New Synergy Group

Systems Thinking... Enterprise Performance

Turbo-Charging Agile Software Development
with
Lean Methods and Systems Thinking

Technical Seminar
NY Spin Meeting
Microsoft
1290 Avenue of the Americas, NYC
6 September 2011

Dr. Satish Thatte

CEO, New Synergy Group

Kendall Park, NJ

732.410.9990

www.NewSynergyGroup.com

Agenda

- Challenges facing software industry
- Waterfall vs. Agile/Scrum
- Overview of Agile/Scrum Framework
- Journey from Waterfall/Ad hoc to Agile to Agile-Lean
- Agile/Scrum framework in Action
- Overview of Lean Framework
- Lean framework in action
- Overview of Systems thinking
- Synergies among Agile/Scrum, Lean and Systems Thinking

Challenges facing Software-Intensive Industry

- Failure to meet the real requirements of users
- Delays in time to market
- Cost overruns
- Low quality, productivity, or "Productivity x Quality"
- Failure to meet important goals:
  - Performance
  - Usability
  - Scalability
  - Security
  - Extensibility, etc.
- Failure to understand and resolve *organizational systems* issues

- These challenges *will* continue to be with us, as the bar is rising:
  - Ever more ambitious and complex software systems
  - Rising expectations of users
  - Fierce competition to deliver innovative solutions and services in ever shorter cycle times
  - Increasingly volatile business environment
  - Fast changing technology platforms and interoperability with legacy applications
- Little change in human nature!

Scrum Framework

- Develop complex products, while productively and creatively delivering highest possible value.
- Employ various processes, practices and techniques within the framework that suits your organization.
- The framework: Scrum Team, and **3 roles, 3 artifacts, 5 events**, and many rules that bind them.
- Based on empirical process control: Transparency, Inspection, Adaptation
  - **Lightweight process**
  - **Simple to understand:** roles, artifacts, events and rules
  - **Extremely difficult to master:** How to use customized processes, practices, and techniques consistently throughout an organization; How not to be a *Scrumbut*; How to do continuous improvements; Scalability

Three Roles in Scrum Team

- ScrumMaster
- Product Owner
- Development Team

Three Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Five Scrum Events

- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review
- Sprint Retrospective

The Scrum Guide, July 2011

Example: Smart Phone Product

- **Basic phone operations:** Make call, Receive call
- **Set-Up:** Sound, Display, Ring tones, Blue Tooth
- **Contact Management:** New, List, Group, Search
- **Call Log Management:** Dialed, Received, Missed, All
- **Messaging:** NewText, InBox, Voice Mail, MobileIM, Chat
- **Tools:** Volume, Calendar, WorldClock, AlarmClock, Notes
- **Music:** Play Lists, Artists, Genre, Album, Settings & controls
- **Camera:** Picture, Album, Settings

Waterfall vs. Scrum

Sequential Engineering

Waterfall (Release 1.0): 6 months .. delay .. 9 months			
Analysis	**Design**	**Code**	**Test**
Basic phone Set-Up Tools	Basic phone Set-Up Tools	Basic phone Set-Up Tools	Basic phone Set-Up Tools

Concurrent Engineering

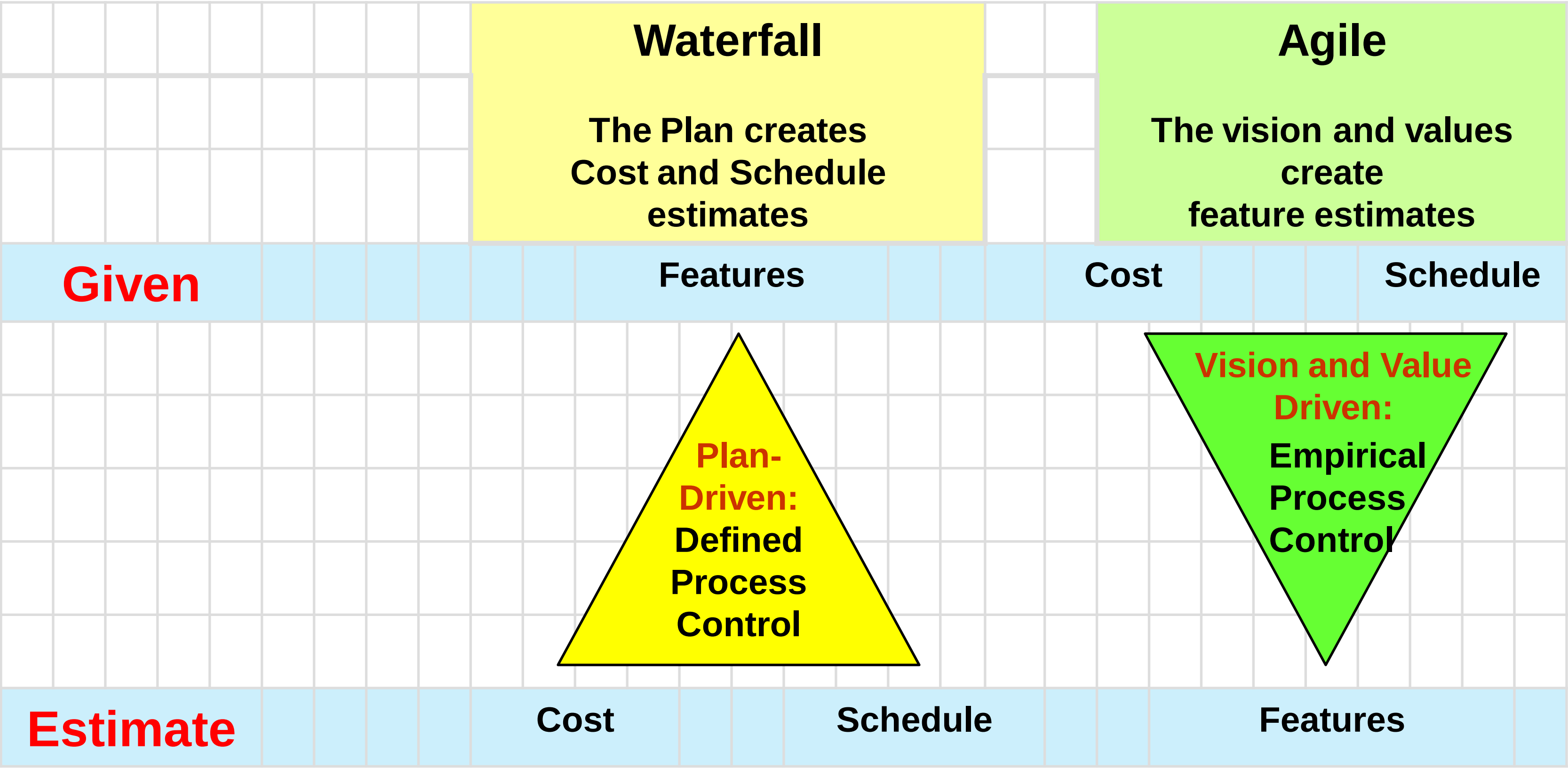
Scrum (Release 1.0)				
Sprint 0 Archi-tecture Framework	**Sprint 1** Basic phone	**Sprint 2** Set-up	**Sprint 3** Tool Set	**Hardening Sprint 4**
Analysis	**Analysis**	**Analysis**	**Analysis**	**Analysis**
Design	**Design**	**Design**	**Design**	**Design**
Code	**Code**	**Code**	**Code**	**Code**
Test	**Test**	**Test**	**Test**	**Test**

Consider feedback from each sprint to the next sprint

Waterfall development depends on a perfect understanding of requirements at the outset and minimal errors in each phase.

Scrum blends all development activities into each sprint, adapting to discovered realities.

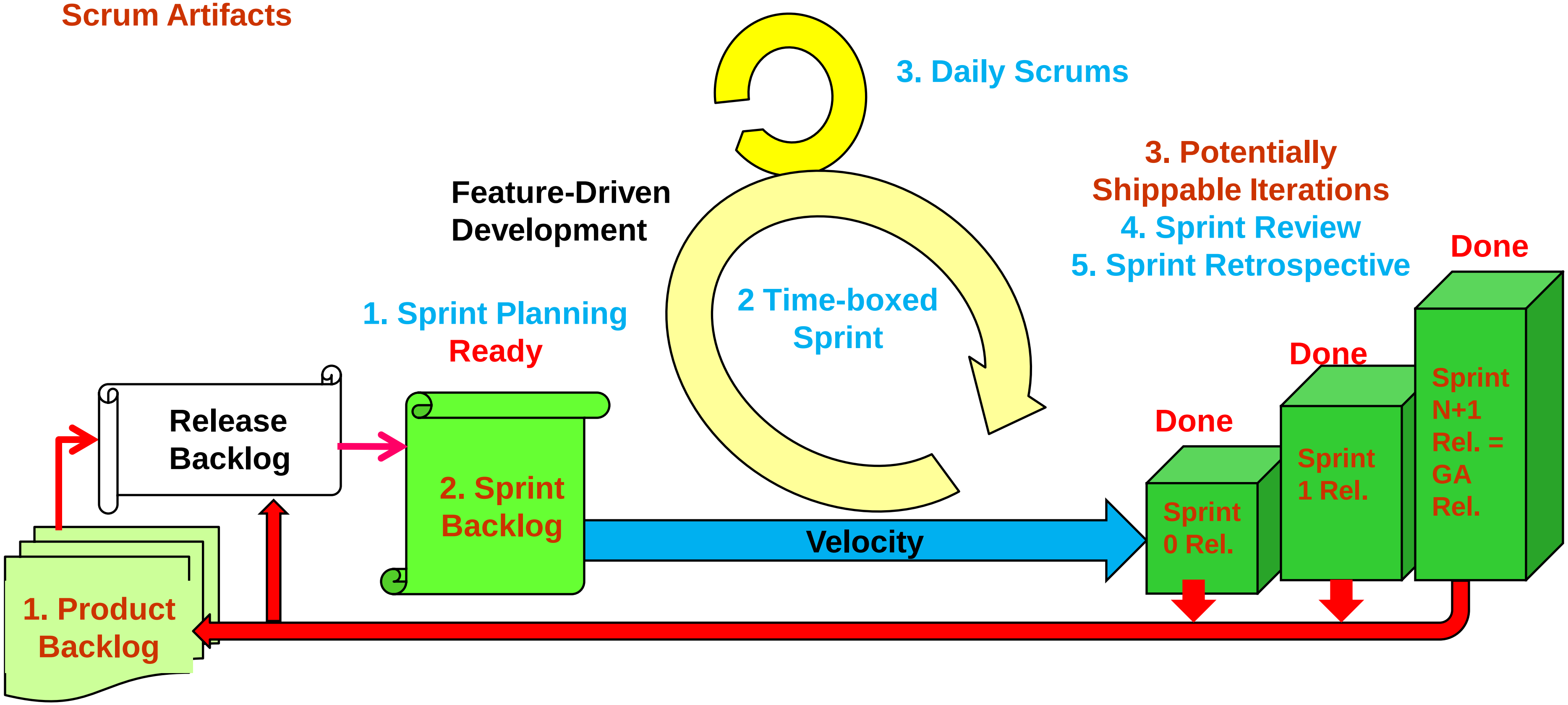
Waterfall vs. Agile Projects



	Waterfall The Plan creates Cost and Schedule estimates	Agile The vision and values create feature estimates
Given	Features	Cost Schedule
	Plan-Driven: Defined Process Control	Vision and Value Driven: Empirical Process Control
Estimate	Cost Schedule	Features

Scrum Framework: In Action

Scrum Events
Scrum Artifacts



Scrum Roles and Responsibilities

Product owner: responsible for "*what*"

- Defines the right product or solution.
- Represents the customer needs.
- Maximizes business value.
- Prioritizes and clarifies requirements.
- Actively manages the product, release and sprint backlogs.
- Typically spends 50% time with the team, and 50% with customers.
- Is not the boss of Scrum team.

ScrumMaster: responsible for "*how*"

- Owns the Agile/Scrum process.
- Serves the team (servant leader).
- Removes impediments.
- Coaches team in Scrum framework and practices.
- Facilitates Sprint planning, Daily Scrum, Sprint Review and Retrospective
- Helps the team make timely decisions.
- Is not the boss of Scrum team.

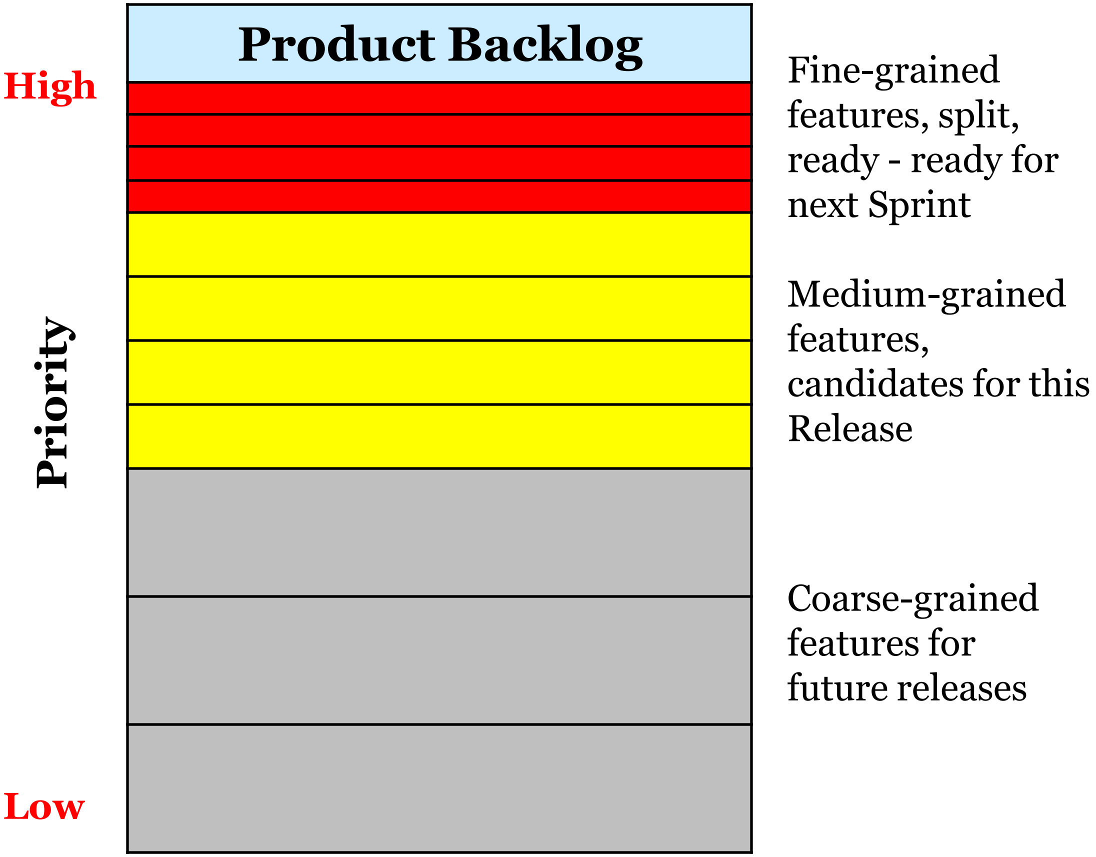
Development team: Cross functional, self-organized team with 7+/- 2 members

Product Backlog Items is DEEP; DRiVE Carefully

- **D**etailed appropriately
- **E**stimated
- **E**mergent
- **P**rioritized

Product backlog items are linearly prioritized based on the **DRiVE** criteria:

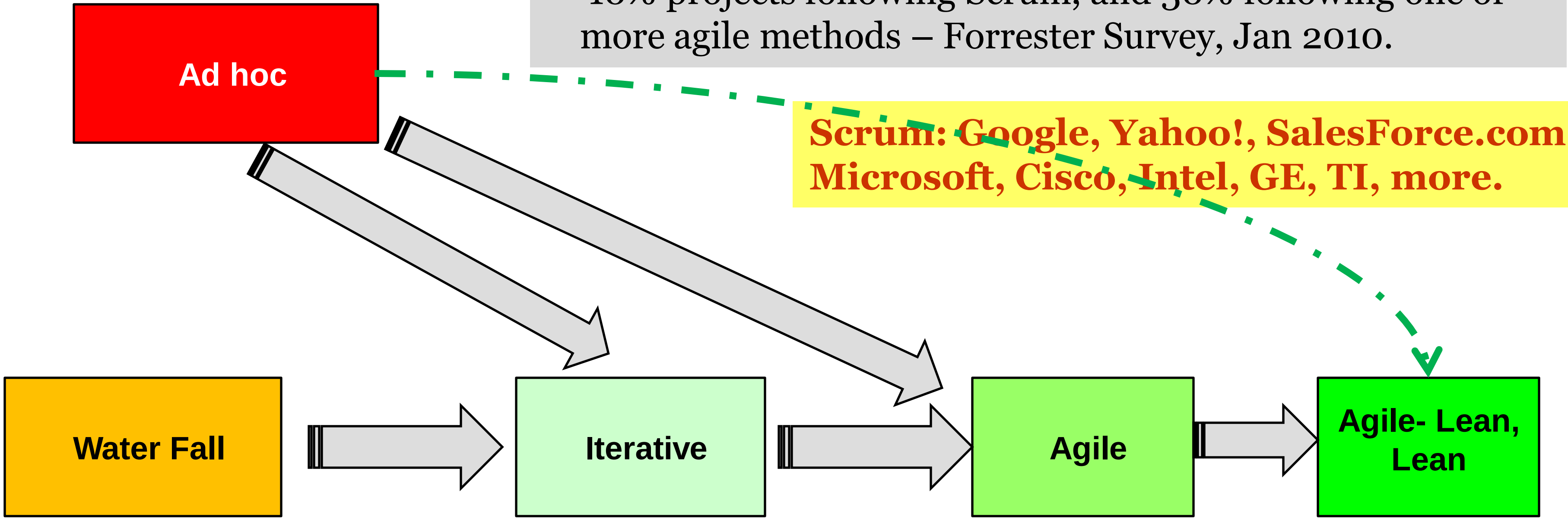
- **D**ependencies
- **Ri**sks: Business and technical
- Business **V**alue
- Estimated **E**ffort


Product Backlog
High
Priority
Low
Fine-grained features, split, ready - ready for next Sprint
Medium-grained features, candidates for this Release
Coarse-grained features for future releases

Journey to Agile to Agile-Lean

Scrum is the market and mindshare leader

- Simple framework is extensible and customizable.
- 10% projects following Scrum, and 30% following one or more agile methods – Forrester Survey, Jan 2010.



Scrum: Google, Yahoo!, SalesForce.com, Microsoft, Cisco, Intel, GE, TI, more.

Lean, Agile-Lean: Amdocs, BBC Worldwide, CitySearch, Corbis, Facebook, NBC Universal, Premier Healthcare

Lean Framework

- ***Lean*** is the English term given by MIT researchers to describe the system of the *Toyota Way*

- Refers to *Lean production*, as opposed to *Mass production*
- Viewed as a set of operating principles and policies, not as a prescriptive step-by-step methodology.
- Does not mean “lean and mean” – or fire and downsize

Lean Principles & Policies

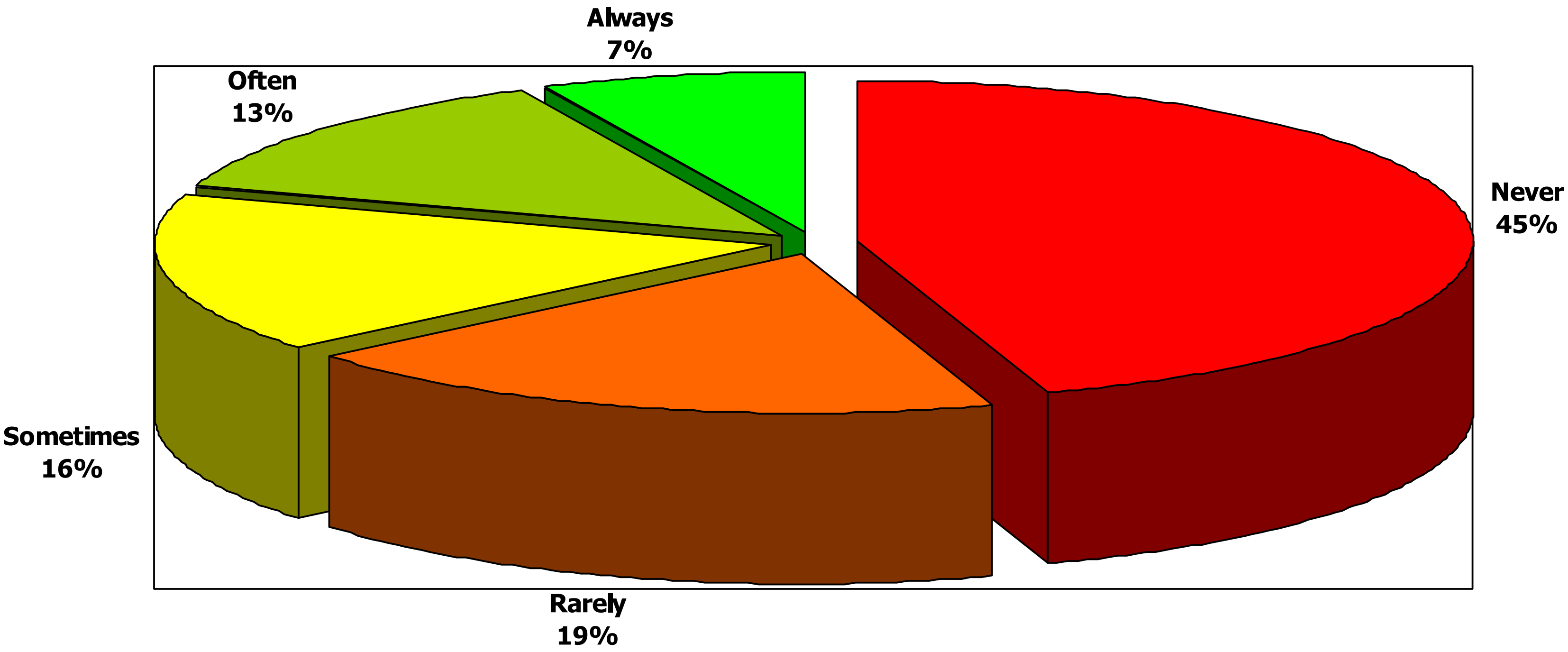
- Minimize waste
- Obliterate/reduce queues
- Increase the value flow
- Pull management
- Level the work
- Kanban – Visual signaling
- Kaizen – Continuous improvement
- Find and eliminate the root cause

Value Ratio, and its Shocking Truth!

- **Value:** The time periods while developing a product that the customer is willing to pay for. Value is in the eyes of (external) customer.
- **Waste:** All other time periods that do not add value, but take resources
  - Multi-tasking, context-switching workers
  - Waiting in queues
  - Bottlenecks: single experts pulled for every crisis; narrow specializations
  - Hand-offs (in sequential waterfall process)
  - Undo-Redo cycles (due to poor communication, poor understanding, etc.)
- **Value Ratio** = Value time / Lead time (concept to consumption time)
  - Value ratio is less than 7% in many product development organizations! [Source: Larman & Vodde, 2008; Poppendiek, 2006]
- Value-adding moments (< 7%), and huge amounts of waste (> 93%)
  - Improvement strategy needs to be subtractive and not additive
- Requires mindset change and need to develop "Eyes for Waste"

"There is nothing so useless as doing efficiently that which should not be done at all." -- Peter Drucker

Most Products are Wastefully Over-Engineered



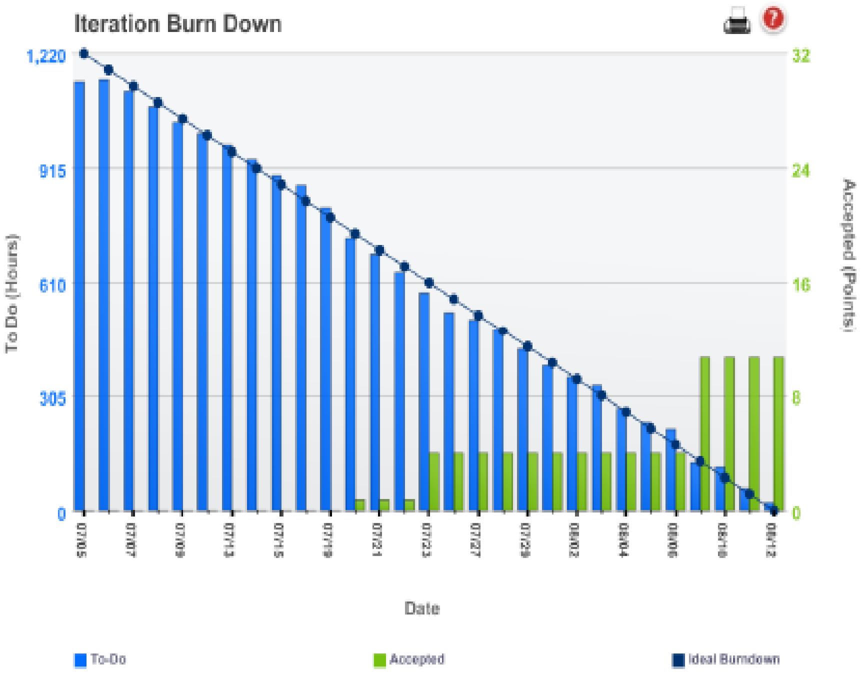
Source: Jim Johnson of the Standish Group, Keynote Speech XP 2002

Product Development: Flow Management

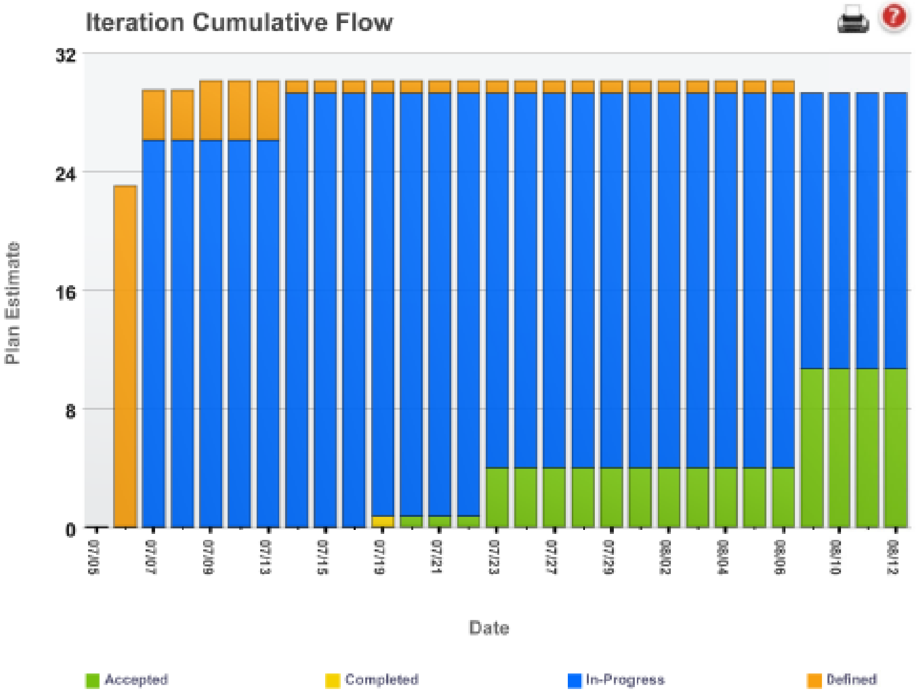
- Make value flow without delay to the customer
  - Reliable cadence of delivering sprints every n weeks
  - Limit Queues and Work in Progress (WIPs)
  - Minimize backflows (re-work trying to clean up the mess)
- Flow is a perfection challenge: probably never achieved 100%; but it is a very important goal to strive
  - Single focus, no multi-tasking
  - Zero waste by reducing batch size, variability, WIPs, queues and delays, and other wastes

Scrum Team 1: No Emphasis on Lean

Burn-Down, Burn-Up Charts

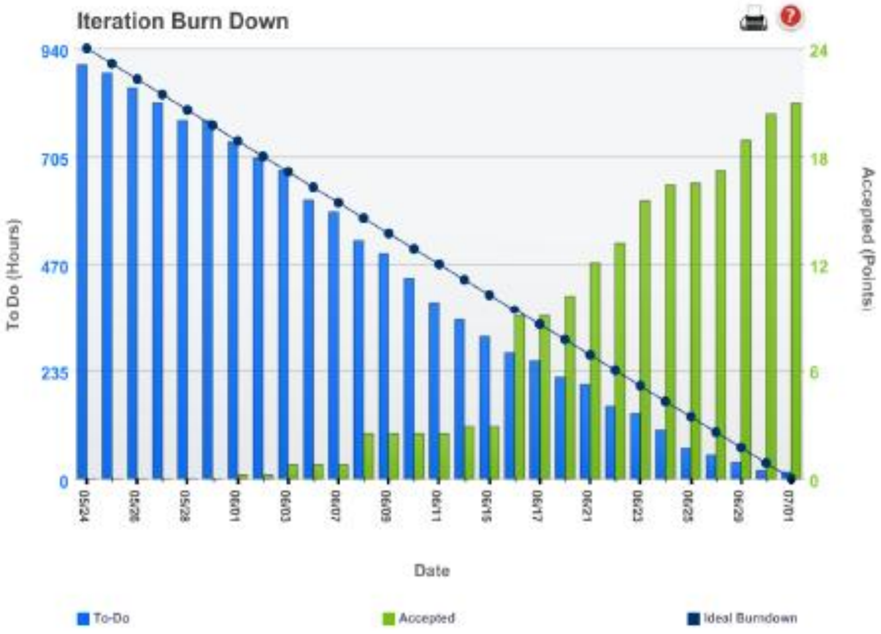


Sprint Cumulative Flow

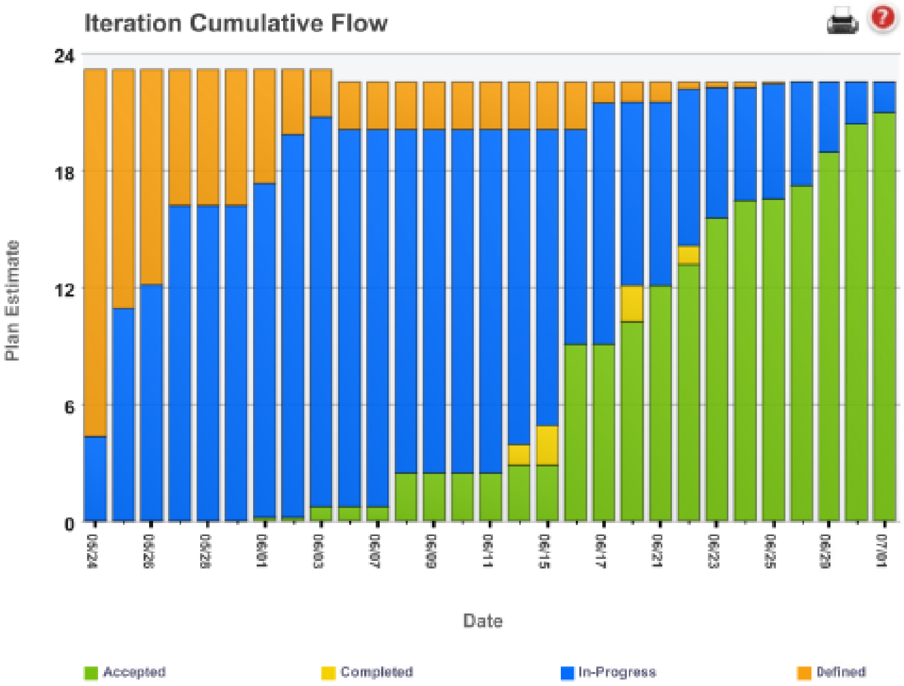


Scrum Team 2: Some Emphasis on Lean Methods

Burn-Down, Burn-Up Charts



Sprint Cumulative Flow



Kanban Agile Development Flow: Set WIP Limits

WIP	Queue	WIP	Queue	WIP, 1	Queue	WIP, 1	WIP
				2		2	
			1	3		3	
1		1	2	4		4	1
2	1	2	3	5	1	5	2
3	2	3	4	6	2	6	3
Features with Design in Progress	Features with Design Done	Features with Development in Progress	Features with Development Done	Features with Test in Progress	Feature s with Test Done	Features with Defects fixing in Progress	Features with Defects Verification Progress

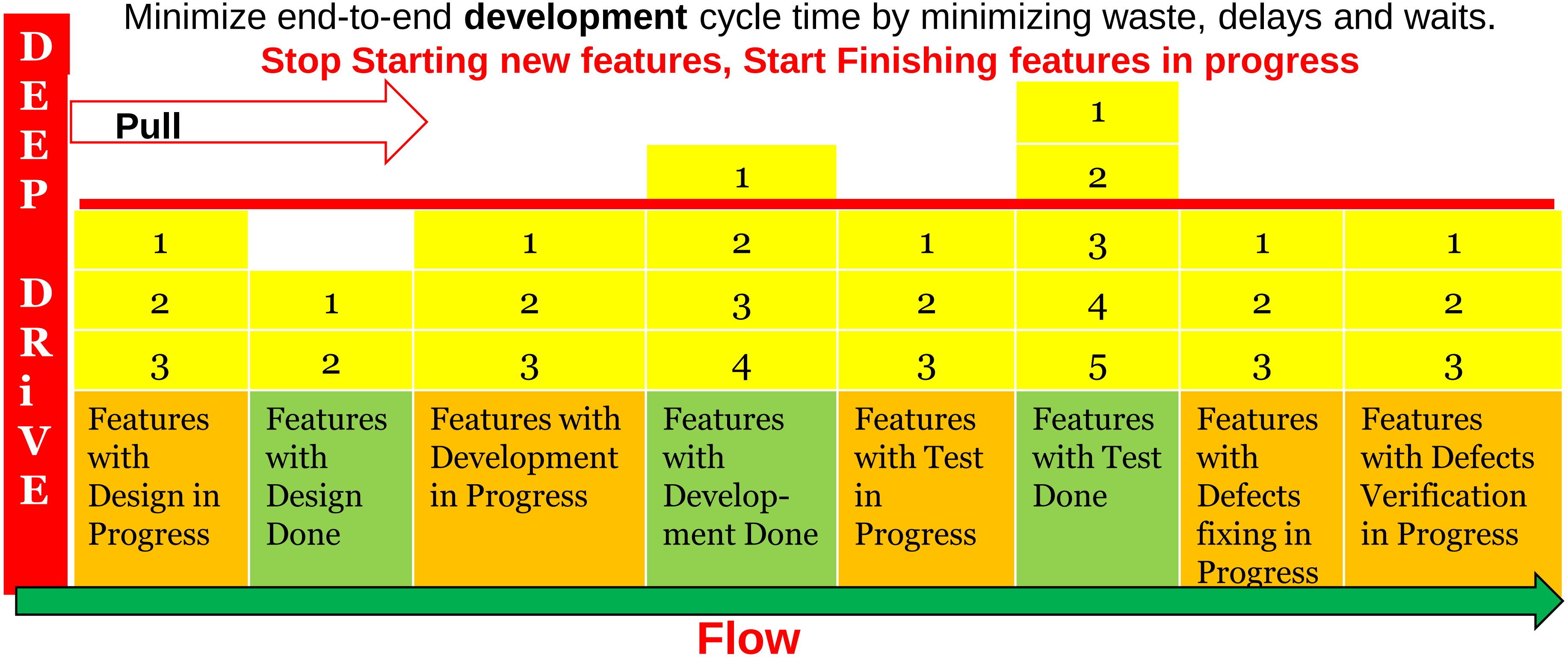
Flow

- Cross-functional team smoothens the flow based on WIP Limit = 3 (in this example)
- Address bottlenecks: More resources for testers to test, and for developers to fix defects

Kanban Agile Development Flow: Set WIP Limits

Minimize end-to-end **development** cycle time by minimizing waste, delays and waits.

Stop Starting new features, Start Finishing features in progress



					1		
			1		2		
1		1	2	1	3	1	1
2	1	2	3	2	4	2	2
3	2	3	4	3	5	3	3
Features with Design in Progress	Features with Design Done	Features with Development in Progress	Features with Develop-ment Done	Features with Test in Progress	Features with Test Done	Features with Defects fixing in Progress	Features with Defects Verification in Progress

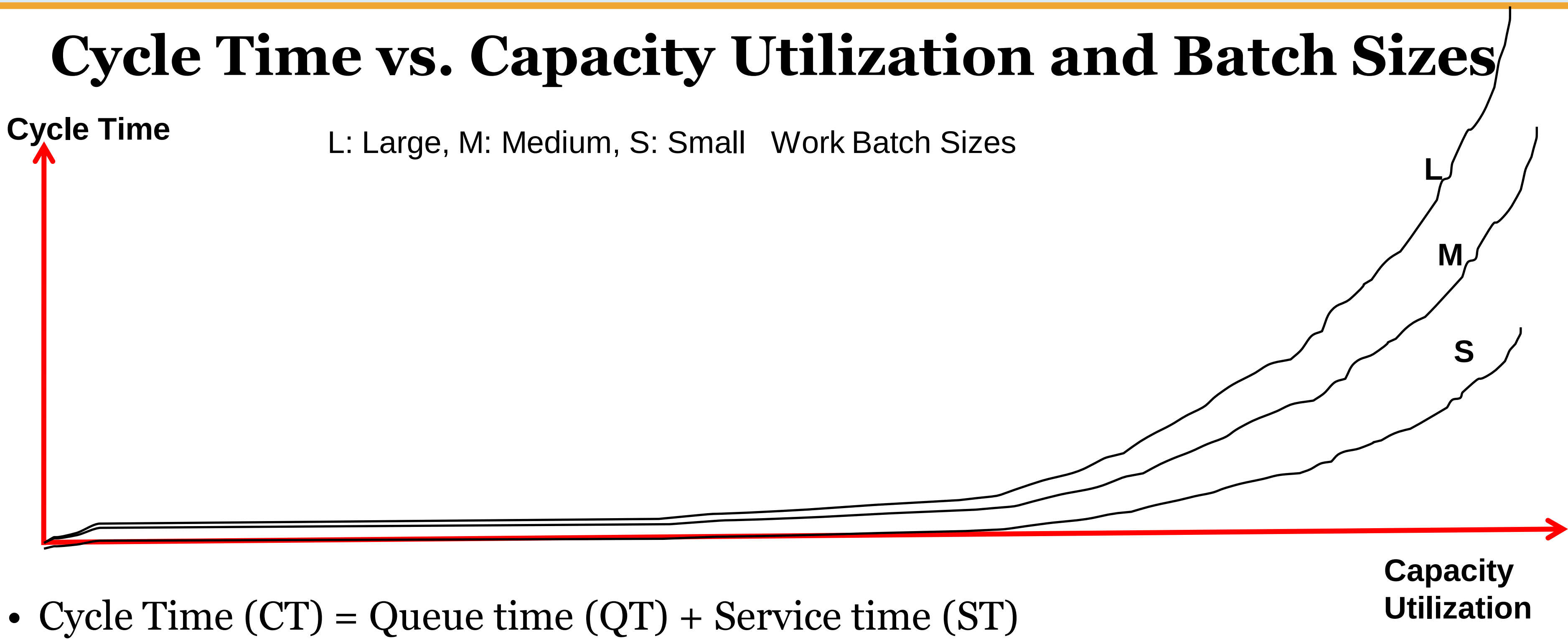
- Combine Scrum's DEEP DRiVE sprint backlog with *lean* flow of value
- Avoid Kanban starvation: Empty Queues or Empty WIPs

Pull Management

- Work done only in response to a pull signal from a downstream process step or work initiated by an executive sponsor
- Scrum team pulls sufficiently detailed fine-grained features for the current sprint from the sprint backlog
- The business team (Product Owner, Business Analyst) is working one time box ahead in its own flow so items are always in a ready state to be pulled from the prioritized backlog.
- “Stop-and-fix-root-cause” mindset allows you to fix the root cause quickly while things are still fresh in your mind

It is easier and smarter to pull a thread, not to push it!

Cycle Time vs. Capacity Utilization and Batch Sizes



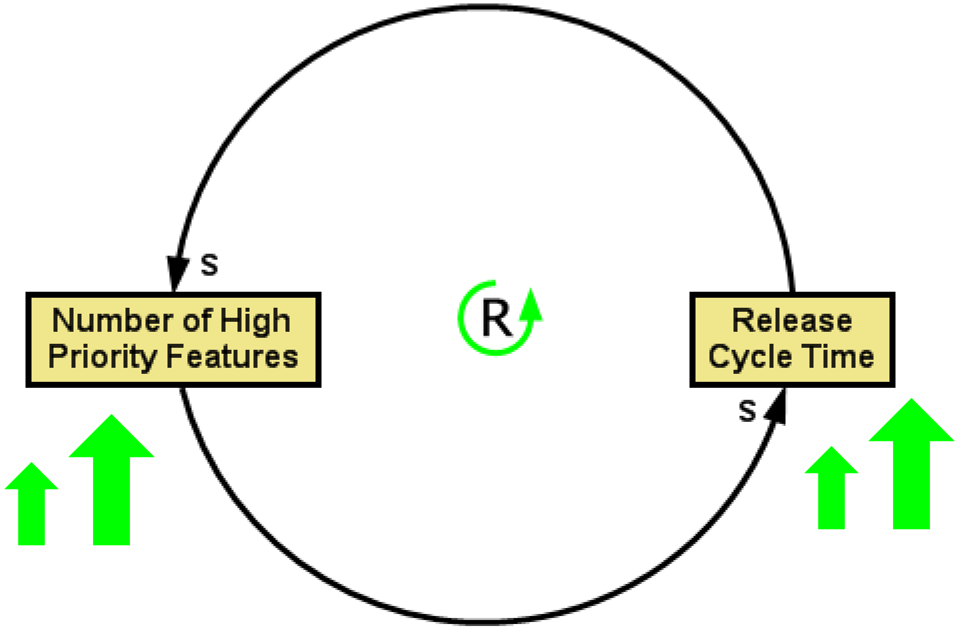
- Cycle Time (CT) = Queue time (QT) + Service time (ST)
- As capacity utilization or batch size increase, queue time increases non-linearly
- Decompose large features into smaller sub-features hierarchy; at the leaf level, no feature taking more than ¼ N staff-weeks of effort for N-week sprint

Introduction to Systems Thinking

- **System**: interacting components or subsystems forming an integrated whole.
  - **Purpose or function:** Usually deduced by observing system behavior, involving inputs (information, material or energy), processing, and outputs (information or material or energy).
  - **Components:** People, principles and values, processes and workflows
  - **Interactions:** Functional and structural relationships among components allowing them to interact
  - **Structure:** Defined by components and their interconnections
- A **model** is used to represent a system, capturing only the key aspects
  - **Causal Loop Diagrams (CLDs):** Great for qualitative understanding and reaching consensus; they cannot be simulated
  - **Stock-and-Flow Networks (SFNs):** Can be simulated for quantitative analysis and *what-if* experiments, but require more effort to build and validate

Note: We are *not* talking about only software system to be developed, but *systems* in general…specifically *organizational* systems.

CLD for Relationship between Release Cycle Time and Number of High Priority Features



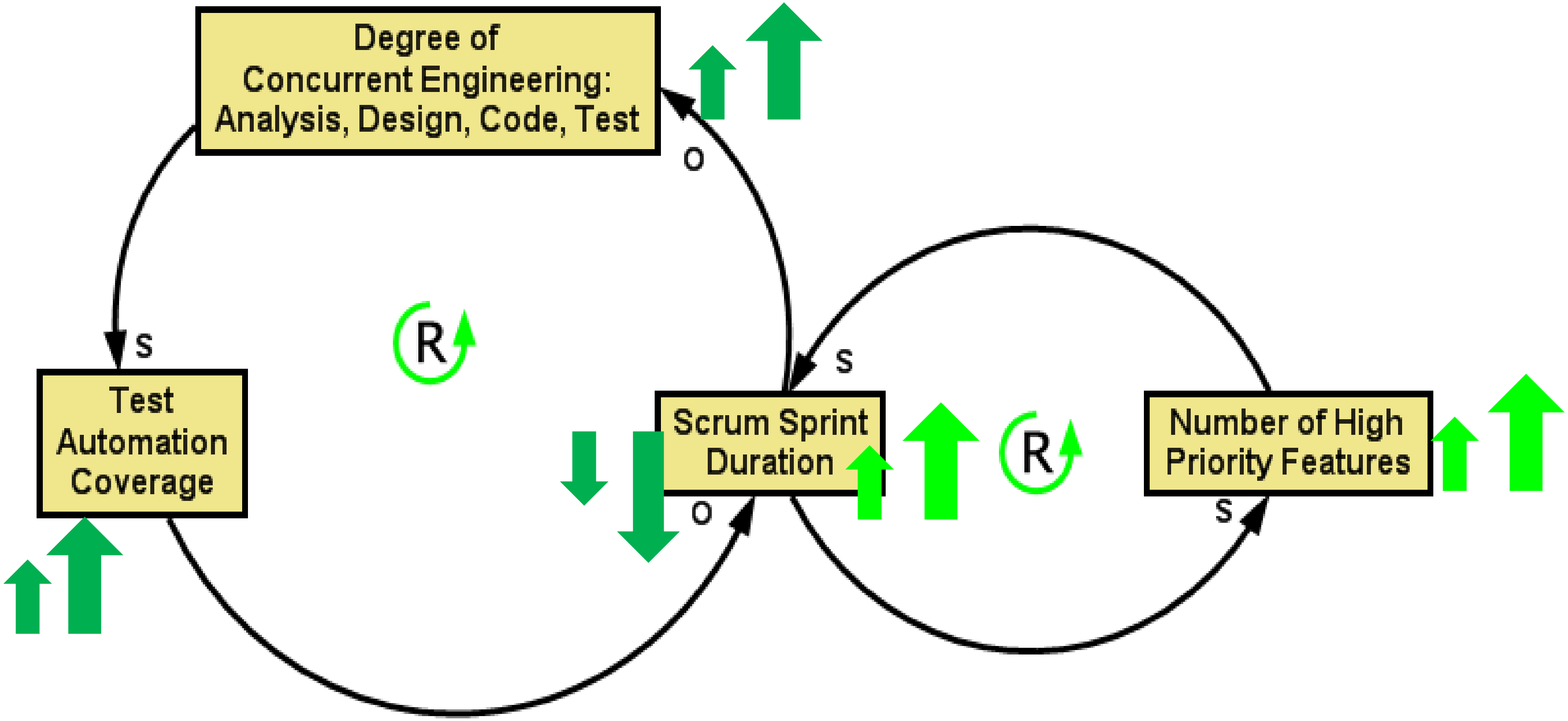
R: Reinforcing (positive) feedback loop
B: Balancing (negative) feedback loop

Notation

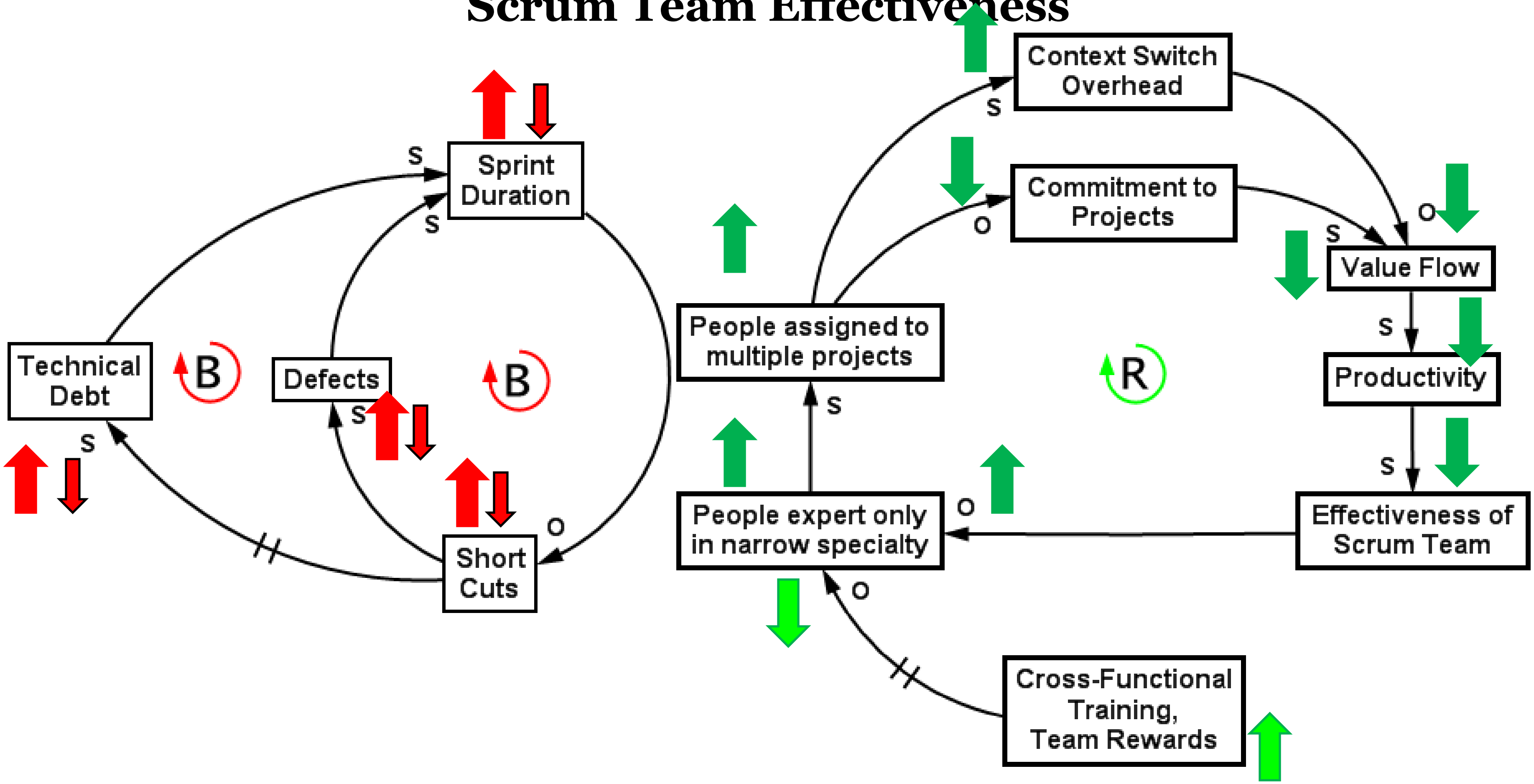
Cause and **Effect** variables are shown in rectangular boxes, with links showing causal relationships

- **S** (Supports): If the cause **increases**, the effect **increases** *above what it would otherwise have been*, and if the cause **decreases**, the effect **decreases** *below what it would otherwise have been.*
- **O** (Opposes): If the cause **increases**, the effect **decreases** *below what it would otherwise have been*, and if the cause **decreases**, the effect **increases** *above what it would otherwise have been.*

CLD for Relationship among Sprint Duration, Test Automation, Number of High Priority Features


Degree of Concurrent Engineering: Analysis, Design, Code, Test
O
S
R
Test Automation Coverage
S
Scrum Sprint Duration
O
R
Number of High Priority Features
S

CLDs: Short-cuts and Sprint Duration, Scrum Team Effectiveness


Context Switch Overhead
Sprint Duration
S
S
S
Commitment to Projects
O
S
O
Value Flow
Technical Debt
B
Defects
S
B
People assigned to multiple projects
S
R
S
Productivity
S
S
O
Short Cuts
People expert only in narrow specialty
O
Effectiveness of Scrum Team
O
Cross-Functional Training, Team Rewards

Wisdom from Systems Thinking

- **Faster is slower; slower is faster: Rush job → defects, delays**
  - Work in sustainable steady sprints…avoid death march
- **There are cause →effect → cause feedback loops with possible delays…avoid causation fallacy**
  - Every effect has a cause… but don’t assume you can tell which is which!
  - *5-Why’s* method for root cause analysis may not always work
- **Optimize the whole system, not sub-optimize its parts**
  - Optimize the total cost of (Anticipatory design + Adaptive design)
  - Optimize the total cost of (Defect prevention + defect correction)
- **Cause and effect are often far removed in space and time**
- **Avoid short-termism**
  - Short-term pain vs. long-term gain | Short-term gain vs. long-term pain
  - Today’s problems often come from yesterday’s “solutions”

Reference: “Thinking in Systems: A Primer” by Donella Meadows

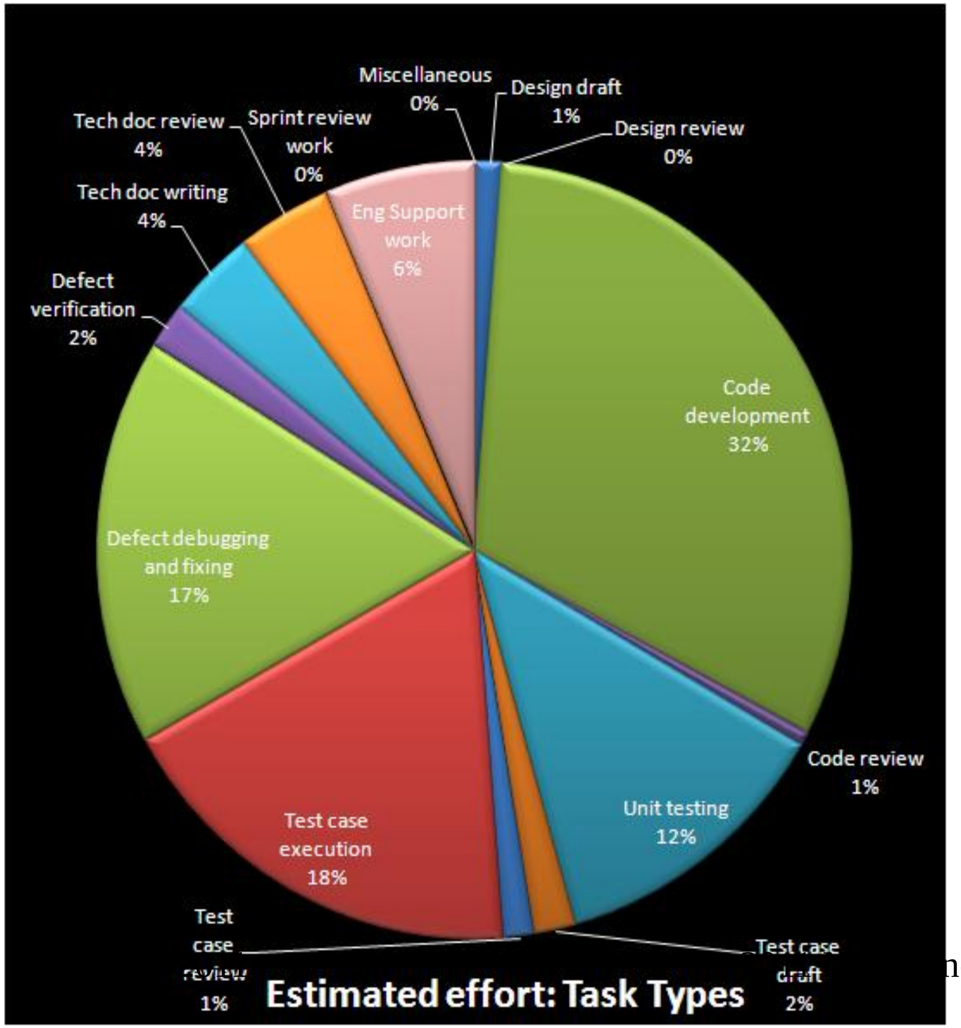
Other Major Application Areas of Systems Thinking to Agile-Lean Development and Beyond

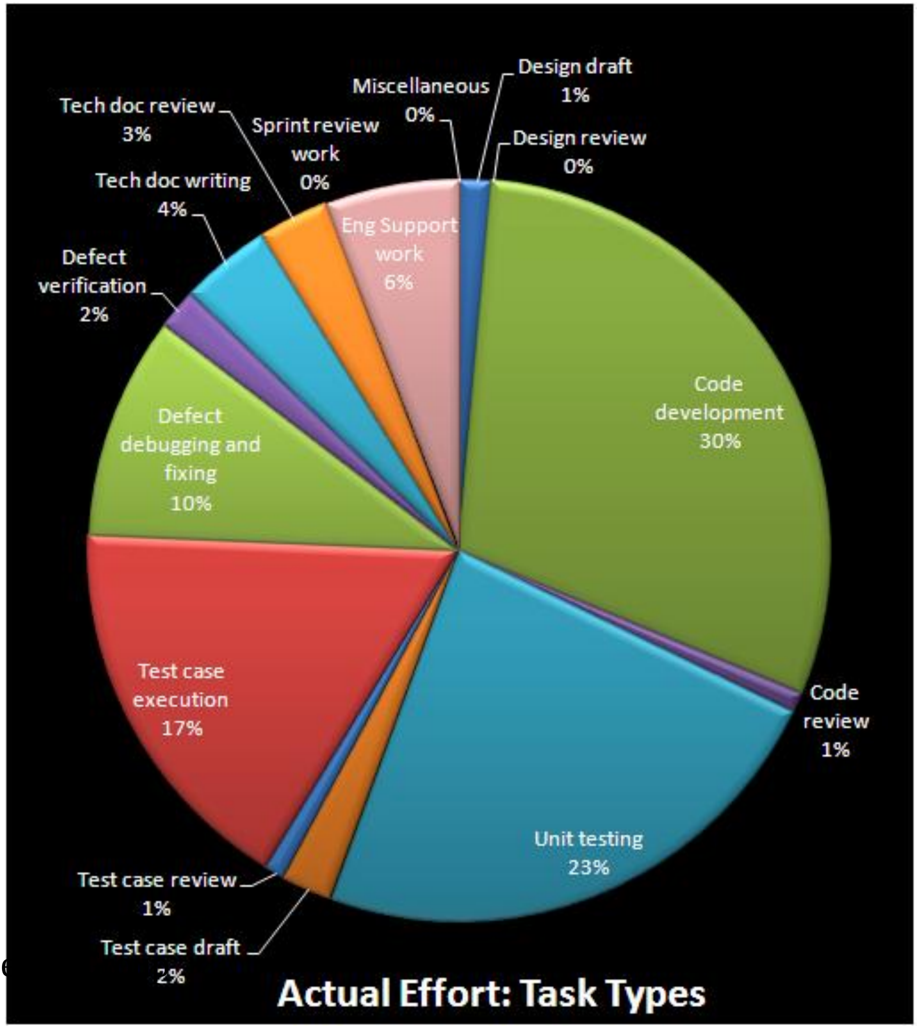
- **Use standardized practices and instrumented templates**
  - Ensure consistency across different Scrum teams and across multiple sprints, with minimal overhead for compliance
  - Provide timely feedback for corrective actions and process improvement
- **Make agile and lean methods work well with broader organizational processes for commercial success**
  - Operations: Product development lifecycle, Manufacturing, Marketing, Customer Support, Sales, Legal, Alliances, etc.
  - Business strategy and operations must be aligned
- **Change the *organizational system* to change *culture***
  - *A bad system, will defeat a good person, every time.* – Edward Deming
  - *Changing the system, will change what people do. Changing what people do, will NOT change the system.* – Peter Scholtes

Instrumented Templates for Agile/Scrum Framework

Product Owner	ScrumMaster/ Project Manager	Domain & Feature Analysis	Design & Development	QA
Biz value & Biz risks estimation[1]	Tech effort & Tech risks estimation[1]	Problem domain analysis [4]	Feature-driven Development[4]	Defects backlog management[3]
Product backlog prioritization[1]	Daily Scrums[1]	Feature specs[3]; and Non-functional requirements specs[3] divided into tasks: • Design • Code development • Code review • Unit testing • Test case develop. • Feature testing • Defect fixing • Defect verification • User documents • Acceptance testing	Architecture Sprint-0[4]	**Layered testing**[3] Regression Tests, System tests, Acceptance tests, Beta tests
Sprint reviews[2]	Impediment log[1]		Agile User documents[2]	
Release plan and data sheet[2]	Capacity _Workload_ Availability_Tracking[1]			
Product vision[2]	Steady pace of sprint[1] Weekly Mini-Sprints[1]		**Legend** **• 1: Spreadsheet** **• 2: Word** **• 3: Agile tool** **• 4: White board, camera**	
Elevator pitch[2]; Product roadmap[2]; Trade review [2]	Ready-Ready[3]; Done-Done[3]; Sprint retrospectives[1]			

Instrumented Template for Sprint Effort

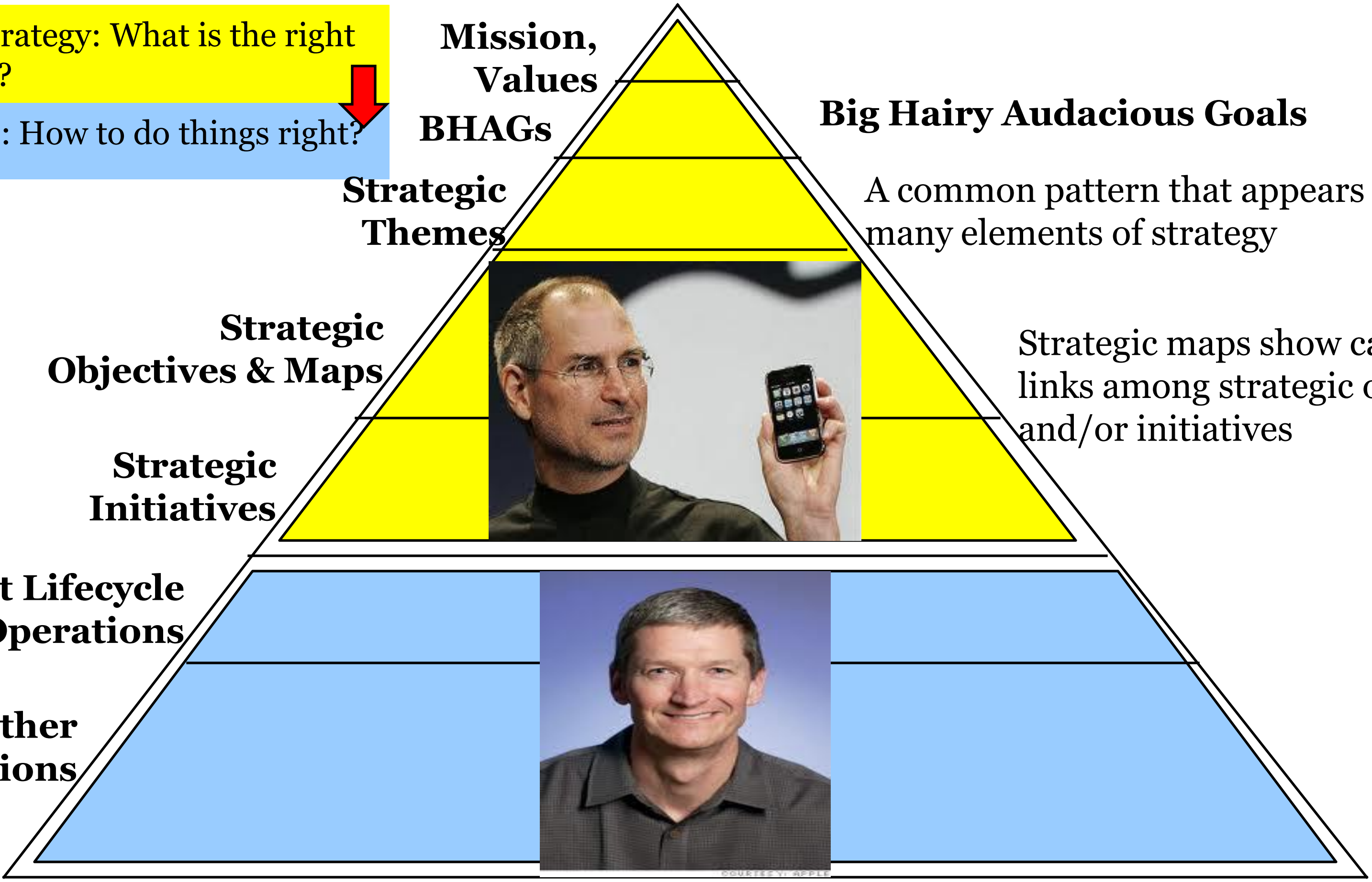




Business Strategy and Operations

Business strategy: What is the right thing to do?

Operations: How to do things right?



Mission, Values

BHAGs

Big Hairy Audacious Goals

Strategic Themes

A common pattern that appears across many elements of strategy

Strategic Objectives & Maps

Strategic maps show cause-effect links among strategic objectives and/or initiatives

Strategic Initiatives

Product Lifecycle Operations

All Other Operations

Synergies among Agile/Scrum, Lean & Systems Thinking

	Agile/Scrum	Lean, Systems Thinking
Time-boxing	Yes	No
Reduce end-to-end feature cycle time	No	Yes; Level features to reduce avg. cycle time; *Stop Starting, Start Finishing* mantra
Value Flow	No	Yes
Visibility	Burn charts, Impediment log	Cumulative flow, Visible queues, Kanban board
Optimization and scalability	Automation, Refactoring, Scrum of Scrum, Meta Scrum	System of systems (systems thinking); Optimize the *whole system* (not parts)
Control mechanism	Daily Scrums, Sprint review and retrospective	WIP limits, Queue reduction
Validation and Verification	Acceptance test as part of feature spec, Test-driven development	Reduce the total cost of (Defect prevention + Defect detection and correction); Find the root cause and fix it (*5-Why's* may not always work)

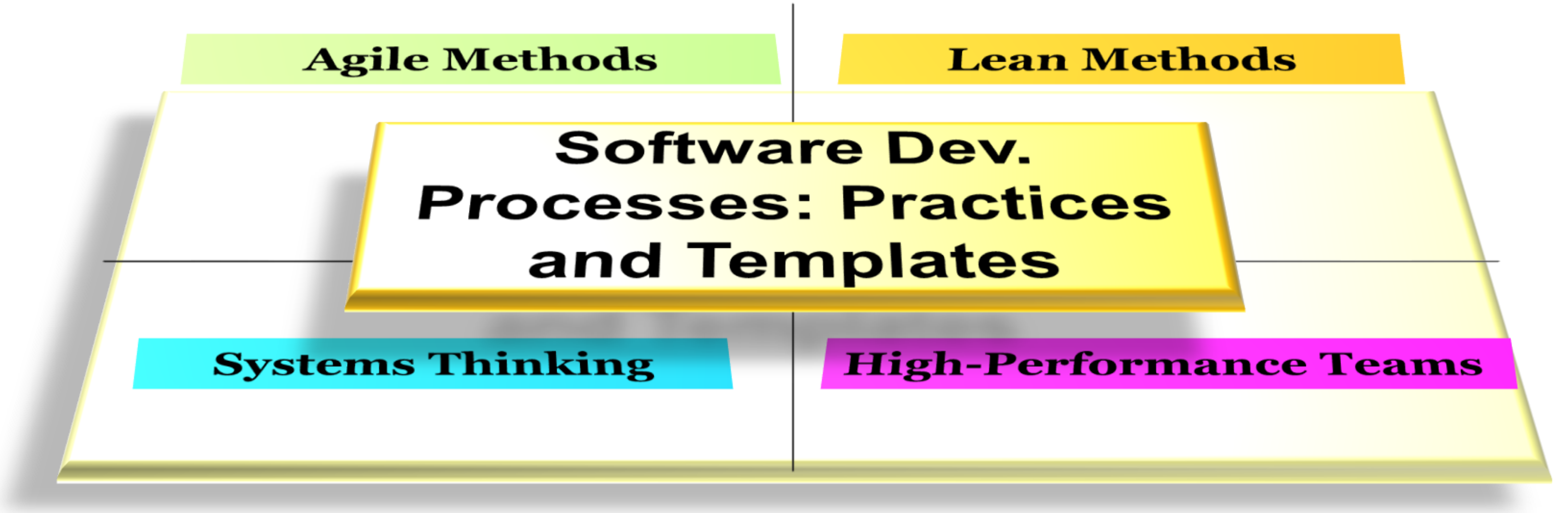
New Synergy Group

Systems Thinking... Enterprise Performance

Software Development Processes Turbo-Charged with:

Agile and Lean Frameworks, Systems Thinking, High-Performance Teams

Focus on Synergistic Patterns, Not on Wars of Dogmas and Commandments
Methodologies is *not* the end goal; Pleasing Customers and Growing Business *is*!



smthatte@NewSynergyGroup.com
www.NewSynergyGroup.com