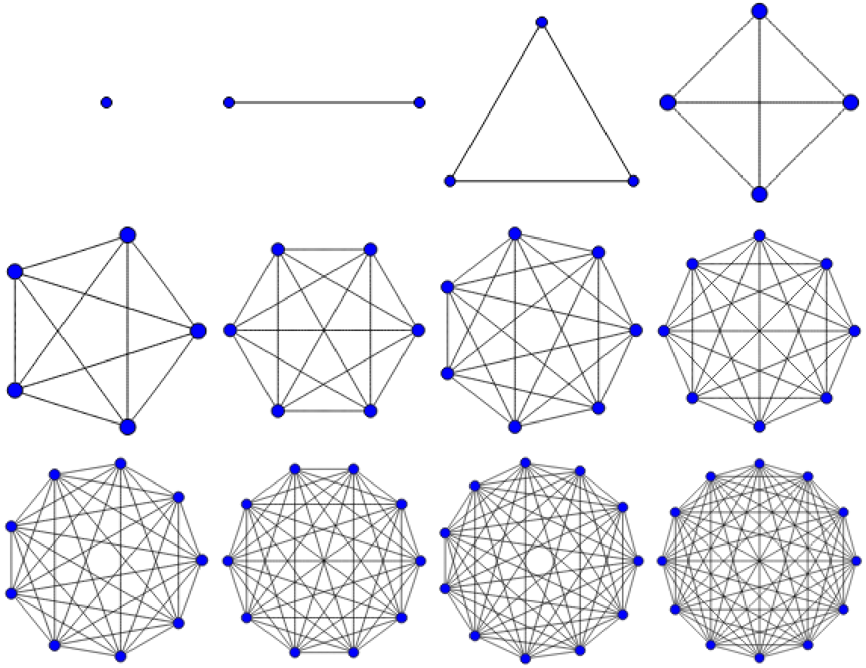
Source: http://thoughtleadership.sismarketresearch.com/industrial-b2b-journal /2009/3/10/smb-communications-pain-study-white-paper-uncovering-the-hid.html

Total estimated annual cost of poor communication per enterprise knowledge worker: **$50,562**

Source: http://thoughtleadership.sismarketresearch.com/industrial-b2b-journal/2009/3/10/smb-communications-pain-study-white-paper-uncovering-the-hid.html

Communication Structures - Complete Graph

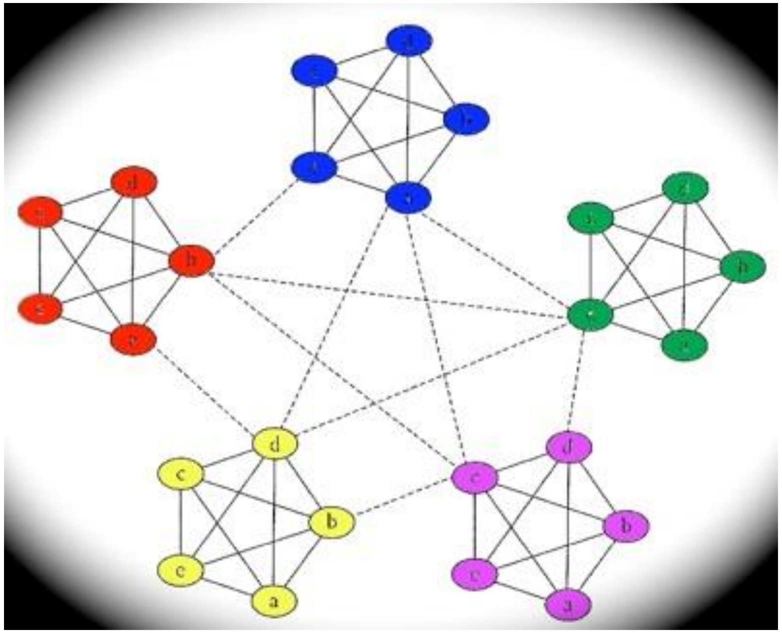
- Communication network modeled as a complete graph
  - Nodes = people
  - Edges = channels
- Edges increase by O(n^2) in the number of nodes
- Becomes extremely inefficient very fast
- Keep teams small



Source: Wikipedia

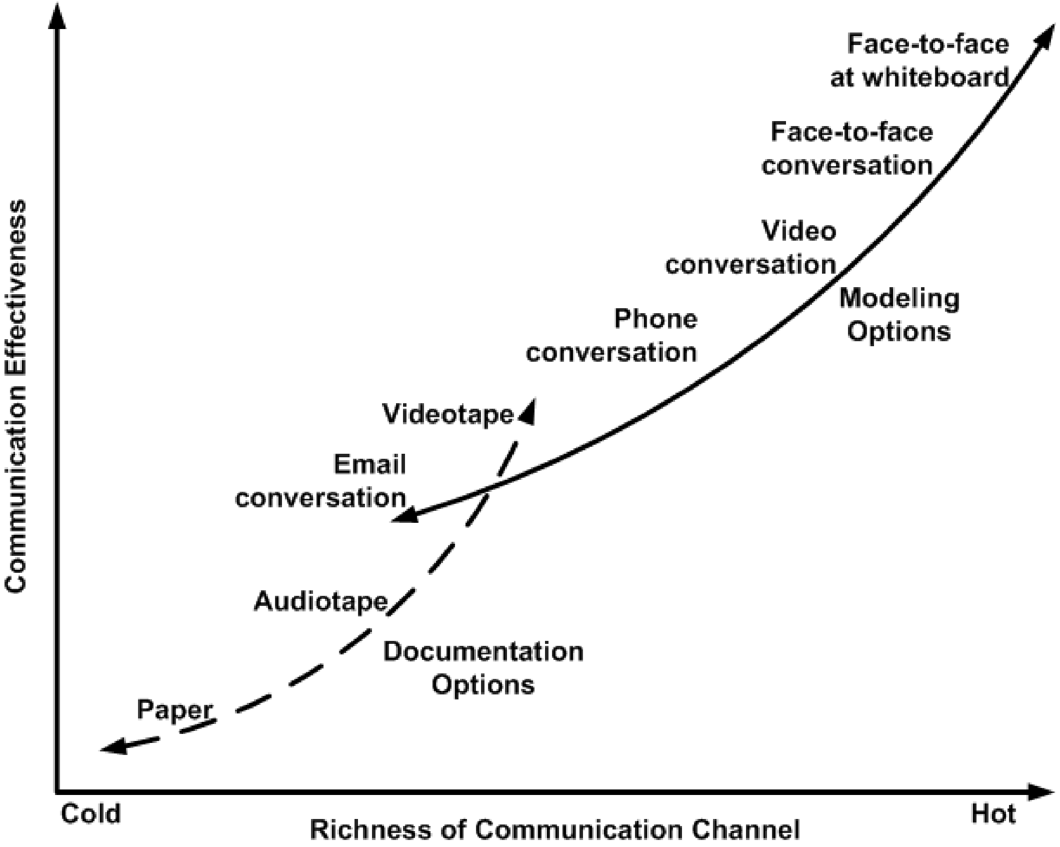
Communication Structures – Sparse Network

- Sparse Network
- More efficient for large numbers of nodes
- Highly cohesive units
- Loosely coupled whole
- Optimal interconnection densities follow 80/20 rule



Source: http://howtosavetheworld.ca/images/networkofdenseclusters.jpg

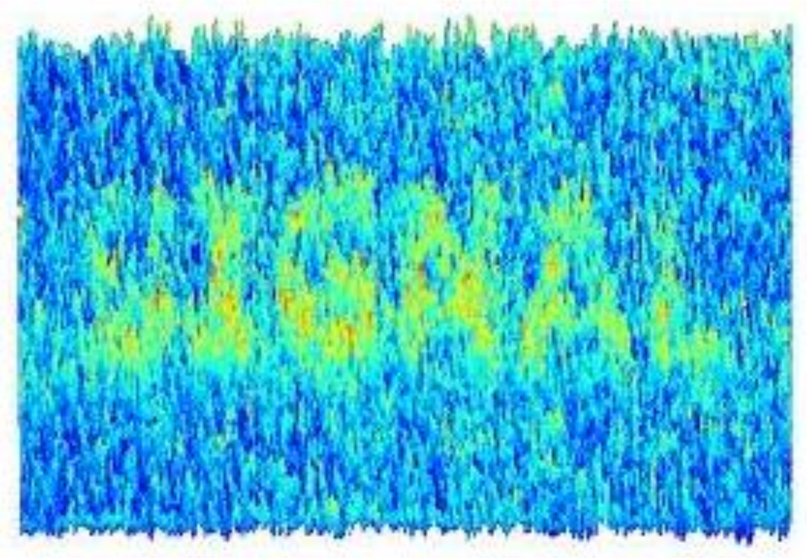
Effectiveness of Communication



Copyright 2002-2005 Scott W. Ambler
Original Diagram Copyright 2002 Alistair Cockburn

Signal-to-Noise Ratio

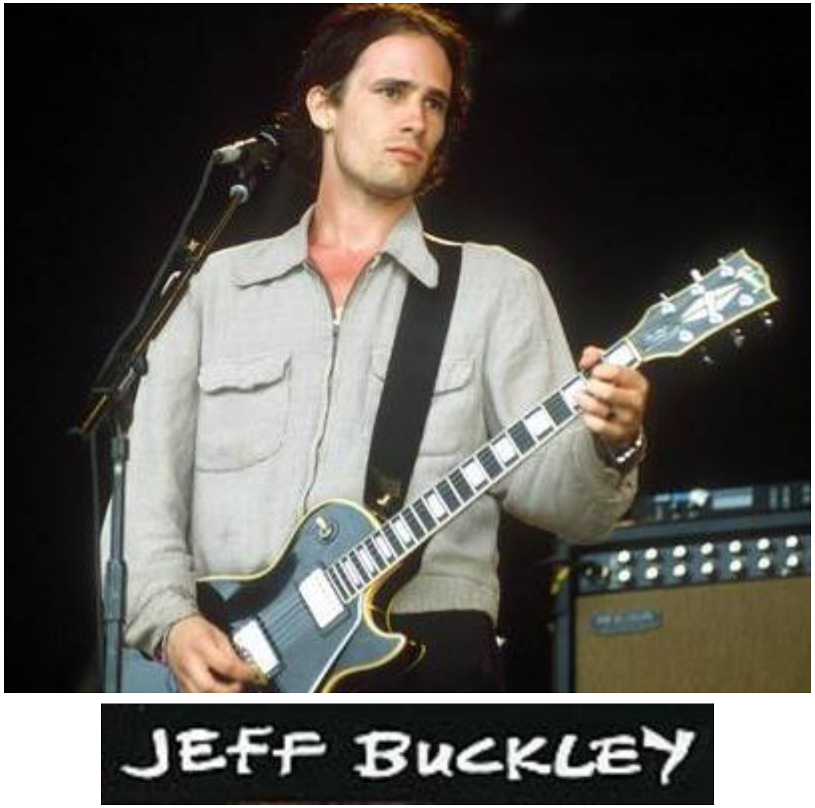
- SNR = P(signal) / P(noise)
- Signal = message
- Noise = everything else
- Goal is to maximize signal-to-noise ratio



Source: http://uber.la/2012/05/signal-to-noise/

Maximize Signal

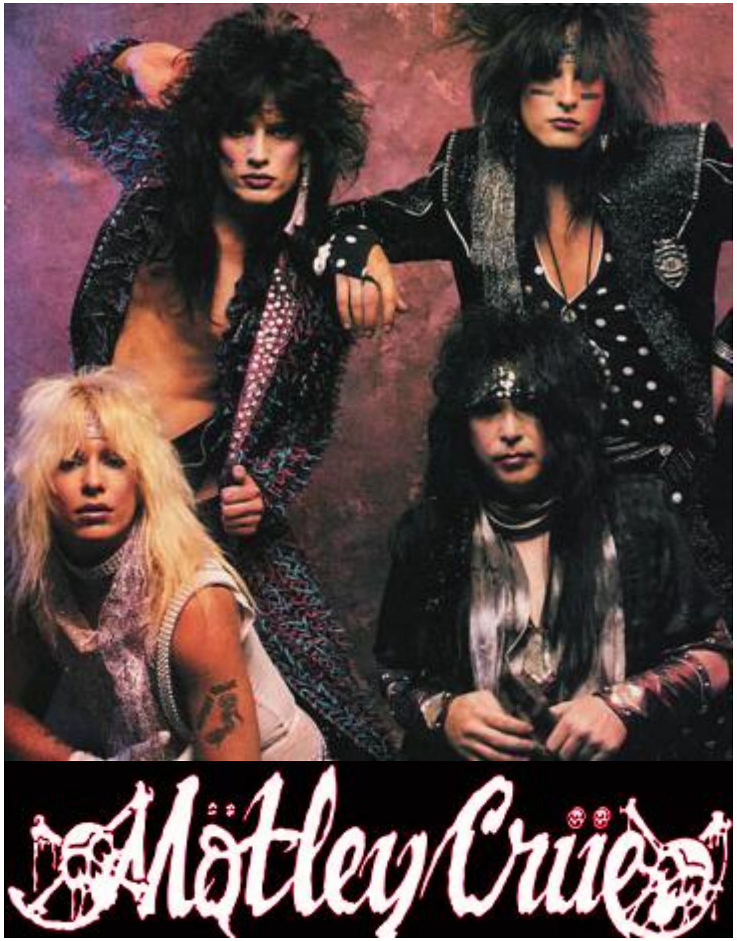
- Signal is necessary communication
- Maximize bandwidth of the signal
  - Face-to-face
  - Whiteboard
- Reduces likelihood of miscommunication



Source: http://www.virginmedia.com/images/jeff-buckley-431.jpg

Minimize Noise

- Noise is unnecessary communication
- Minimize Noise
  - Minimize bulk emails
  - Room layouts
- Environmental noise
  - Intelligible speech noise
- Cost of context switching from interruption is huge



Source: Motley Crue

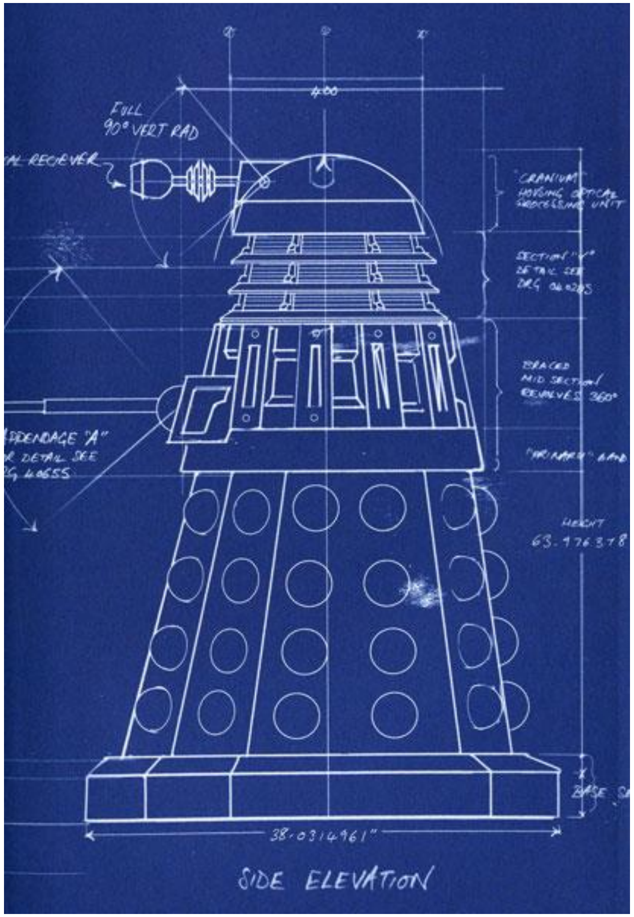
Agile Communication

- Small teams
- Whole team
- Face-to-face Collaboration
- Pair programming
- Daily stand-up
- Information radiators
- Transparency

Source: http://www.dvdizzy.com/peanuts-1970s-vol2.html

Waterfall: Documentation is Blueprint

- Waterfall:
  - Documentation is the blueprint
  - Code is the product
- Documentations is:
  - Typically is out of date
  - Rarely ever read
  - Not executable



Source: http://www.thinkgeek.com/product/f373/images/16693/

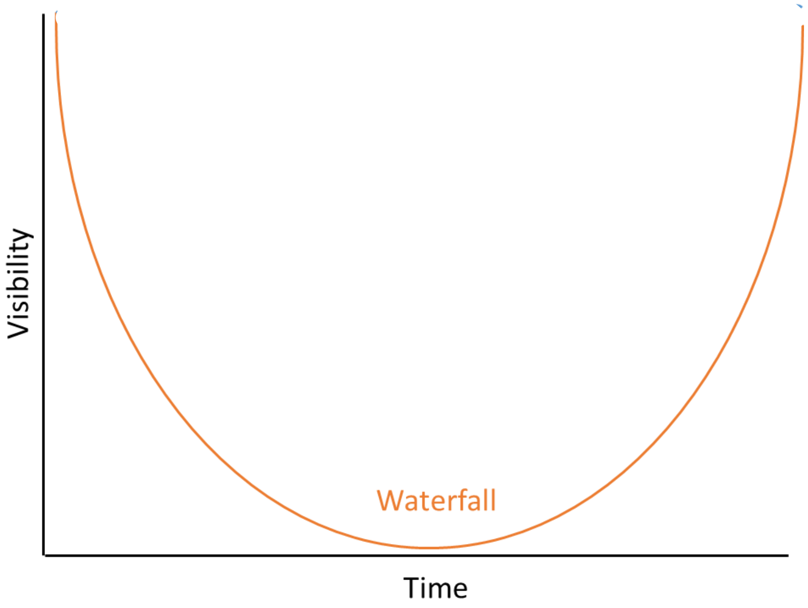
Agile: Code is Blueprint

- Agile:
  - Code is the blueprint
  - Executable software is the product
- Code is:
  - Always up to date
  - Continuously read
  - Executable specifications
- Self-verifying blueprints

Source: http://www.firebox.com/product/3621/Dalek-Replica

Visibility

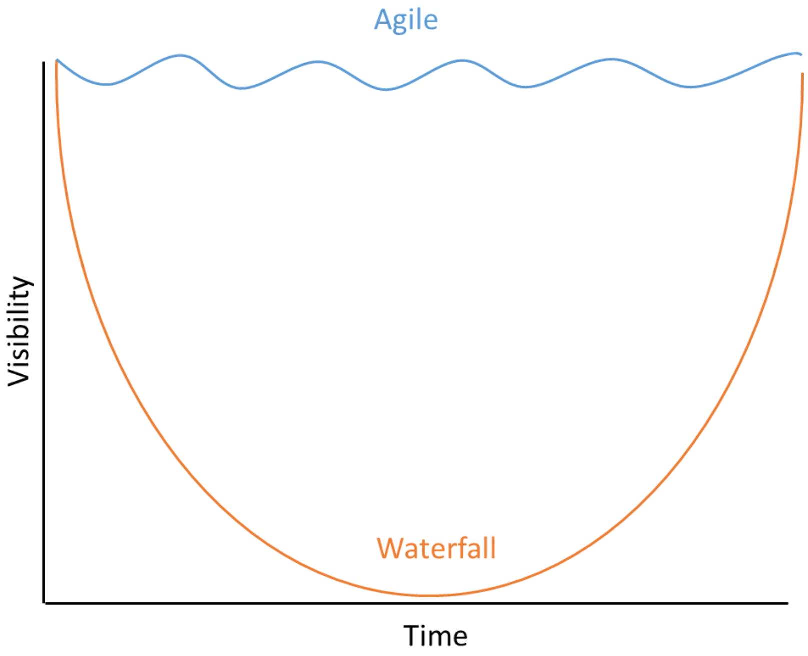
- Waterfall tends to hide many problems
- High visibility in the beginning
- Low visibility in the middle
- High visibility in the end



Original source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Visibility

- Agile provides visibility:
  - Information radiators
  - Regular inspection and adaptation
  - Frequent delivery of working software
- Agile is on the surface with project visibility
- Problems have no where to hide



Original source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Metrics

- Derived from traditional cost accounting
- Originally designed to manage workers and machines in assembly line processes
- Has since been applied to software development
- Goal is cost reduction
- Labor is variable cost
- Assumes that labor is interchangeable
- Doesn’t work for software development

Metrics Have Consequences

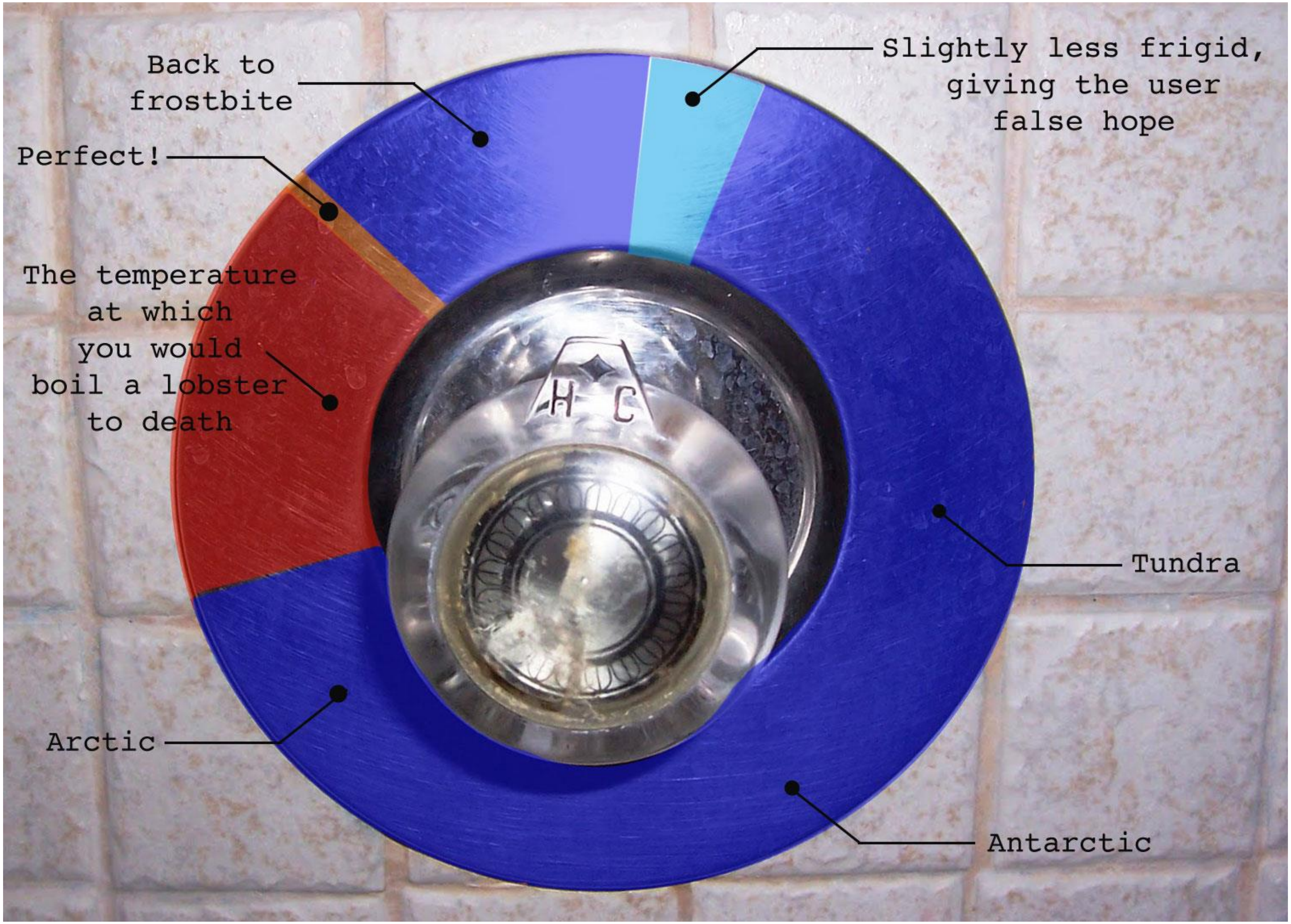
- Observer Effect
  - Measuring changes phenomena being measured
  - Observational studies
- Project metrics
  - Gathering metrics interferes with the process
  - Those being observed begin to "game" the system

Source: http://www.accuproav.com/securitycameras.html

Metrics Have Consequences

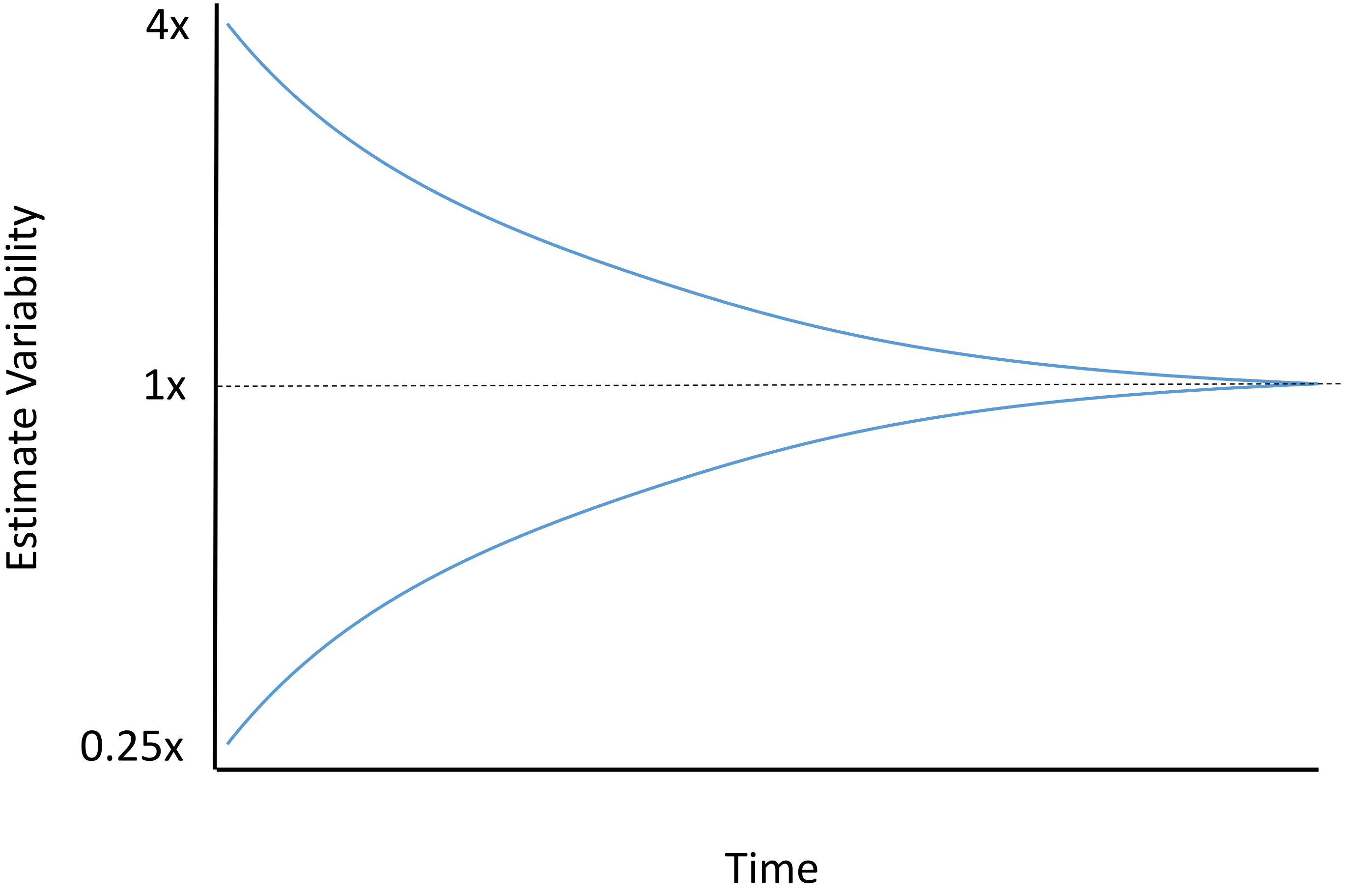
- Gathering metrics has a cost
    - Direct costs
    - Indirect cost
- Most metrics don’t provide much benefit in Agile
- Agile has better ways to gain similar knowledge
    - Retrospective – inspection and adaptation
- Measures that matter are:
    - Working software
    - Time-to-feedback

7. Feedback


Back to frostbite
Slightly less frigid, giving the user false hope
Perfect!
The temperature at which you would boil a lobster to death
H
C
Tundra
Arctic
Antarctic

Source: http://www.letterstobuffoons.com/wp-content/uploads/2012/09/ShowerHandle.jpg

Cone of Uncertainty



Original Source: Barry Boehm, Software Engineering Economics (1981)

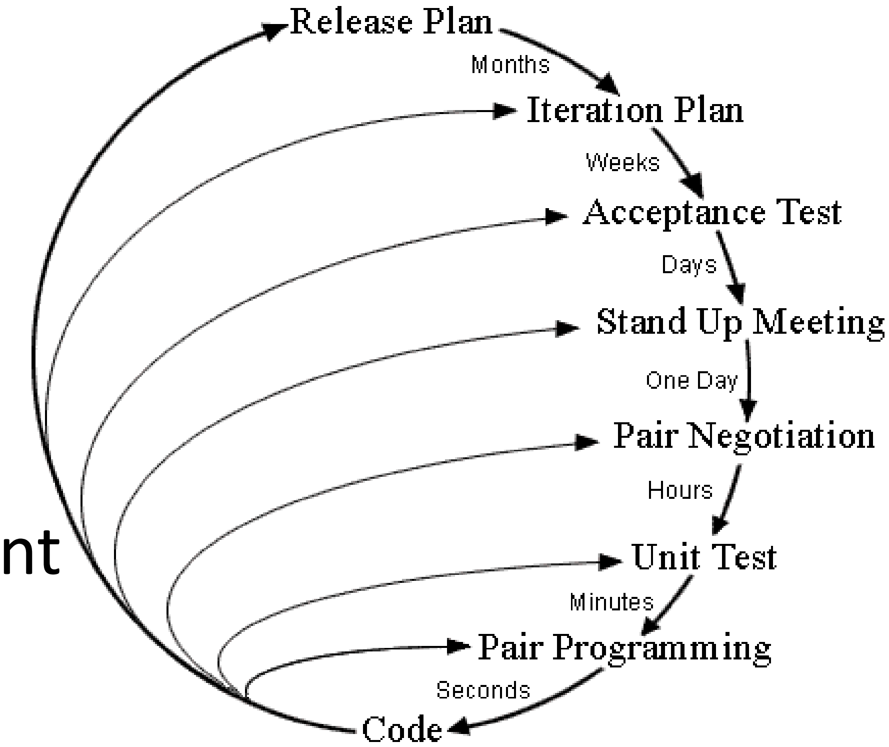
Feedback and Learning

- We reduce uncertainty by learning
- Feedback is necessary for learning
- Feedback is the most important factor in student success
- Continuous and rapid feedback allows us to learn more effectively

Source: http://www.icanhascheezburger.com

Agile Feedback

- Continuous and rapid feedback
- Short iterations
- Frequent releases
- Continuous integration
- Test-driven development



Source: http://www.agile-process.org/communicate.html

Smart Failure

- Short and frequent experiments
  - Lots of small failures
  - Lots of small successes
- Low cost and high value
- Old world vs. new world
- Requires mindset change
  - It’s ok to fail small
  - It’s ok to fail smart
  - However…

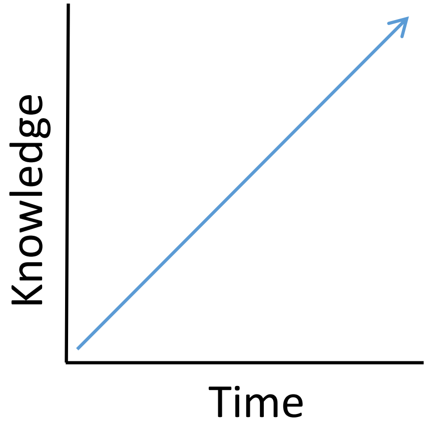
Source: http://craftfail.com/2011/08/cookie-monster-cupcake-fail/

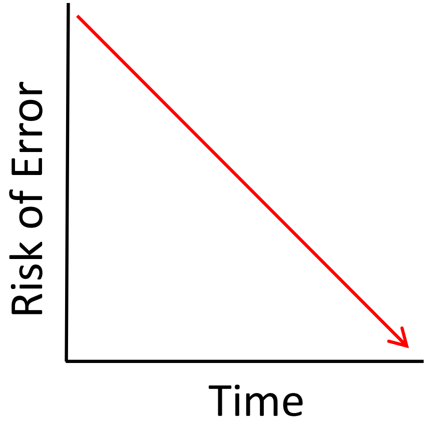
It’s Not OK to Fail BIG!

Source: http://t4toby.files.wordpress.com/2008/07/epicfail1.jpg/

Delay Decisions

- Delay decisions until the Last responsible moment
- Learn before making critical decisions
- Decide when more information is available
- Reduces risk of error
- May seem counterintuitive





Minimum Viable Product

- Primary Features
  - What users need
- Secondary Features
  - What users want
- Solve the core use case first
- Release an MVP
- Gather feedback
- Validate assumption

Minimum Viable Product

Product

Source: http://pandodaily.files.wordpress.com/2013/02/product_donuts-copy.jpg

Know When to Pivot

- Pivot = change direction
- When our assumptions are incorrect we pivot
- Pivot early, not late
- Minimize cost to pivot

Source: http://thesalespivot.com/wp-content/uploads/2011/07/left-turn-sign.jpg

Sunk Cost Fallacy

- Psychological bias when cutting losses
- Sunk cost
  - Previously incurred cost
  - Cannot be recovered
- Rational decisions
  - Don’t consider past loses
  - Only consider future returns

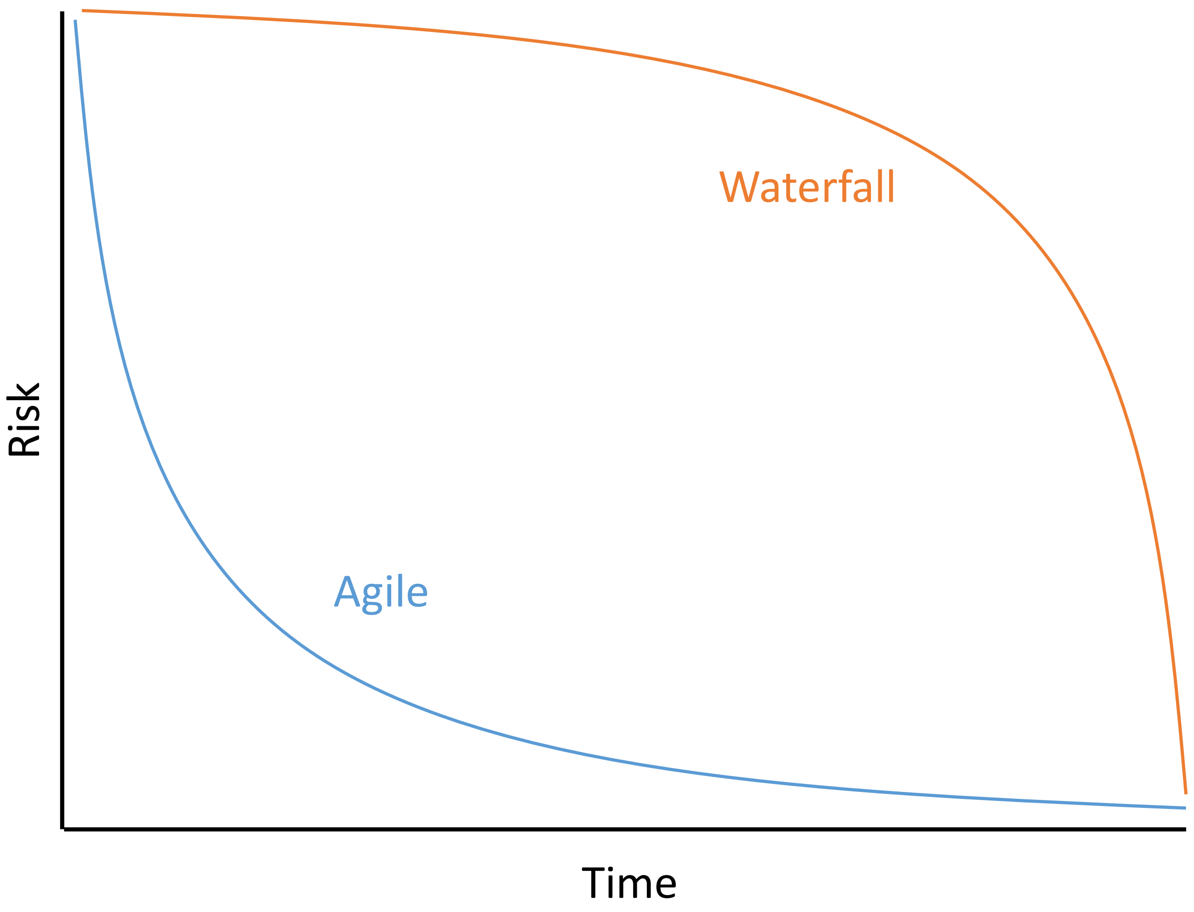
Source: http://www.amusingplanet.com/2010/04/fantastic-sinking-boat-by-julien.html

Sunk Cost Fallacy

- Humans are predictably irrational
- Avoid fallacy by:
  - Incremental steps
  - Feedback
  - Pivot early
- Stay minimally invested

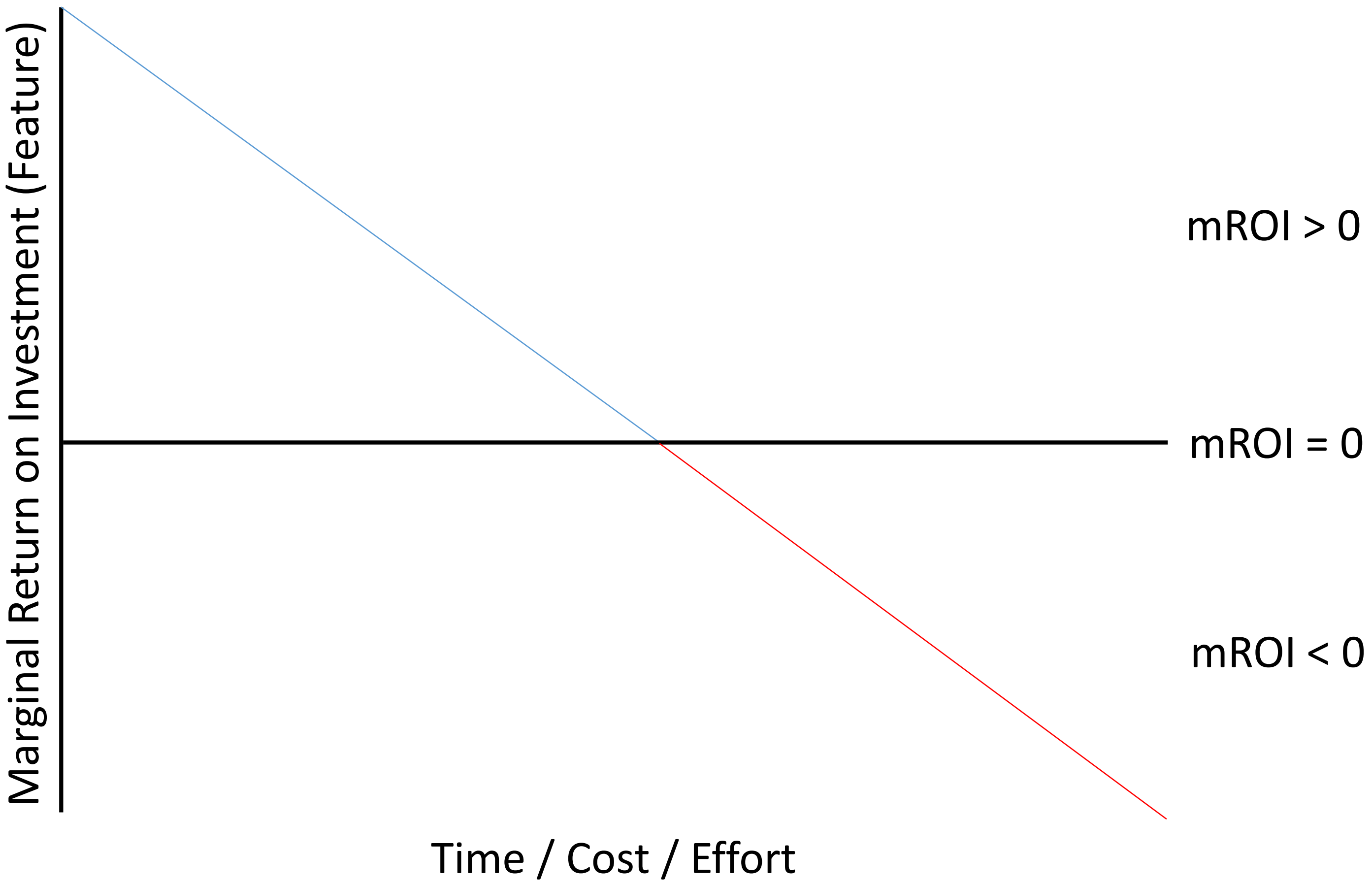
Source: http://www.amusingplanet.com/2010/04/fantastic-sinking-boat-by-julien.html

Risk



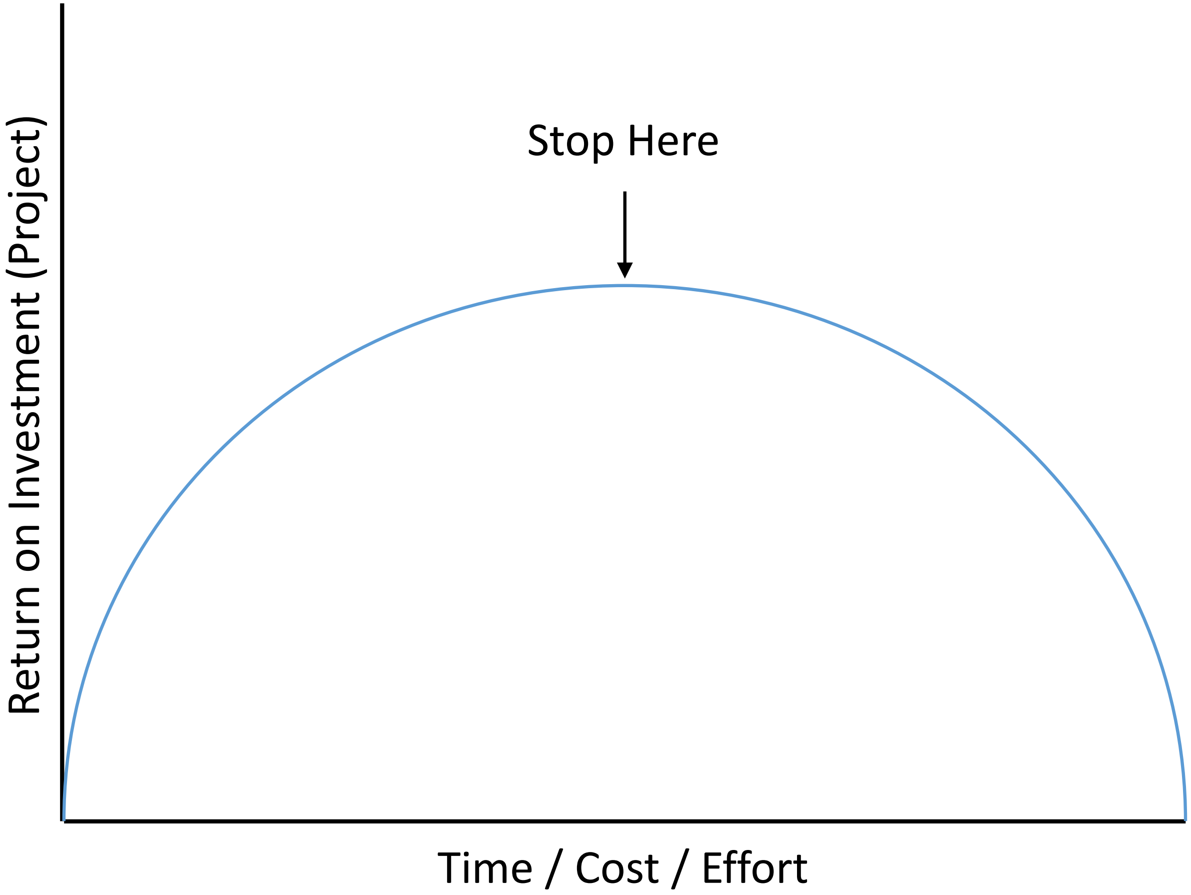
Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Know When to Stop



Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Know When to Stop



Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Know When to Stop

- Everything else:
  - Cost of Complexity
  - Eliminating Waste
  - Inventory Hides Problems
  - Overloading Bounded Contexts
  - Embracing Human Factors
  - Information Gain / Entropy
  - The Mythical Man Month
  - Kanban and Queuing Theory
  - TDD, Dopamine, and Crack
  - Sustainable Development
  - Agile is an Emergent Property
  - and much more…

Source: http://www.rounds.com/blog/wp-content/uploads/2010/11/stop-hammertime.png

Conclusion

Why is Agile so Successful?

1. It is well adapted to the world after midnight.
2. It inverts the project management constraints.
3. It prioritizes features by business value.
4. It embraces change verses following a plan.
5. It utilizes the efficiencies of self-organization.
6. It creates efficient communication.
7. It provides continuous and rapid feedback.

Feedback

- Was this valuable?
- What could I have done to make it better?

Contact Info

Matthew Renze

matthew@renzeconsulting.com

Renze Consulting

www.renzeconsulting.com