Sprint 0

Why Agile is NOT enough:
Lessons from Lean and other related approaches.

Dr Peter Morales
VP, CIO - CIEE.ORG

Objectives today

- Let’s look at the intersection of
    - Agile Methodologies
    - Lean Methodologies
    - Product development
    - Enterprise Architecture
    - Portfolio Management
    - Quality Assurance
    - Operations

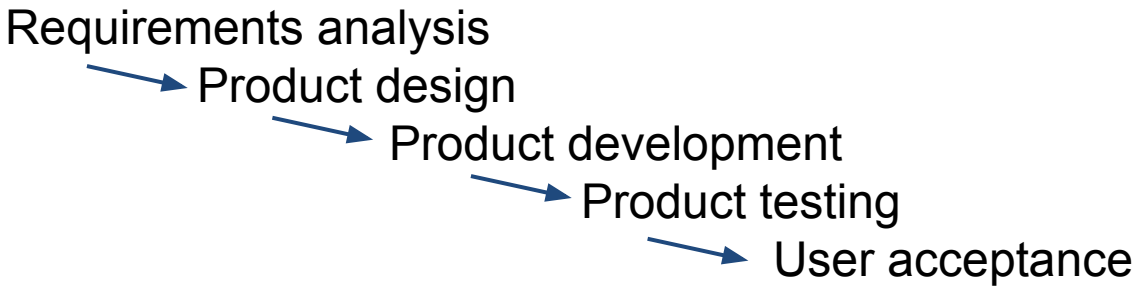
  Are they all relevant?

  No one methodology covers all bases?

Product vs Project Management

- How shall we design the product?
  - who is our target customer?
  - what are the features?
  - other considerations?

- Typical approaches use a “Waterfall” model



- Why is this bad?

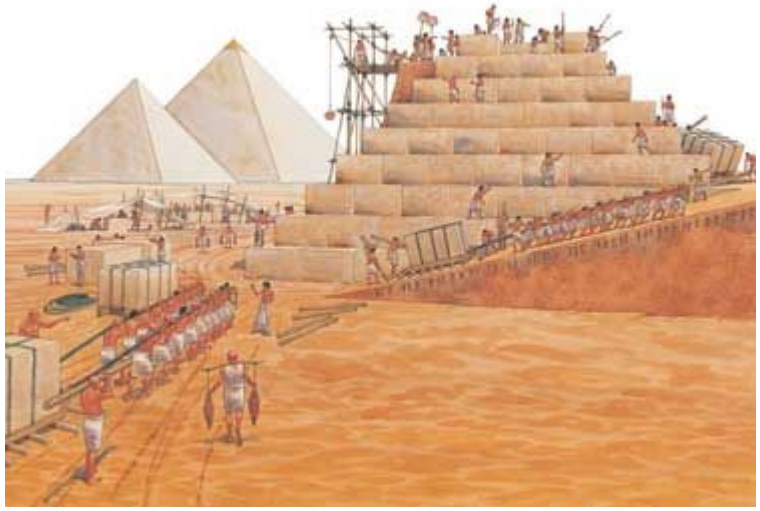
What is Agility

Agility = Speed x Accuracy

How does weight impact agility?

Project Success

- **Question(s): How does a project scale? What are the impediments?**
- **One approach: Brute Force**



The Agile Manifesto

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

http://agilemanifesto.org/

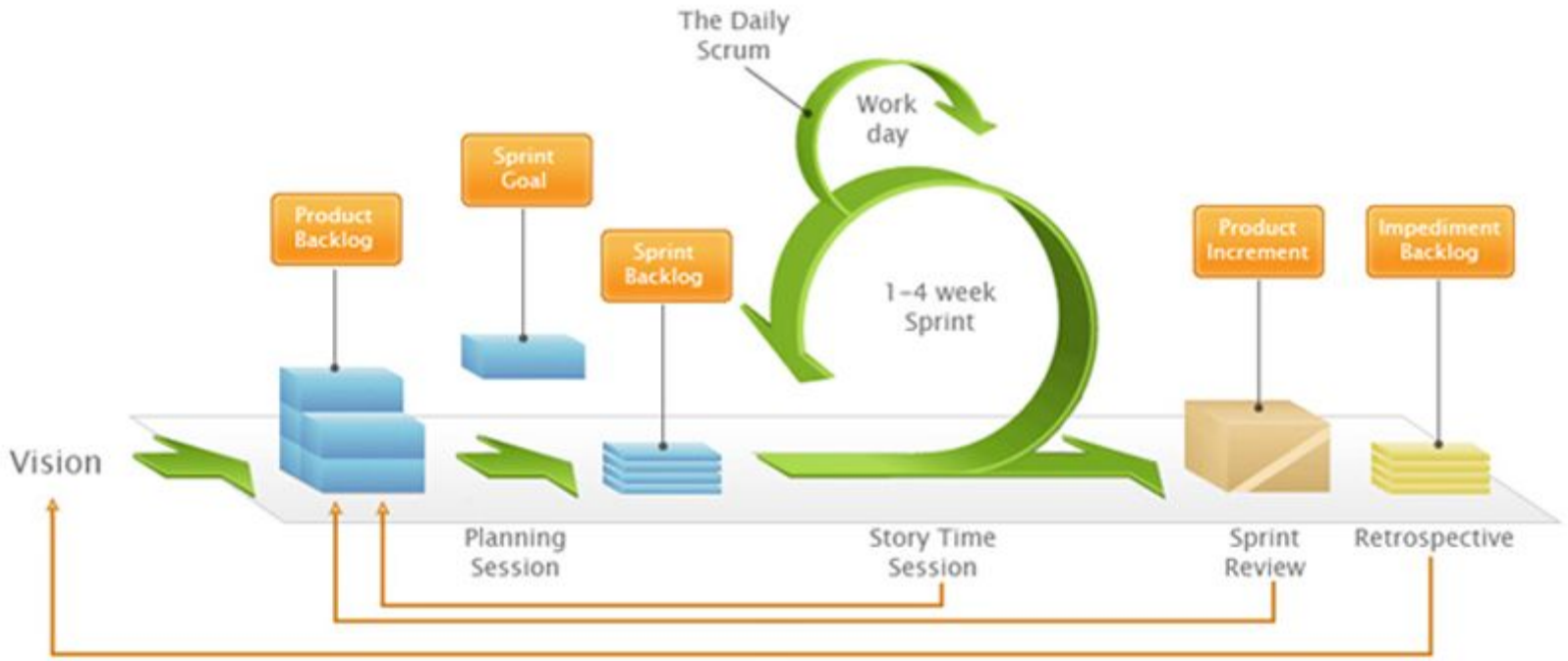
Roots in Lean Manufacturing

Agile traces some of its roots back to Lean Manufacturing, the original seven ***muda*** (sources of waste) are:

- **Transport** (moving products that are not actually required to perform the processing)
- **Inventory** (all components, work in process, and finished product not being processed)
- **Motion** (people or equipment moving or walking more than is required to perform the processing)
- **Waiting** (waiting for the next production step, interruptions of production during shift change)
- **Overproduction** (production ahead of demand)
- **Over Processing** (resulting from poor tool or product design creating activity)
- **Defects** (the effort involved in inspecting for and fixing defects)[19]

http://leanmanufacturingtools.org/71/muda-mura-and-muri-lean-manufacturing-wastes/

Agile Development - SCRUM



http://www.scrumhint.com/scrum-empirical-process-control-action/

Agile Development - SCRUM

Key Roles

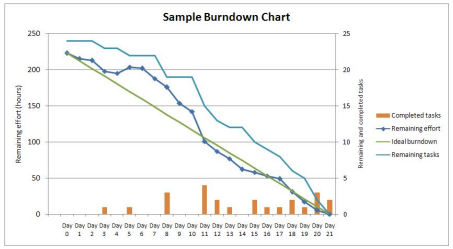
- Product owner - embedded business side customer
- Development team - small group, self organizing / flat governance
- Scrum master - coach, not a dictator

Key Events

- Sprint planning - Time boxed objectives (Epics, User Stories)
- Daily scrum - Reduce the cost / time of communicating
- Sprint review / sprint retrospective - continuously re-plan

Key Artifacts

- Product backlog - All the work left to complete
- A scrum task board - simple view of activities
- Product increment - always improving a working product
- Sprint burndown chart - need to understand the team's "velocity"



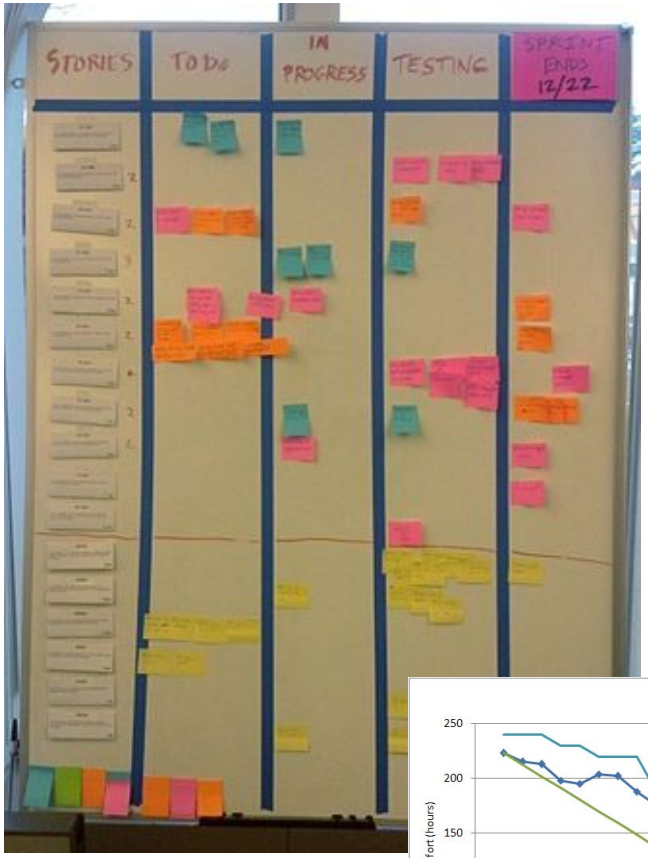
- Ruthlessly frugal -
  - "Less is more."
  - "If in doubt, leave it out !"
  - "Carry only what you can lift!"

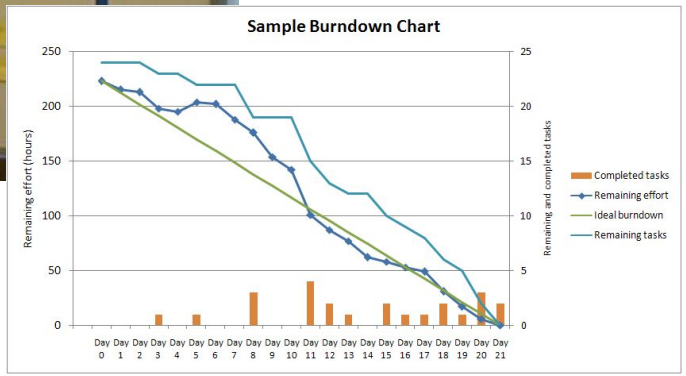
Agile Development - SCRUM

Why is this better than the Waterfall model?

Is this enough?

What's missing?





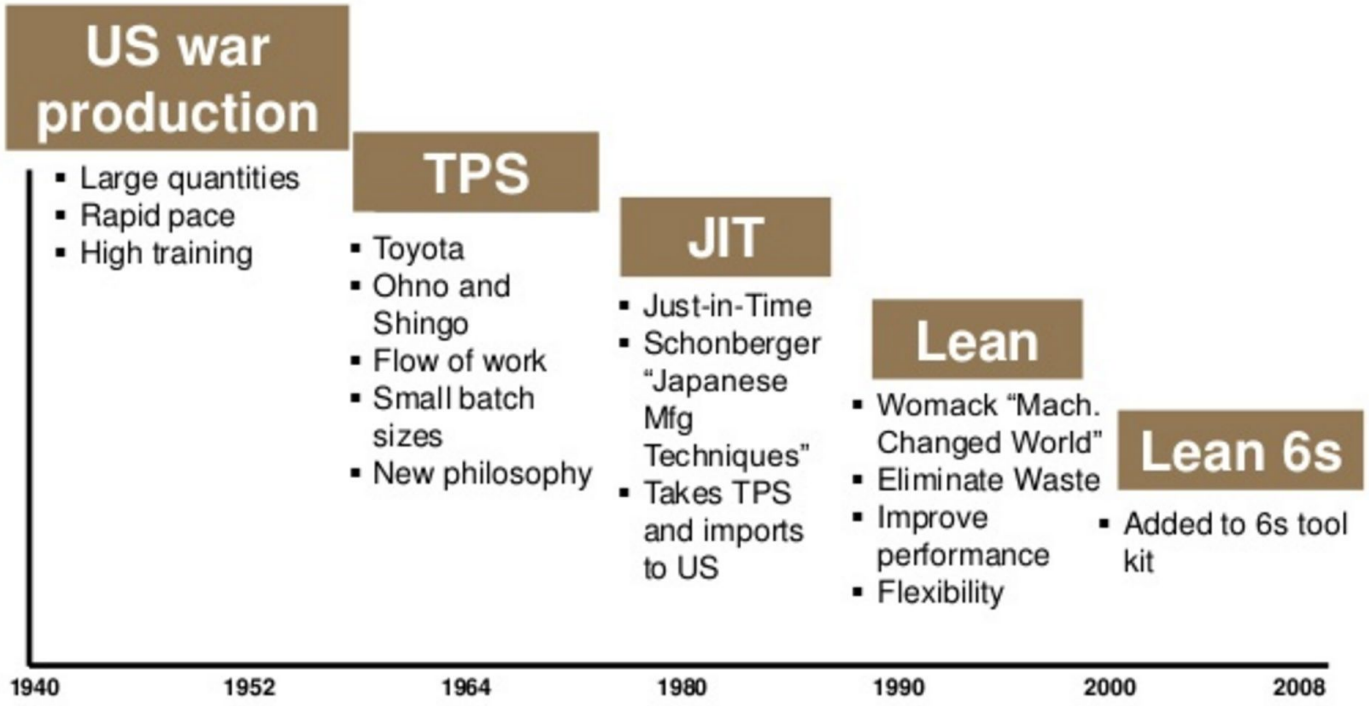
Criticisms:

- Scaling issues for larger projects / teams?
- Not predictable enough - project end date is a guess OR the features delivered can vary?
- Does not take Architecture epics into account?
- Does not address a portfolio of related projects?
- Allows the “customer” to change their mind TOO much?
- Need a ‘scoping’ sprint ?

Let’s look at related approaches

History of Lean

- **Another approach is Agile / Lean methods**



** Courtesy of the Juran Institute

History of Lean

WWII had ended & Japan was struggling with:

- Limited capital
- A small domestic market
- Demand for a wide range of vehicle types
- High energy costs
- Competition was high as automakers from other countries were eager to establish themselves in Japan
- Japan was in a depression and the Americans had restricted credit
- Toyota was facing bankruptcy
- Toyota president (Kiichiro Toyota) proposed firing ¼ of workers, which caused a revolt
- The Toyota's union was strong due to American had strengthened labor union rights (1946)

http://www.sixleansigma.com/index.php/wiki/lean/birth-of-lean/

History of Lean

The goals of a **Lean system** is to do more with less:

- less time,
- less space,
- less human effort,
- less machinery
- less materials

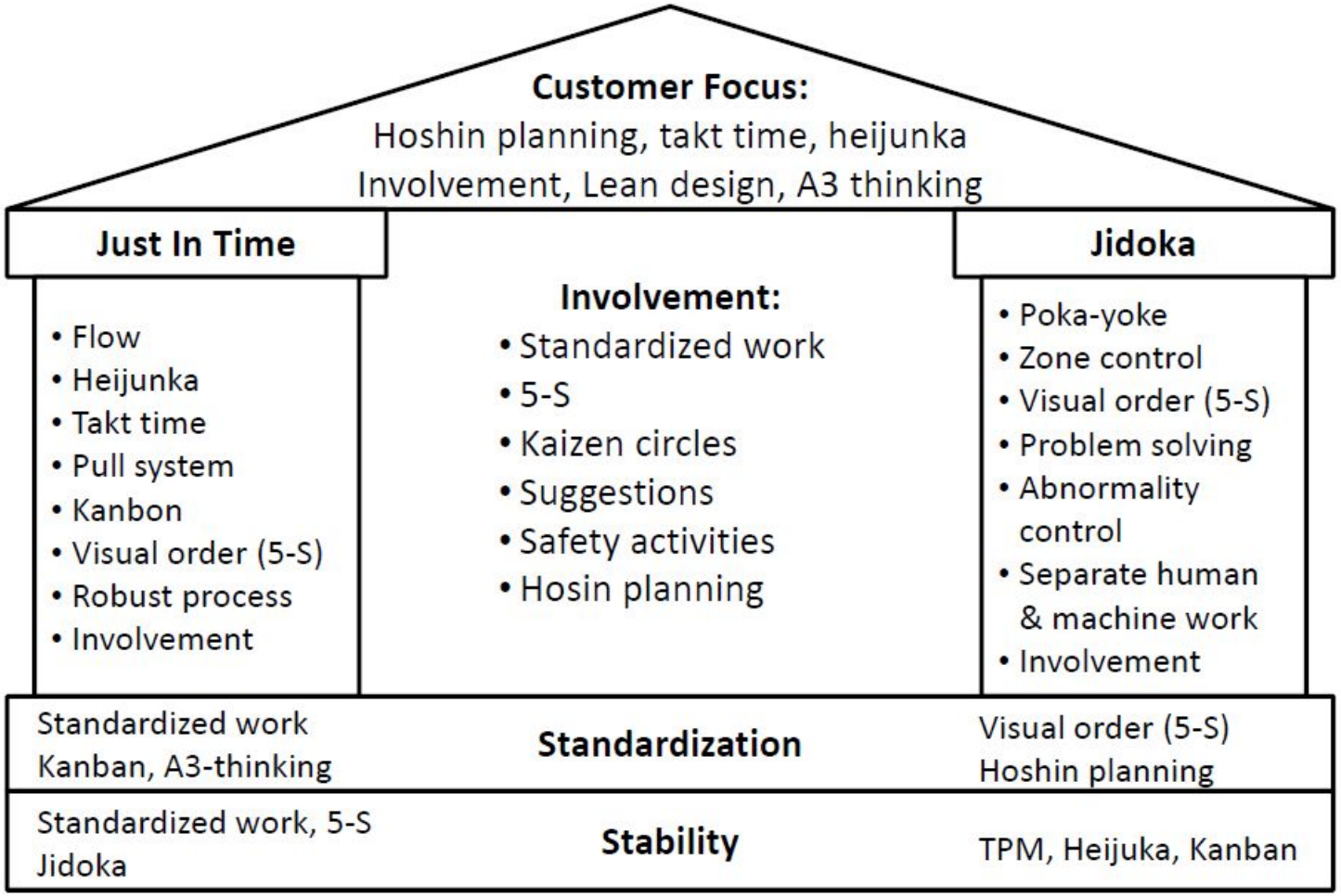
The key elements of a lean operation include the following items:

- Waste elimination in process
- Pull based production system
- One piece flow (as opposed to focus on batch or lots)
- Value stream mapping
- Setup time reduction
- Work cells

Cells typically have multiple processes are integrated together. Most often used for limited family of similar products or a single product. Product moves from process step to process step in small pieces batches or single pieces.

http://www.sixleansigma.com/index.php/wiki/lean/birth-of-lean/

House of Lean



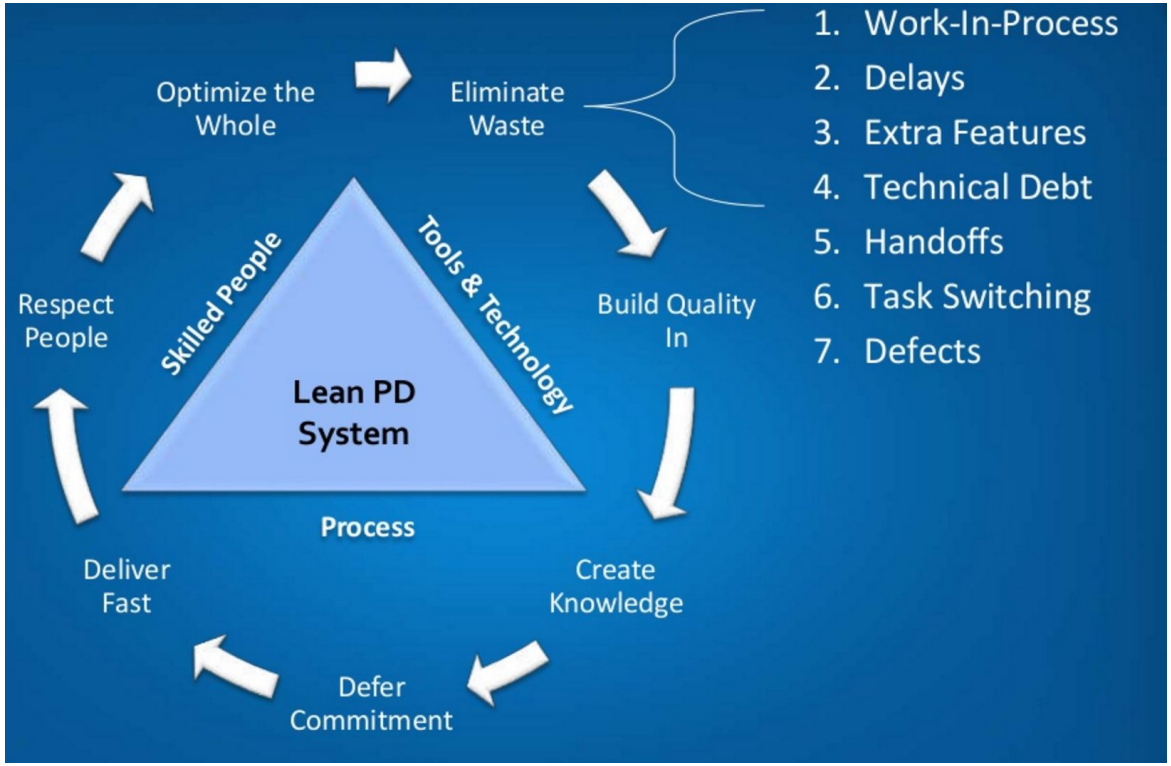
Lean Guiding Principles

Guiding Principles	Supporting Principles
▪ Respect every individual ▪ Lead with humility	▪ Assure a safe environment ▪ Develop people ▪ Empower and involve everyone ▪ Nurture long-term relationships
▪ Seek perfection ▪ Assure quality at the source ▪ Flow and pull value ▪ Embrace scientific thinking ▪ Focus on process	▪ Integrate improvement with work ▪ Identify and eliminate waste ▪ Keep it simple and visual ▪ Focus on value stream ▪ Insist on direct observation ▪ Standardize processes ▪ Rely on data ▪ Stabilize processes
▪ Think systematically ▪ Create constancy of purpose	▪ Align strategy ▪ Align systems ▪ Focus on long-term ▪ See reality
▪ Create value for the customer	▪ Identify cause and effect relationships ▪ Align behaviors with performance ▪ Measure what matters

** Courtesy of the Juran Institute

Lean Methods

- **Lean Objectives map to Agile Objectives**



Lean Methods

- Though there are significant differences between Lean and Agile
- Agile is Brutally Frugal. ANYTHING that is not driven by customer value is WASTE!
- Scrum in particular does not embrace significant planning for:
  - Architecture alignment across the enterprise
  - Portfolio alignment of projects
  - Portfolio planning of product features

Lean Versus Agile

Sprint 0?

- More and more teams are looking at ways to augment Agile to align with the needs of the larger enterprise
- Sprint 0 is one approach. The Agile community would argue:
  - You are having trouble moving away from Waterfall
  - The organization is having trouble giving control to a self empowered team.
  - Planning is another way of doing Big Design Up Front.

https://www.scrumalliance.org/community/articles/2013/september/what-is-sprint-zero

Sprint 0?

- In this author's opinion, too many agile projects struggle with lack of clarity about what the business really needs and how to integrate the project with the rest of the environment.

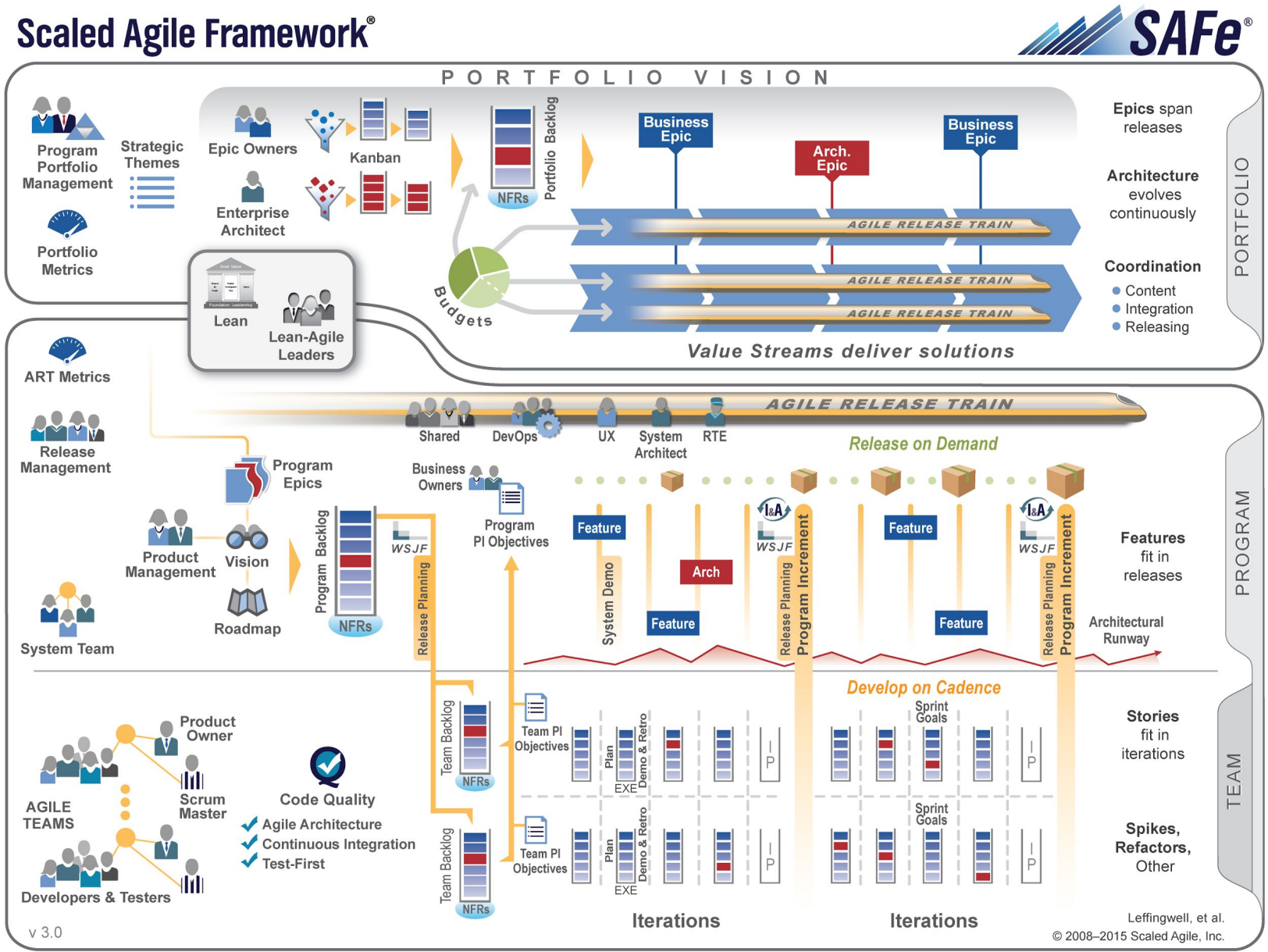
What about the rest of the Organization?

Other Areas of Focus

- Operations and support
- Program Management
- Vendor Management
- Portfolio Management and Planning
- Systems Architecture
- Quality Assurance
- Knowledge Management
- People, culture, and continuous improvement!

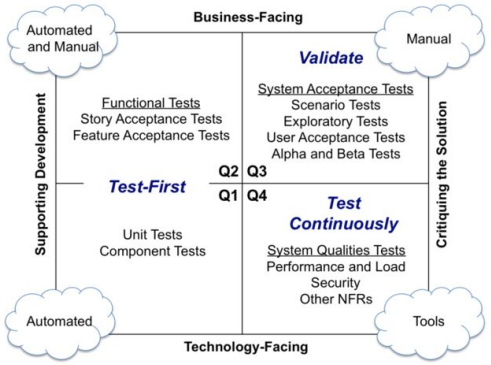
They may be outside the scope of Scrum and Agile but certainly need to integrate.

Another emerging model - SAFe

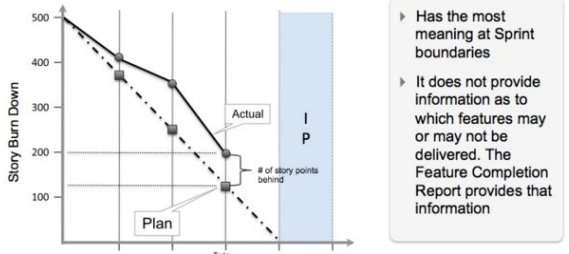


SAFe Core Values

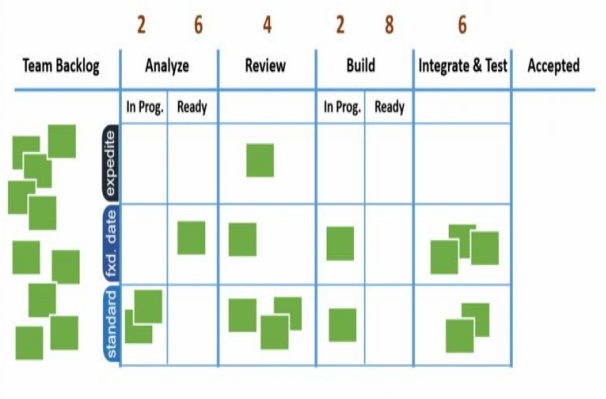
1. Quality - Test Driven Development, Continuous Integration and Delivery
2. Program Execution - Sprint Epics driven by Business and Architecture
3. Alignment with Business Epics (high level objectives - ‘traunches’)
4. Alignment with Product Road Maps
5. Alignment with Reference Architecture
6. Transparency and Metrics (KPIs, Dashboards, Kanban,...)



QA Strategy



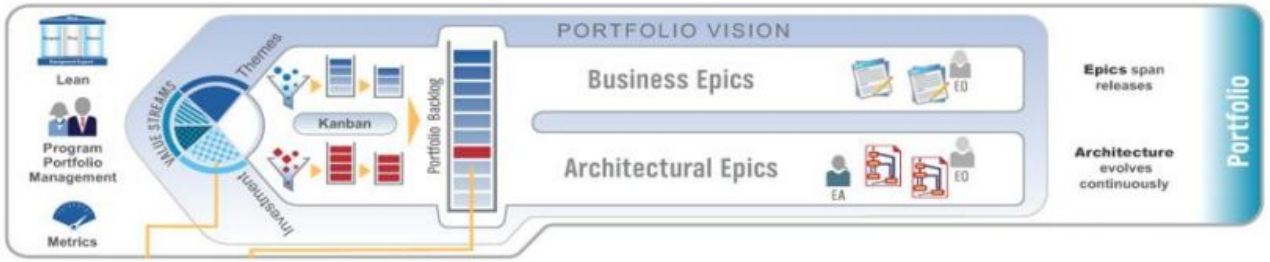
Burndown Chart



Kanban

http://www.scaledagileframework.com/

Scale to Portfolio

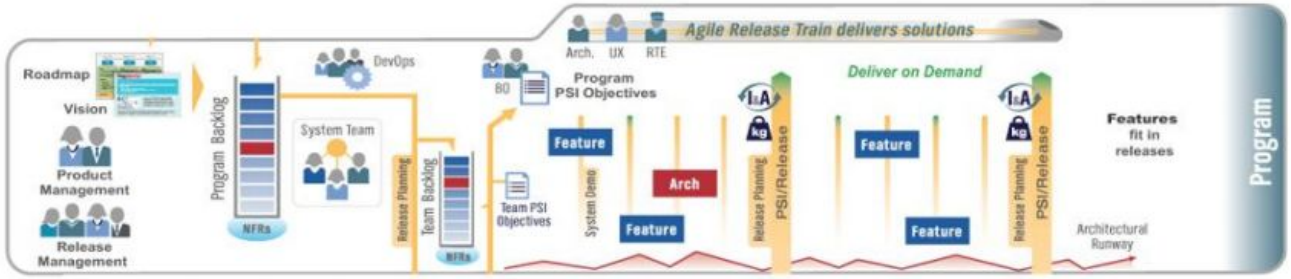


- Centralized strategy, decentralized execution
- Investment themes provide operating budgets for trains
- Kanban systems provide portfolio visibility and WIP limits
- Objective metrics support governance and kaizen
- Value description via ***Business*** and ***Architectural Epics***

http://www.scaledagileframework.com/

Scale to Program

- Self-organizing, self-managing team-of-agile-teams
- Continuous value delivery
- Aligned to a common mission via a single backlog

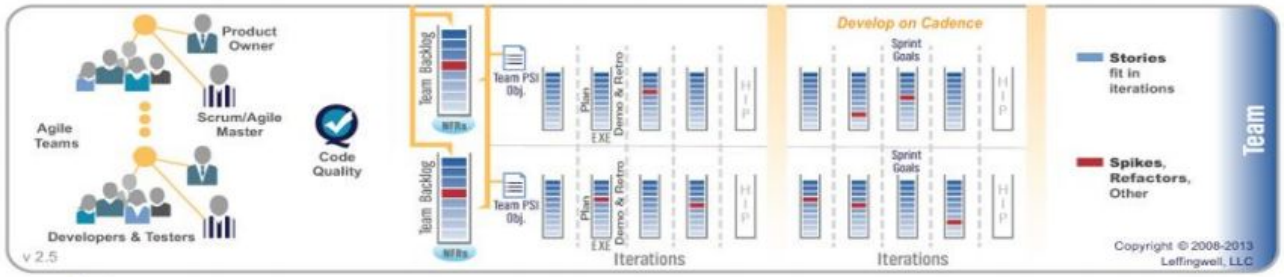


- Common sprint lengths and estimating
- Face-to-face planning cadence for collaboration, alignment, synchronization, and assessment
- Value description via ***Features*** and ***Benefits***

http://www.scaledagileframework.com/

Scale Teams

- Empowered, self-organizing, self-managing cross-functional teams
- Valuable, fully-tested software increments every two weeks
- Scrum project management practices and XP-inspired technical practices
- Teams operate under program vision, system, architecture and user experience guidance
- Value description via ***User Stories***



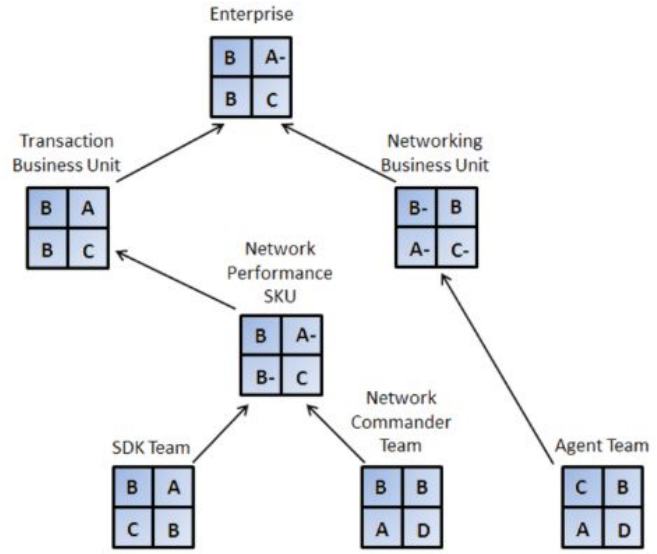
http://www.scaledagileframework.com/

Measure Objectives

Efficiency	Value Delivery
Sample Measures: • Contribution Margin • Organizational Stability • Team velocity vs. capacity	Sample Measures: • Number of releases • Value feature points delivered • Release date percentage • Architectural refactors
Quality	**Agility**
Sample Measures: • Defects • Support calls • Support satisfaction • Product Satisfaction • Escalation Rate percentage	Sample Measures: • Product Ownership • Release Planning & Track. • Iter. Planning & Tracking • Teamwork • Testing and Dev practices

Identify Metrics
(sample - not sure I agree with the ones chosen)

Roll up scorecards
(bottom up)



http://www.scaledagileframework.com/

Too Heavy?

Balanced Approach - Guiding Principles

- Promote a culture where data helps us be more responsive to the right issue at the right time.
- Define and continually improve process agility (without excess bureaucracy)
- Partner with the business to do the most impactful work first.
- Embrace and support agile development.

  Why: because agile has shown to be more responsive to change.
- Add additional support from Lean community.

  Why: because Lean has practices that encourage thoughtful planning and architecture.

Information Technology - Operations

Ops Guiding principles

- Monitor and respond before business is impacted
- Superb customer focus
- Continuous improvement through data and metrics

Ops initiatives

- Improve data center operations: back ups, recovery, run a DR / BC test
- Define response strategy based on application Tier 1, 2, 3 criticality
- Implement Change Management process with Change Control Board for releases
- Implement a deep monitoring strategy. availability PLUS performance, security and escalation
- Implement a Critical Incident Response process and roles (with defined overlapping / support roles) – objective: respond before the business is affected
- Develop a virtual ‘war room’ with a web accessible Operations Dashboard and Console
- On board an Information Security Manager (use a simplified version of FISMA practices for security model) and integrated security monitors
- Begin Elastic Public Cloud proof of concept
- Implement Key Performance Indicators

Information Technology - Operations

Ops KPIs

- Incidents per month
- Operational uptime
- Mean time to recovery

Operational Staff Development

- Develop internal knowledge management and training programs for operations
- Regularly Scheduled ‘lunch and learn’ sessions include :
  - Critical Response Process,
  - Root Cause Analysis Techniques,
  - Cloud Operations Training,
  - Disaster Recovery / Business Continuity Process
- Promote sharing between teams, provide a platform to show case successes

Information Technology – Program Management

PMO guiding principles

- Partner with the business to validate workflows (facilitate LEARNING in product development cycle)
- Prioritize project work by risk and value to the business
- Validate business workflows using light weight tools
- Provide User Acceptance Testing framework to measure project progress
- Accompany each release with Change Plan (roles, delta training, rollout)

PMO initiatives

- Implement a Vendor Management Office team with refined legal templates enforcing a highly integrated approach with incremental deliverables and milestones
- Implement a Lean oriented methodology that supports the responsiveness of Agile but adds layers for Portfolio Planning, Early Workflow Validation, Test Case Development
- Work with our business partners to review and refine wireframes and workflows
- Feed the Development process with better refined requirements that are validated with the business
- Build and automate test cases to build a full regression testing capability
- Use validated requirements to begin development of multi-media training packages for each release.
- Enhance release adoption with role based training (build a knowledge sharing platform)

Information Technology – Program Management

PMO KPIs

- Releases on time and budget
- Defects found in QA
- Defects found in production

PMO Staff Development

- Develop internal knowledge management and training programs for PMO
- Regularly Scheduled ‘lunch and learn’ sessions include :
  - Project Portfolio Management
  - Product Planning – making effective feature choices
  - Lean Agile Development Methodology,
  - Vendor Management,
  - Test Driven Development
- Promote sharing between teams, provide a platform to show case successes

Information Technology – Application Architecture and Development

Development guiding principles

- Optimize architecture as reusable business services
- Separate form from function (decouple front and back ends)
- Eliminate rework as early as possible

Development initiatives

- Improve prioritization of requirements based on risk and value to the business
- Improve requirements 'grooming' by getting early feedback from the business
- Provide a path for the business to validate and learn. Product development is a learning process.
- Create more stable release cycles by reducing re-work and better testing
- Create more stable releases by providing staff to respond to operational emergencies
- Developers are still responsible for Unit Testing but are supported with Release Regression Test Suite
- Develop a micro-services architecture that supports business process re-use
- Develop a secure and agile workflow development process
- Micro-services architecture provides simpler model for scaling, change and upgrades

Information Technology – Application Architecture and Development

Development KPIs

- Services re-used
- Releases on time,
- Defects found in QA,
- Defects found in prod

Application Dev Staff Development

- Develop internal knowledge management and training programs for Development teams
- Regularly Scheduled ‘lunch and learn’ sessions include :
  - Lean Agile Development Methodology,
  - Vendor Management,
  - Test Driven Development,
  - Business Micro Service Architecture
- Promote sharing between teams, provide a platform to show case successes