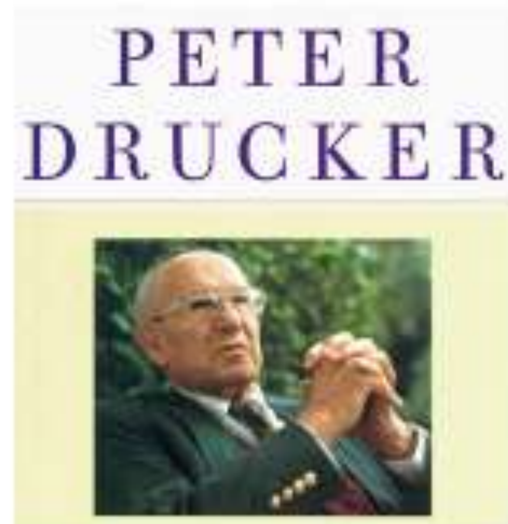
# Lean Agile Metrics and KPIs

- **Beware the Hawthorne Effect**
  - We affect what we measure

- **Knowledge workers need to first define:**
  - What is our business?
  - Who is our customer?
  - What does our customer consider valuable?




**Drucker:** Measures of quality must flow from the above.

Taken from http://aplndc.pbworks.com/f/2009-05-Agile-Metrics.pdf

# Measuring/Managing Organization Performance

What are we here for?

Do Agile?

Develop software?

Generate most "Value for R&D investment" in a sustainable way?

$ for the shareholders?

So how do we perform in a way that aligns with that?

# Assumption – Agile drives Delivery engine

Agility → (1)

Better Software Development →

**Better Business**

(1) Proven in industry-wide research (Cutter, Forrester, QSM), various enterprise companies (Yahoo, Google, Nokia, Salesforce, etc.)

# Assumption – Agile drives Delivery engine

So we…

Measure Software Development →

To Predict Business performance

Measure Agility →

To Predict Software Development Performance

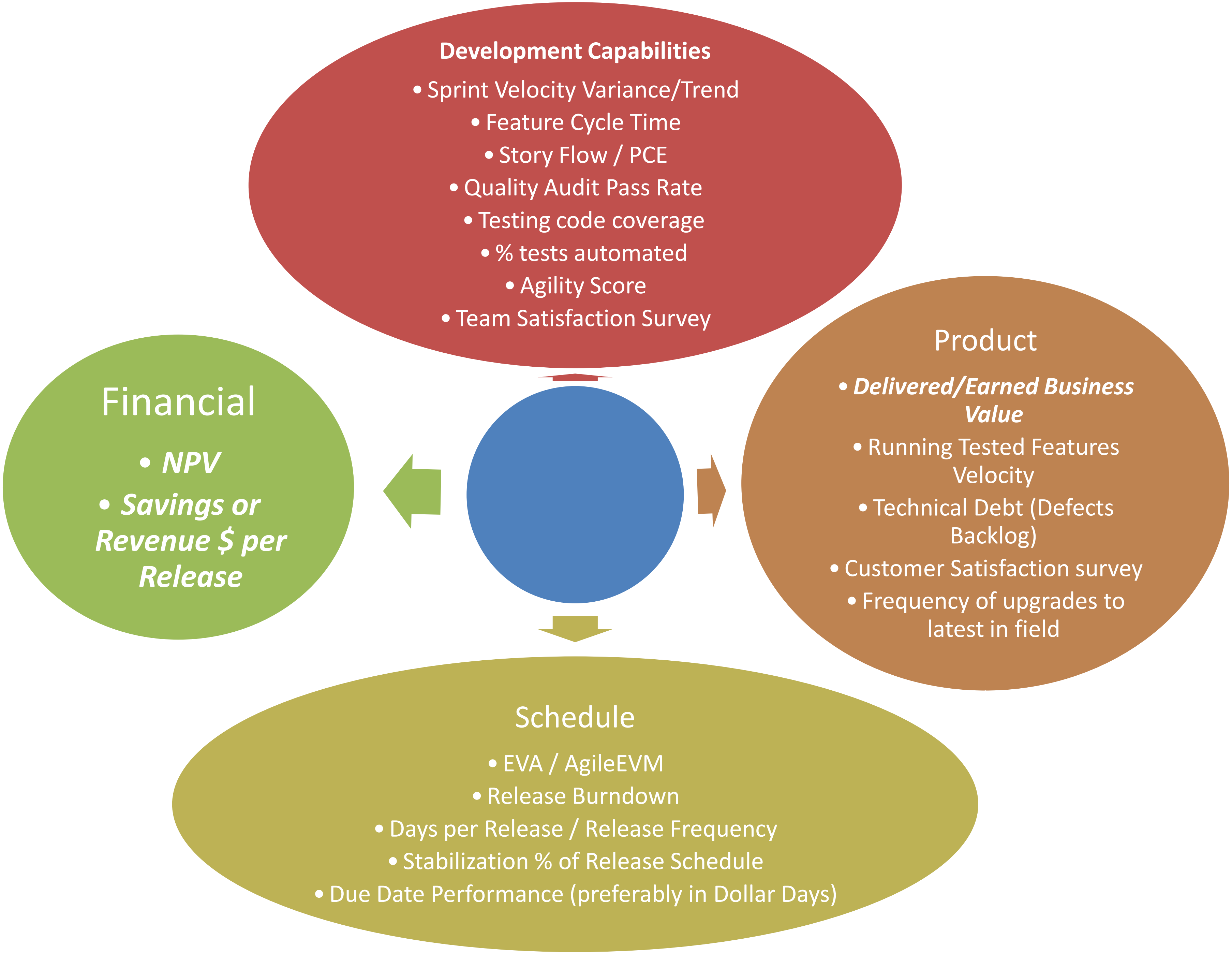
(1) Proven in industry-wide research (Cutter, Forrester, QSM), various enterprise companies (Yahoo, Google, Nokia, Salesforce, etc.)

# Lagging/Leading KPIs

Which leads us to KPIs – **Key Performance Indicators**

There are two major types of KPIs: leading and lagging indicators.

- Leading indicators measure activities that have a significant effect on future performance.
- Lagging indicators, such as most financial KPIs, measure the output of past activity.


Development Capabilities
• Sprint Velocity Variance/Trend
• Feature Cycle Time
• Story Flow / PCE
• Quality Audit Pass Rate
• Testing code coverage
• % tests automated
• Agility Score
• Team Satisfaction Survey
Financial
• NPV
• Savings or Revenue $ per Release
Product
• Delivered/Earned Business Value
• Running Tested Features Velocity
• Technical Debt (Defects Backlog)
• Customer Satisfaction survey
• Frequency of upgrades to latest in field
Schedule
• EVA / AgileEVM
• Release Burndown
• Days per Release / Release Frequency
• Stabilization % of Release Schedule
• Due Date Performance (preferably in Dollar Days)

# Leading Agile Indicators

- Change in velocity
- Change in Team Capacity
- Product Owner Time with Team
- Team co-location time
- Backlog growth
- Defect generation and resolution rates
- Process QA Checklist–
  - Are stand-ups happening?
  - Are demos happening?
  - Are retrospectives happening?
  - Is working software delivered in every sprint?
  - Is testing occurring in every sprint?

**They Want It All:**

- "I want it fast"
- "I want it cheap"
- "I want good quality"
- "It must pass governance"

Use with Care!

# Focus on being really READY and really DONE

- Being READY READY
  - reduces every category of work (defects, rework, total work required, and process overhead) by almost 50%
  - 160%/340% productivity improvement
- DONE DONE - "Build Quality into Sprints"
  - Fix problems fast
  - Don't save quality issues for later
  - With the right infrastructure and mindset, IMPROVES productivity

- Based on Lean/TOC principles…

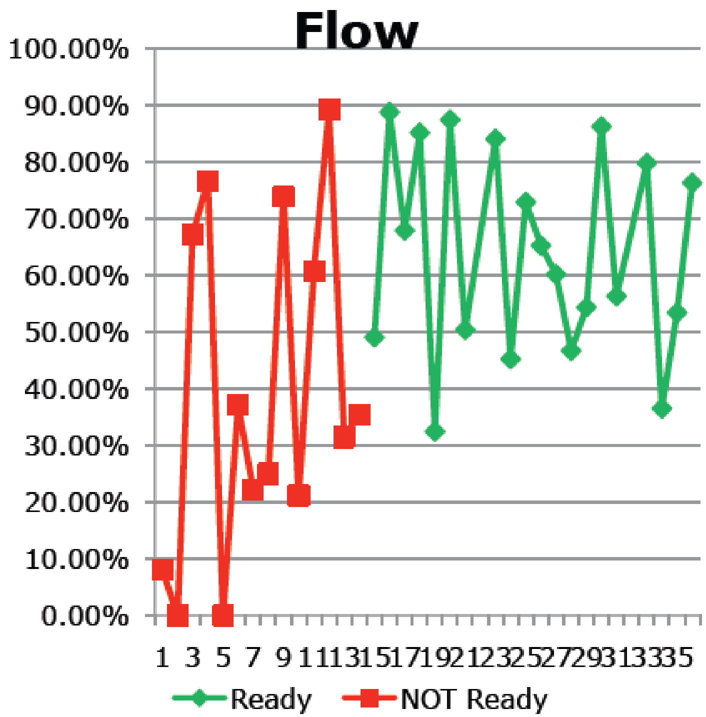
# Relevant Measures

- READY READY
  - Leading:
    - Depth of READY backlog for teams
  - Lagging:
    - Story Flow / Process Cycle Efficiency (PCE) – time story actually worked on versus overall time story in processing
    - Actual Effort variance

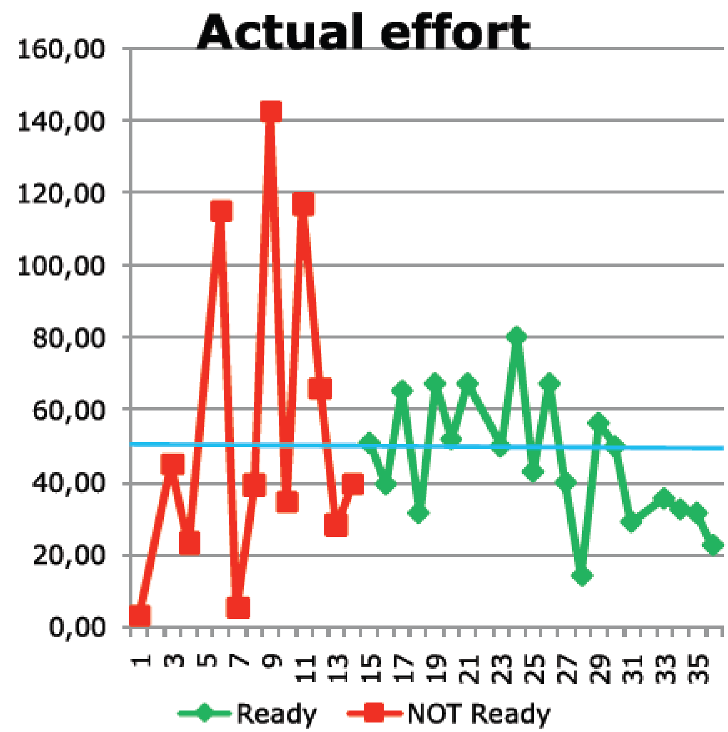
- DONE DONE
  - Average Time to fix for defects (goal – hours-days)
  - Average Time to fix for build failures (goal – minutes)
  - Defects backlog across sprints (goal – 0)
  - Development/Stabilization schedule ratio (~30% normally. 10% goal)

# READY Leads to Flow and Productivity

**When work allocated to sprint is READY, flow and stability is achieved**

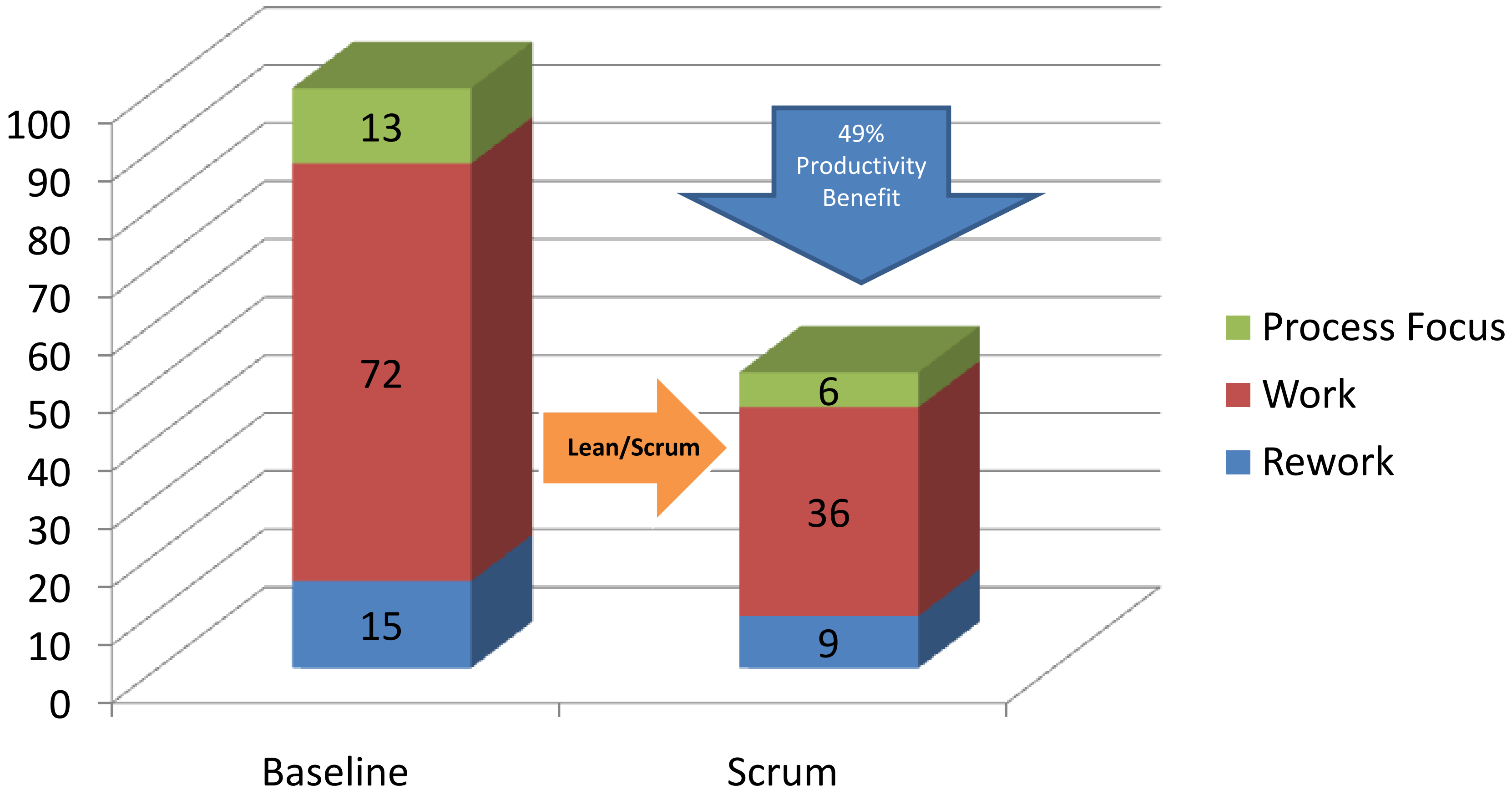



Objective: 60%




Objective: 50h

# Productivity Results in Systemix

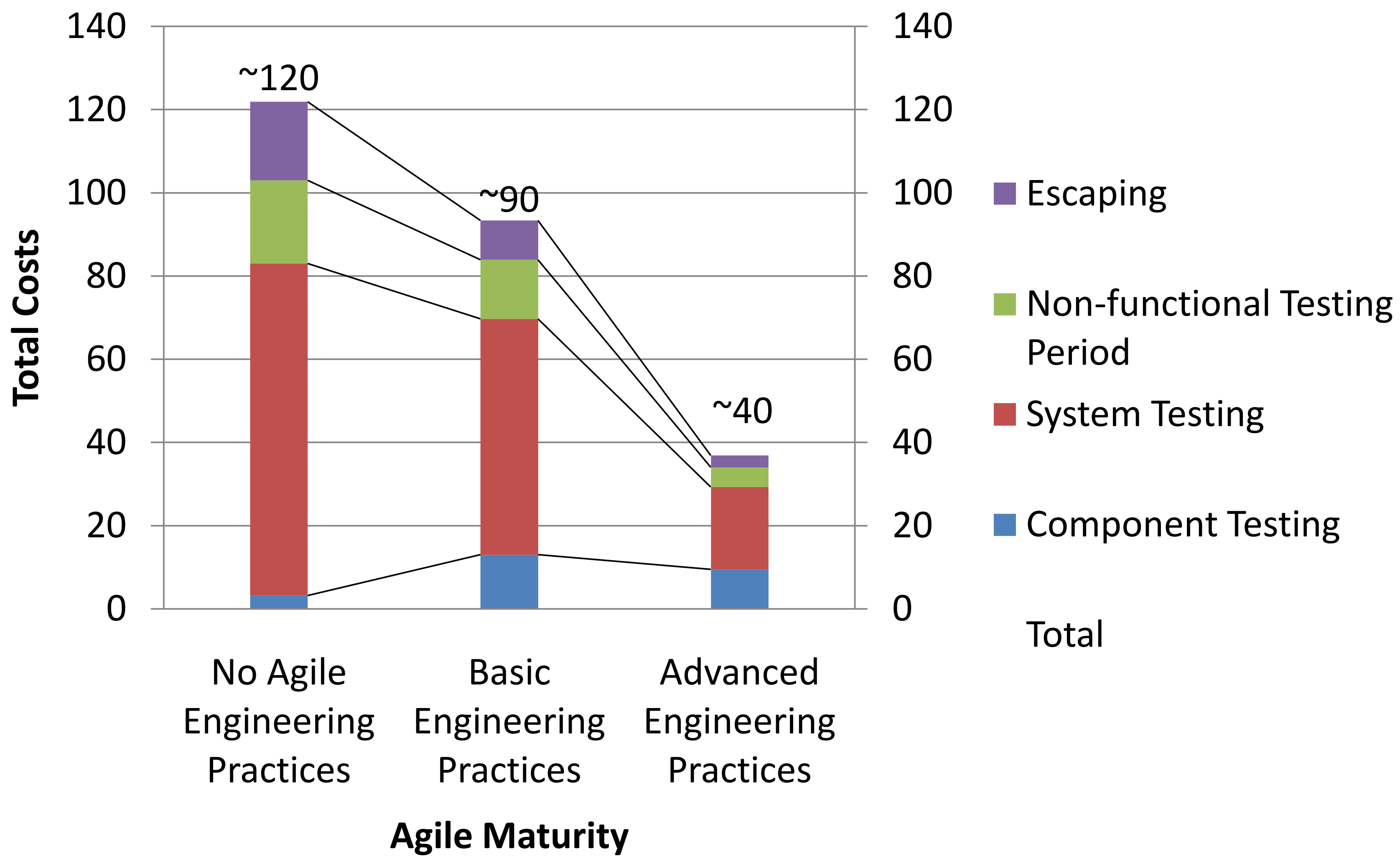


# Why focus on “Time to Fix Defects”

Because Fixing bugs earlier is cheaper

- Why?
  - Early warning (broken/incompatible/conflicting) – faster pin-pointing of root cause and fixing
  - Easy Revert and offline debugging – not in the critical path
- Savings of **~10K$ for each bug** found during sprint
- http://studios.thoughtworks.com/2007/5/10/continuous-integration-in-the-enterprise-with-cruisecontrol
- Example from a product in an Enterprise Portfolio – 40 escaping defects translate into **400,000$** (multiply per product per release)

# Example Bug fix costs per Maturity



# Faster/Cheaper **Stabilization Period**

Integration problems are detected and fixed continuously - no last minute hiatus before release dates

Less cycles / retesting – Arriving to Packaging with higher quality/readiness

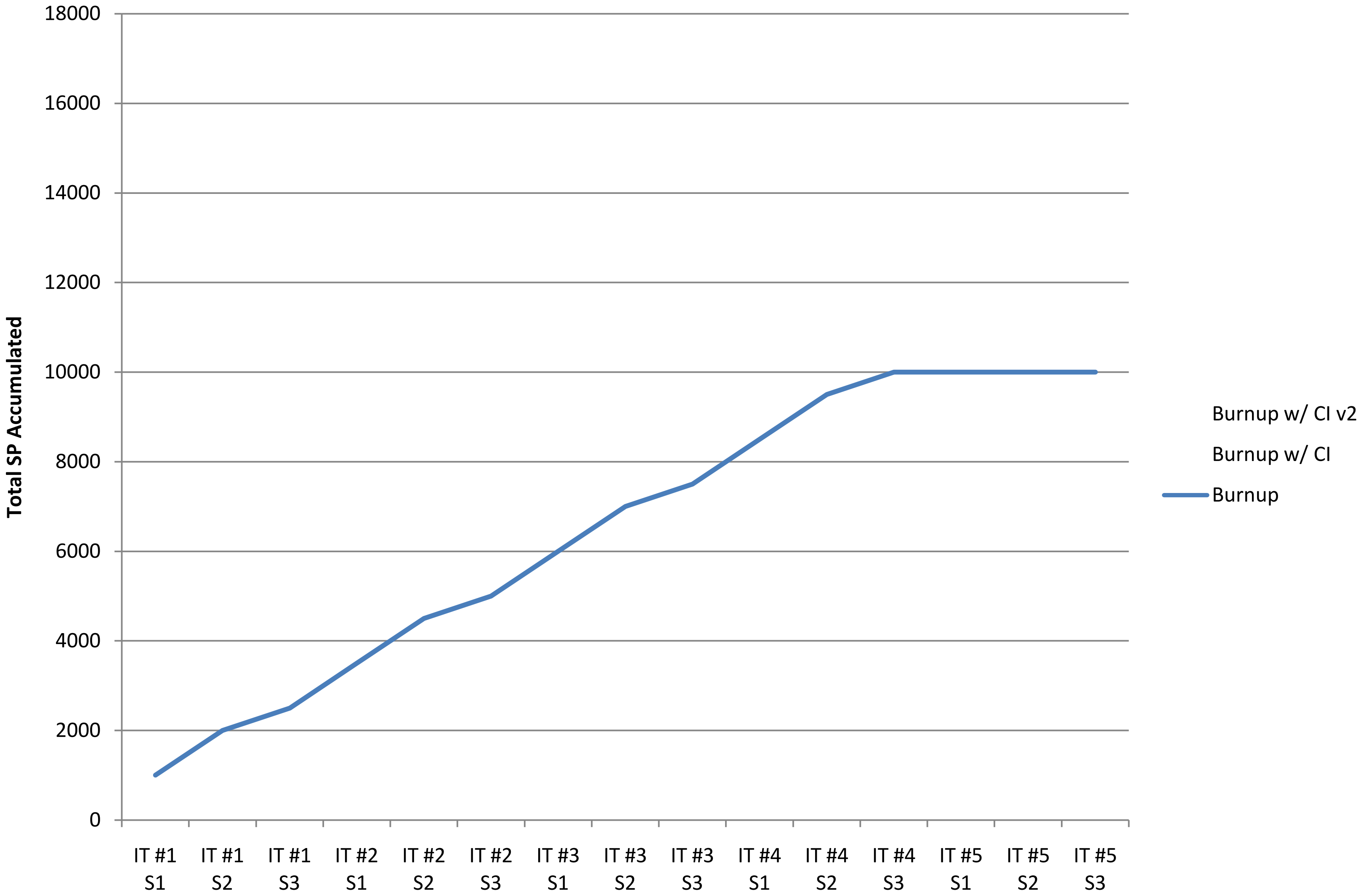
Product/PIL focus on Exploratory, Advanced scenarios sooner, and can finish faster

Better predictability allows squeezing in more functionality until the last minute

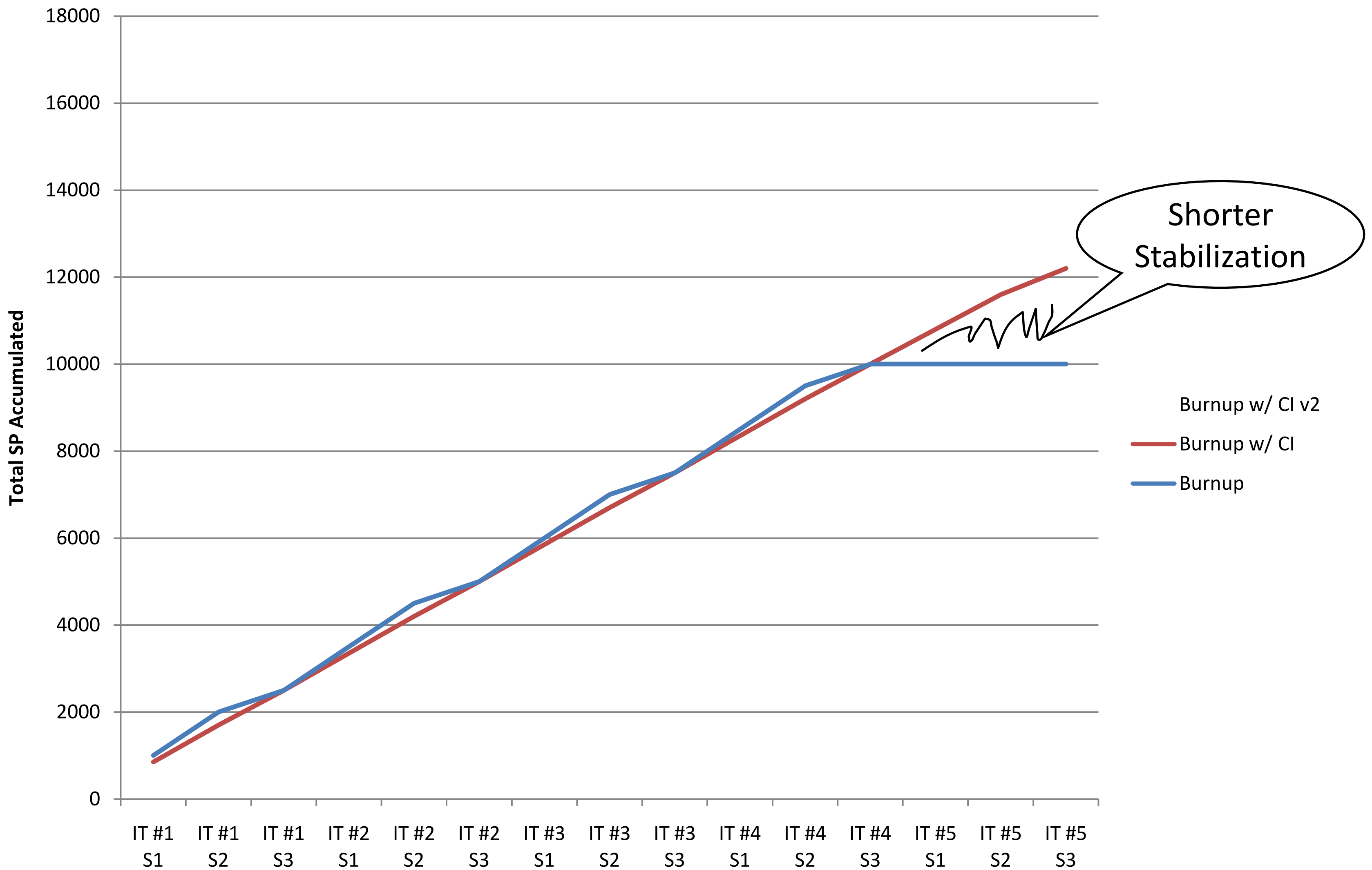
Smoother and more predictable flow – shorter stops

Makes  frequent release strategy more feasible financially

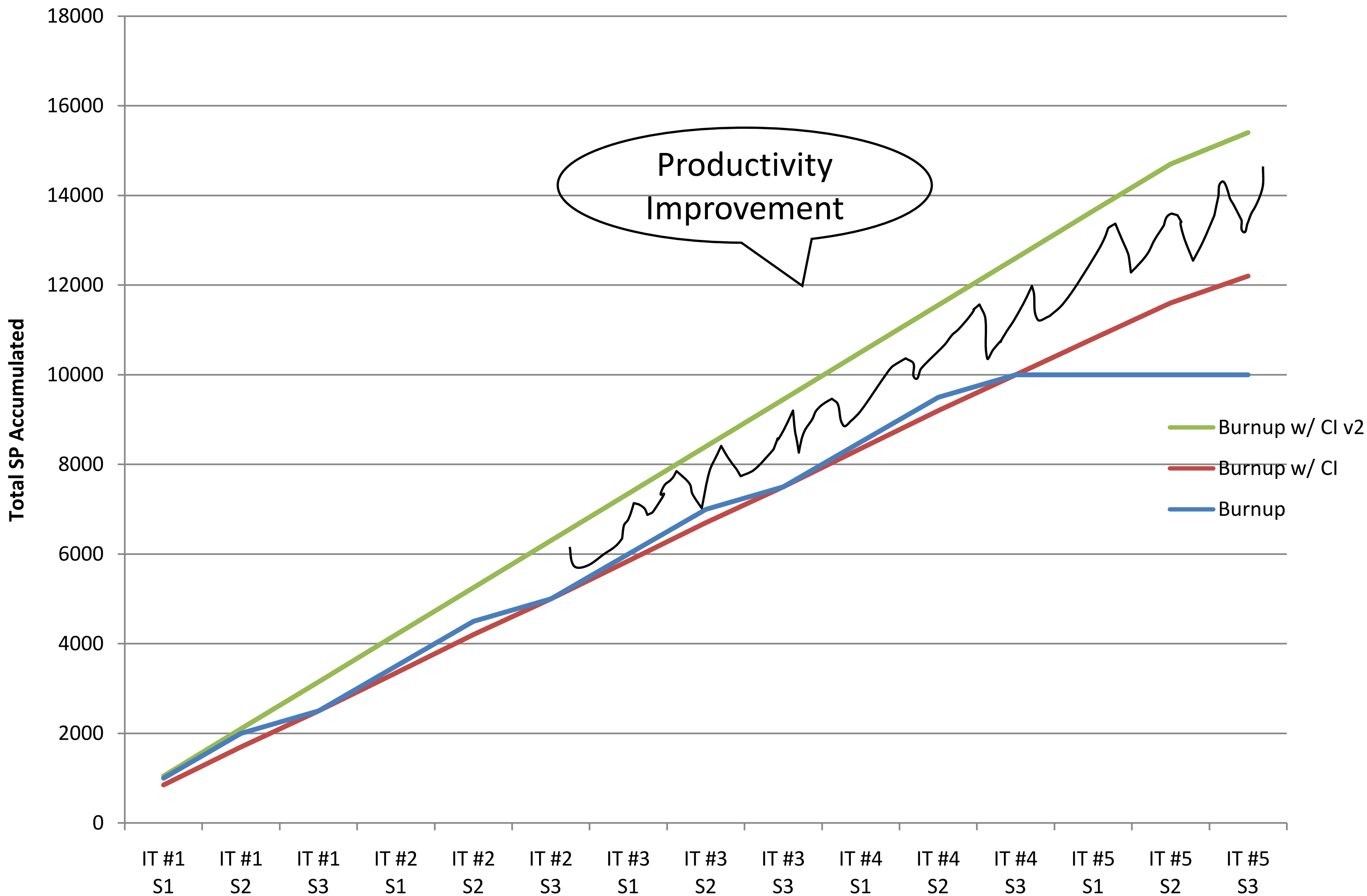
# Stabilization Period implications



# Release Burnup with and without CI



# Release Burnup with and without CI



# Measures for Eng Practices Maturity


% Time build is broken

% Time build is broken due to compilation errors

- Come on…

% Code Coverage

- By Unit Tests
- By Test Automation

Checkin early and often

- Checkin frequency
- Size of checkins (Smaller better)

# Burndown + EVM

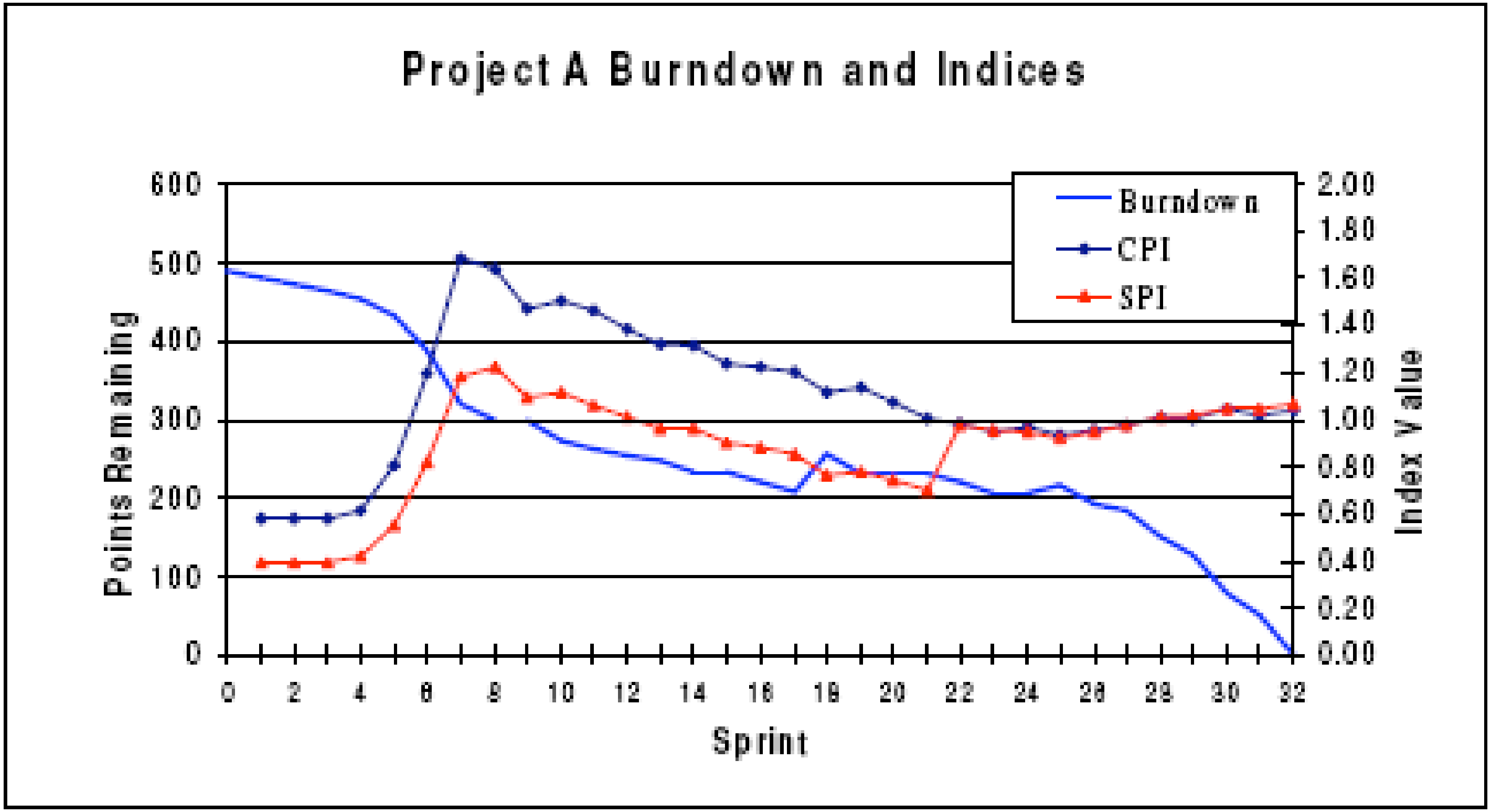



**Fig. 6: Project A Burndown, CPI, SPI**

# Agile Earned Value Mgmt

- First of all, it's not about value as it "goodness" or "business value" – it's value as in "actually done" – and the emphasis is on "earned"
- EVM is about measuring project performance, comparing budgets to actuals in scope, schedule, and resource
- The genius of AgileEVM is the realization that stories are an appropriate thing to measure and count for EVM purposes *
  - In fact, stories are better than the activities we normally measure in software… done is better defined
  - The unit of value that we are measuring is the "earned StoryPoint"; that is, we are measuring stories that are "done"

* http://www.solutionsiq.com/PDF/Sulaiman-AgileEVM.pdf

# AgileEVM Metrics

- The two most important EVM metrics are Cost Performance Index (CPI) and Schedule Performance Index (SPI)
  - CPI answers the question “are we paying what we expected for each SP?”

    *CPI = (baseline Hrs/SP) / (actual Hrs/SP)*

  - SPI answers the question “are we getting the SPs at the rate we expected?”

    *SPI = (actual SP/Sprint) / (baseline SP/Sprint)*

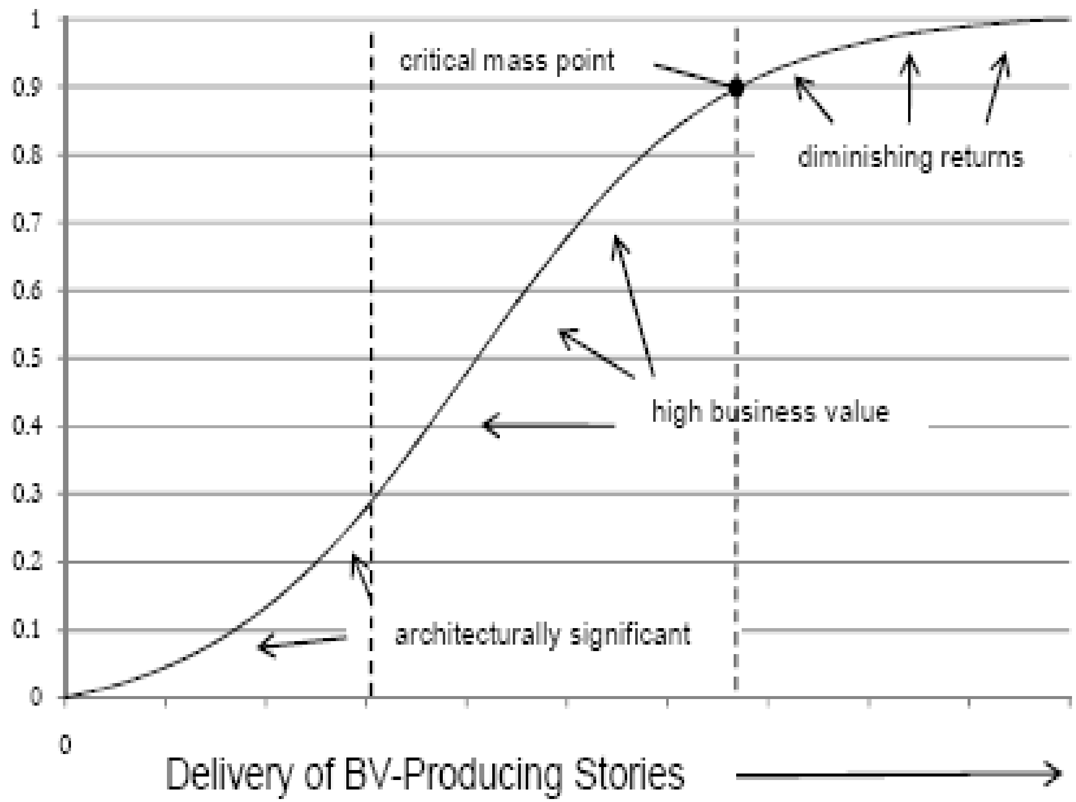
- We like CPI and SPI to be $\geq 1$ in standard EVM

- You’ll have to trust me on the derivations* ☺

* http://danube.com/system/files/Monitoring+Scrum+Projects+with+AgileEVM+and+Earned+Business+Value+_EBV_+Metrics.pdf

# Earned Business Value

- Allocate BV points per minimally marketable feature (MMF)
- Track completion of BV points
- Complements AgileEVM




- We're finding stories that take us along the S-shaped curve of BV delivery…

# Characteristics of Effective KPIs

Effective KPIs often exhibit the following 12 characteristics:

1. **Aligned.** KPIs are always aligned with corporate strategy and objectives.
2. **Owned.** Every KPI is "owned" by an individual or group on the business side who is accountable for its outcome.
3. **Predictive.** KPIs measure drivers of business value. Thus, they are leading indicators of performance desired by the organization.
4. **Actionable.** KPIs are populated with timely, actionable data so users can intervene to improve performance before it is too late.
5. **Few in number.** KPIs should focus users on a few high-value tasks, not scatter their attention and energy on too many things.
6. **Easy to understand.** KPIs should be straightforward and easy to understand, not based on complex indexes that users do not know how to influence directly.
7. **Balanced and linked.** KPIs should balance and reinforce each other, not undermine each other and suboptimize processes.
8. **Trigger changes.** The act of measuring a KPI should trigger a chain reaction of positive changes in the organization, especially when the CEO monitors it.
9. **Standardized.** KPIs are based on standard definitions, rules and calculations so they can be integrated across dashboards throughout the organization.
10. **Context driven.** KPIs put performance in context by applying targets and thresholds to performance so users can gauge their progress over time.
11. **Reinforced with incentives.** Organizations can magnify the impact of KPIs by attaching compensation or incentives to them. However, they should do this cautiously, applying incentives only to well-understood and stable KPIs.
12. **Relevant.** KPIs gradually lose their impact over time, so they must be periodically reviewed and refreshed.

S

## So Agile Maturity Score as a KPI should

1. Be **aligned** with the goal of being a more Agile.
2. Be **owned** by each Product – who are accountable for the outcome of their efforts
3. **Predict** successful software development delivering best business value for the buck
4. Provide **timely, actionable** data, so users can intervene and improve performance BEFORE its too late
5. **Focus on key high-value** indicators
6. **Easy to understand** – users should know how to drive for improved performance
7. **Balanced** with Velocity/Quality/Stakeholder satisfaction/other.
8. **Trigger positive outcomes** in each group working with it
9. **Standardized** scale across company
10. Targets/Objectives should **be driven by context/situation** of each group
11. **Reinforce performance** by loosely connecting to incentive systems? Maybe just as a very low influence. Main driver for incentives should continue to be business/team performance meeting goals. Agile Maturity should drive that
12. **Stay relevant**. Focus on areas which still need work, Add more measures for future objectives when become relevant

Be **aligned** with the goal of being a more Agile company

Be **owned** by each Product – who are accountable for the outcome of their efforts

**Predict** successful software development delivering best business value for the buck

Provide **timely, actionable** data, so users can intervene and improve performance BEFORE its too late

**Focus on key high-value** indicators

**Easy to understand** – users should know how to drive for improved performance

**Balanced** with Velocity/Quality/Stakeholder satisfaction/other.

**Trigger positive outcomes**
in each group working with it

**Standardized** scale across company

Targets/Objectives should **be driven by context/situation** of each group

**Reinforce performance** by loosely connecting  to incentive systems?

**Stay relevant**. Focus on areas which still need work, Add more measures for future objectives when become relevant