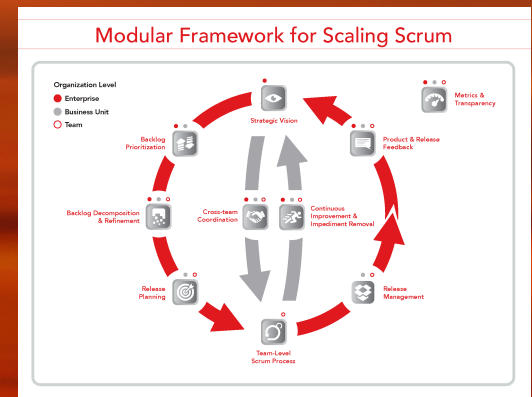


# Scrum at Scale

## Go Modular for Greater Success

*Agile 2014 - Orlando*



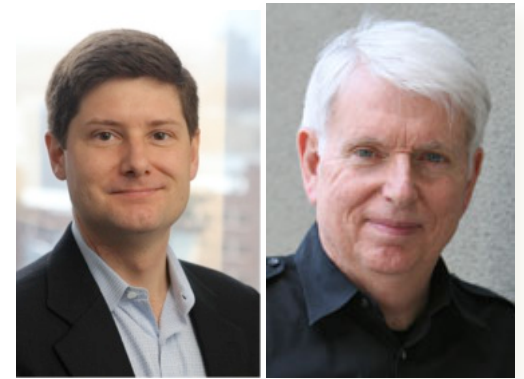
Hosts: Alex Brown  
Jeff Sutherland

# scruminc. Who We Are

Scrum Inc. is the Agile leadership company of Dr. Jeff Sutherland, co-creator of Scrum. We are based in Cambridge, MA.

We maintain the Scrum framework by:

- Capturing and codifying evolving best practices,
- Conducting original research on organizational behavior
- Adapting the methodology to an ever-expanding set of industries, processes and business challenges



We also help companies achieve the full benefits of Scrum through our full suite of support services:

- Training (Scrum Master, Product Owner, Agile Leadership, online courses, etc.)
- Consulting (linking Scrum and business strategy, customizing Scrum)
- Coaching (hands-on support to Scrum teams)
- Publishing and new content development

We run our services company using Scrum as the primary management framework, making us a living laboratory on the cutting edge of “Enterprise Scrum”

Find out more at [www.scruminc.com](http://www.scruminc.com).

# Agenda for Today

- Present a case for a modular scaling approach
- Lay out a high-level framework for discussing Scrum at the enterprise level
- Share several examples illustrating different scaled implementation approaches
- Challenge your thinking about what is possible with Scrum at the Enterprise-level
- Note: We will **NOT** be presenting a “paint by numbers” methodology that is claimed to work everywhere



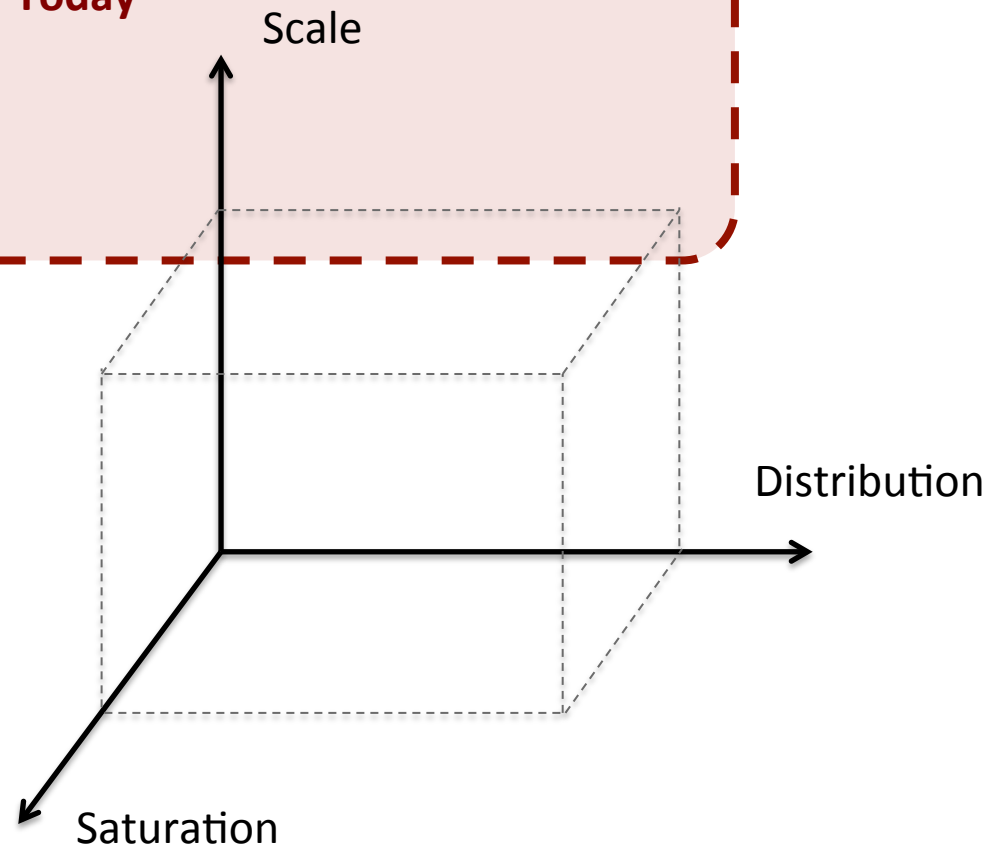
# Three Dimensions of Growing your Scrum

Focus for Today

**Scale** = number of coordinating teams;  
Complexity of projects

**Distribution** = number of different coordinated geographic locations

**Saturation** = Degree Agile principles have pervaded organization → Breaking down traditional “silos”



Improvements along any dimension will grow your Scrum

# The Case for a Modular Approach to Scaling

- 1 Need more general language to talk about Scrum at Scale
  - The world is diverse and Scrum is used in different contexts
  - Proscriptive methods work in some contexts, but not all
- 2 At its roots, Scrum is an Object Oriented Framework
  - Each role, artifact & ceremony defined by objectives, participants, inputs and outputs
  - Core Scrum allows for many different ways to achieve objectives within given input/output constraints
- 3 Modularity allows organizations to establish and improve Agile practices incrementally by focusing on one independent module at a time
- 4 Ultimately, supports “pattern library” of successful approaches that can be used in different contexts

# Context is Very Important, But Too Often Neglected in Discussions of Scaling Approach!

How important is speed of delivery?



How important is innovation?



How important is team empowerment?



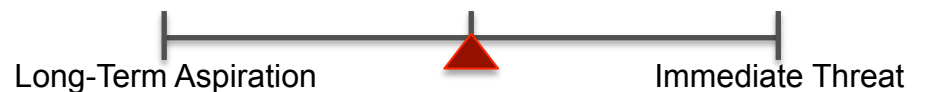
Where are teams located?



How complex and/or tightly integrated is the product?



What is the driving timeframe for becoming agile?



How severe are the repercussions of a product defect?



# We Will Use 3 Very Different Example Companies to Illustrate the Benefits of Modular Scaling

A

## Large Defense Contractor



Name Classified

- Top-down agile transformation motivated by perceived external market pressure
- Company vision to halve the cost of projects

### Key Context:

- Complex, integrated multi-year hardware/software projects
- Each project has one customer
- Reliability a key priority
- Must deliver to detailed contract requirements

B

## Mid-size Software Company



Autodesk

- Opportunistic agile implementation triggered by acquisition of a small Scrum company
- Market leader Looking to stay ahead of competition

### Key Context:

- Redeploying a legacy software product to cloud-based SaaS model
- Goal to increase pace of innovation
- Historically, releases a disruption for customers

C

## Growing "Agile Native" Company



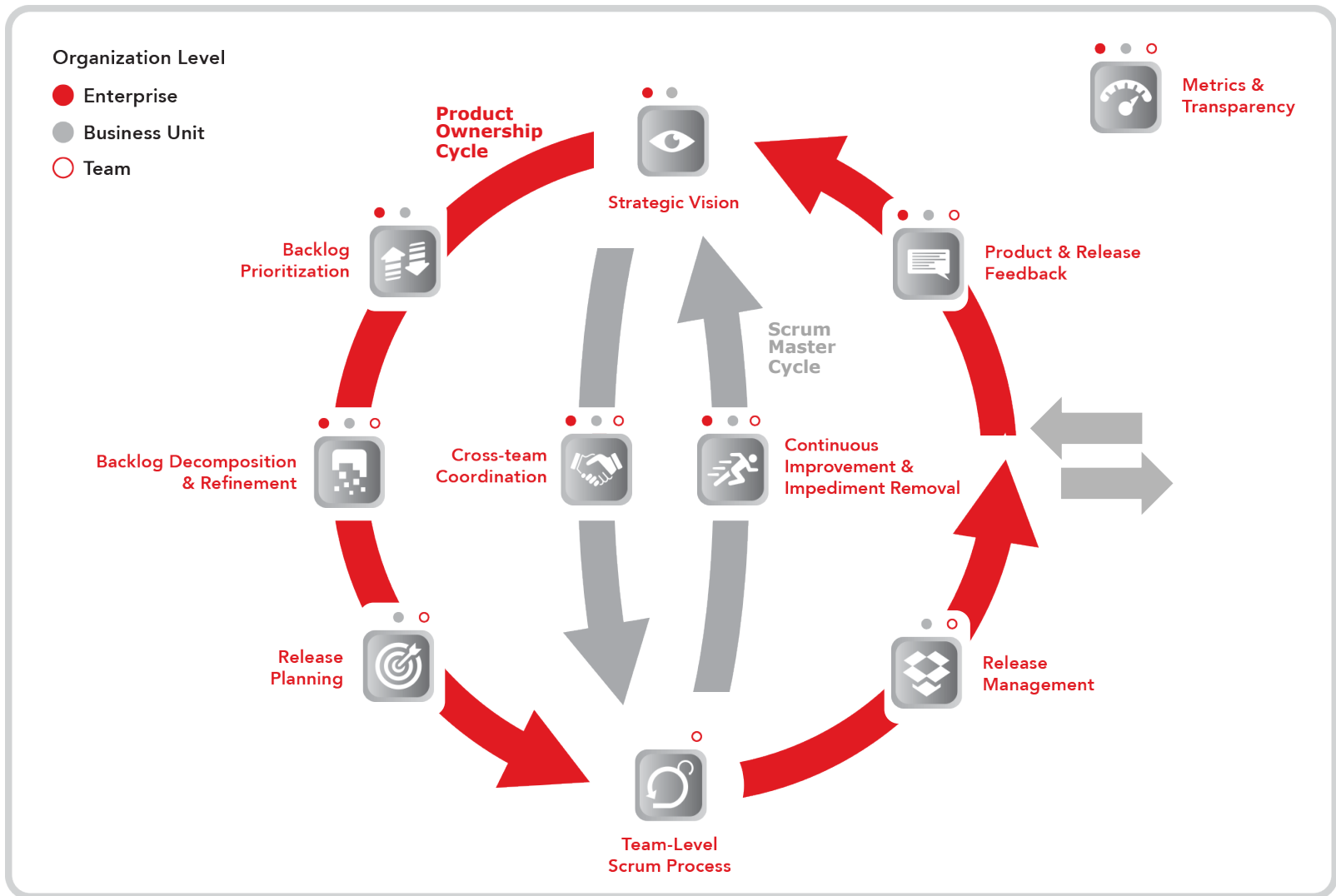
Spotify

- Disruptive technology innovator with successful product looking to scale to keep up with demand
- Leadership are steeped in agile principles

### Key Context:

- Web/app-based product
- Product and company set up modularly
- Allows teams to work independently with minimal coordination
- Teams co-located

# Modular Framework for Scaling Scrum



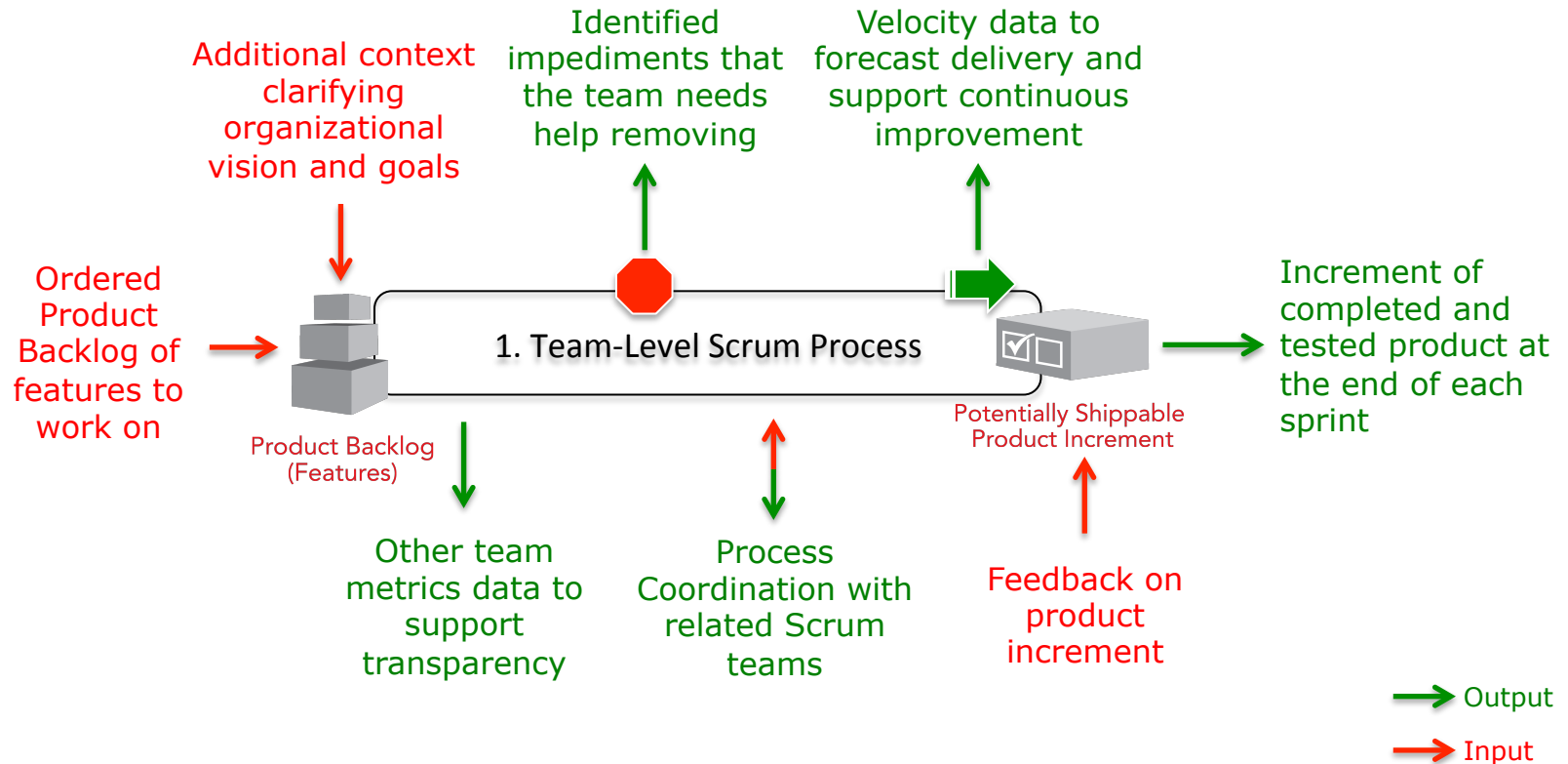




# 1. Team Level Scrum Process





## Module Goals:

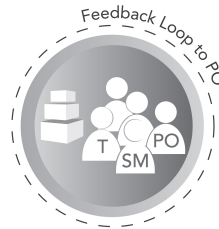
- Maximize the flow of completed and quality tested work
- Try to increase velocity a little each sprint
- Operate in a way that is sustainable and enriching for the team in the long run



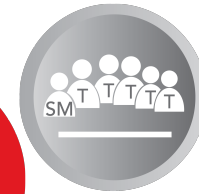


# The Team-Level Scrum Process

 PO - Product Owner  
 SM - ScrumMaster  
 T - Team  
 C - Customer

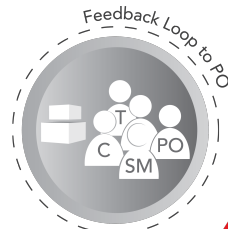


Product Backlog Refinement

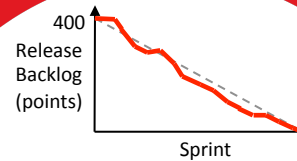


Daily Standup

24 hrs



Sprint Review



## Sprint 1-4 Weeks



Sprint Retrospective



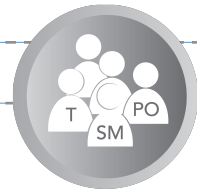
Product Owner



Product Owner



Product Backlog (Features)



Sprint Planning



Sprint Backlog (Stories)



Potentially Shippable Product Increment



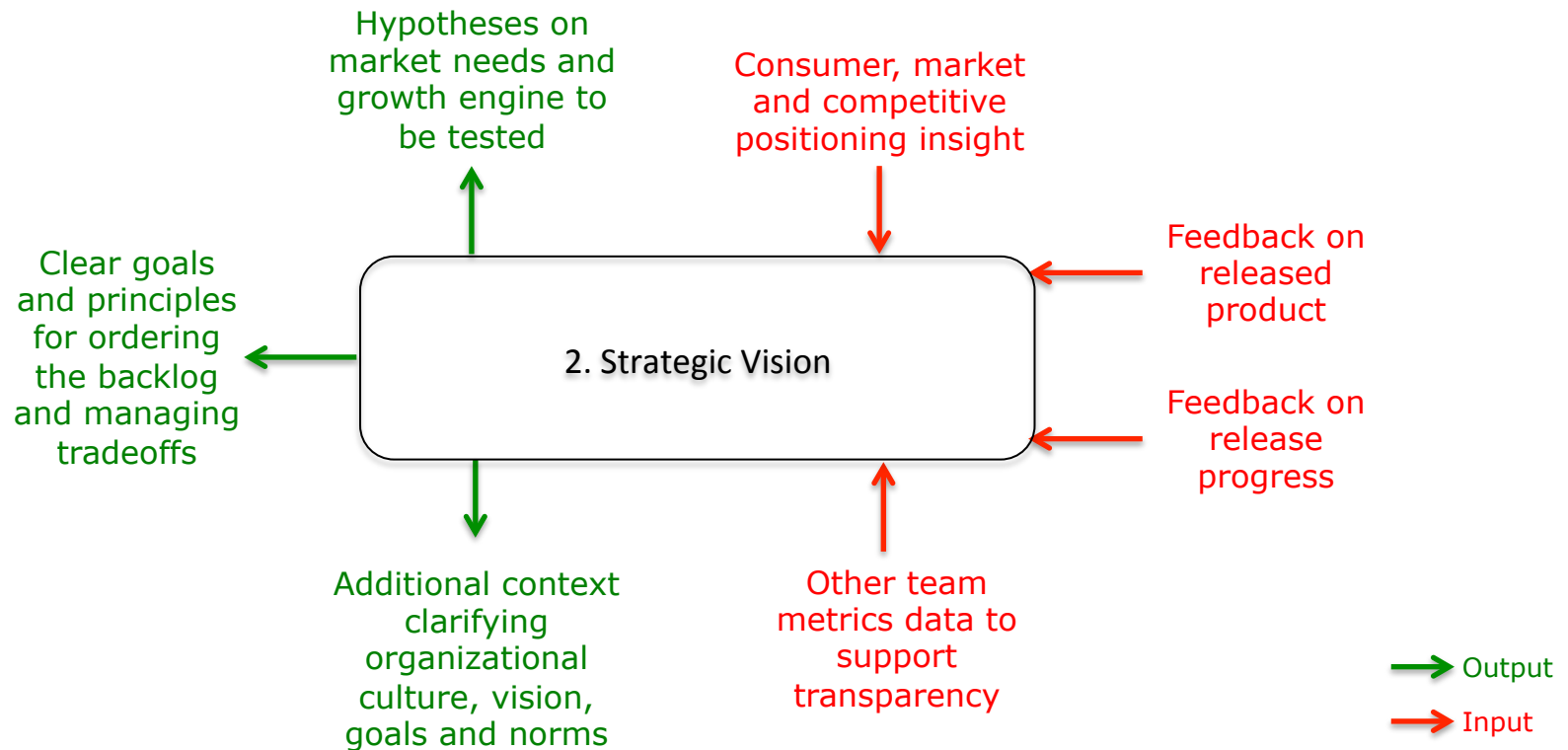
Incremental Product Release



## 2. Strategic Vision

### Module Goals:

- Clearly align the entire organization along a shared path forward
- Compellingly articulate why the organization exists
- Describe what the organization will and won't do to leverage key assets in support of its mission
- Update and fine-tune vision continuously based on feedback to outmaneuver the competition





# Alternate Approaches to Satisfy the “Strategic Vision” Module

A



## Contract Mgmt. Team

- Corporate vision still set and established in traditional model
- Vision includes goals to halve project delivery cost thru agile
- Corporate vision translated to project-level vision and goals through customer discussion & contract negotiation

**Pro:** Does not yet require large organization or customers to change what they are used to doing; meets core productivity goals

**Con:** Still very traditional “waterfall” process that limits ability to innovate faster using customer feedback

B



## PO Team

- Corporate leadership articulates enterprise-level vision and goals and updates to reflect market
- Chief PO for each product maps these goals to given product and maintains working vision that incorporates regular feedback and team discussion

**Pro:** Provides a highly centralized vision, while also responding to change and leveraging product/team-level input

**Con:** Still quite hierarchical and enterprise-level vision, in particular, not updated as frequently

C



## Empowered POs

- Strong culture of team empowerment & collective ownership
- Leadership articulates corporate “objectives & key results” quarterly
- “Tribes” of component teams work together facilitated by POs to interpret that vision at the component level

**Pro:** Lightweight approach; leadership focused on big picture only, and teams develop ownership of vision

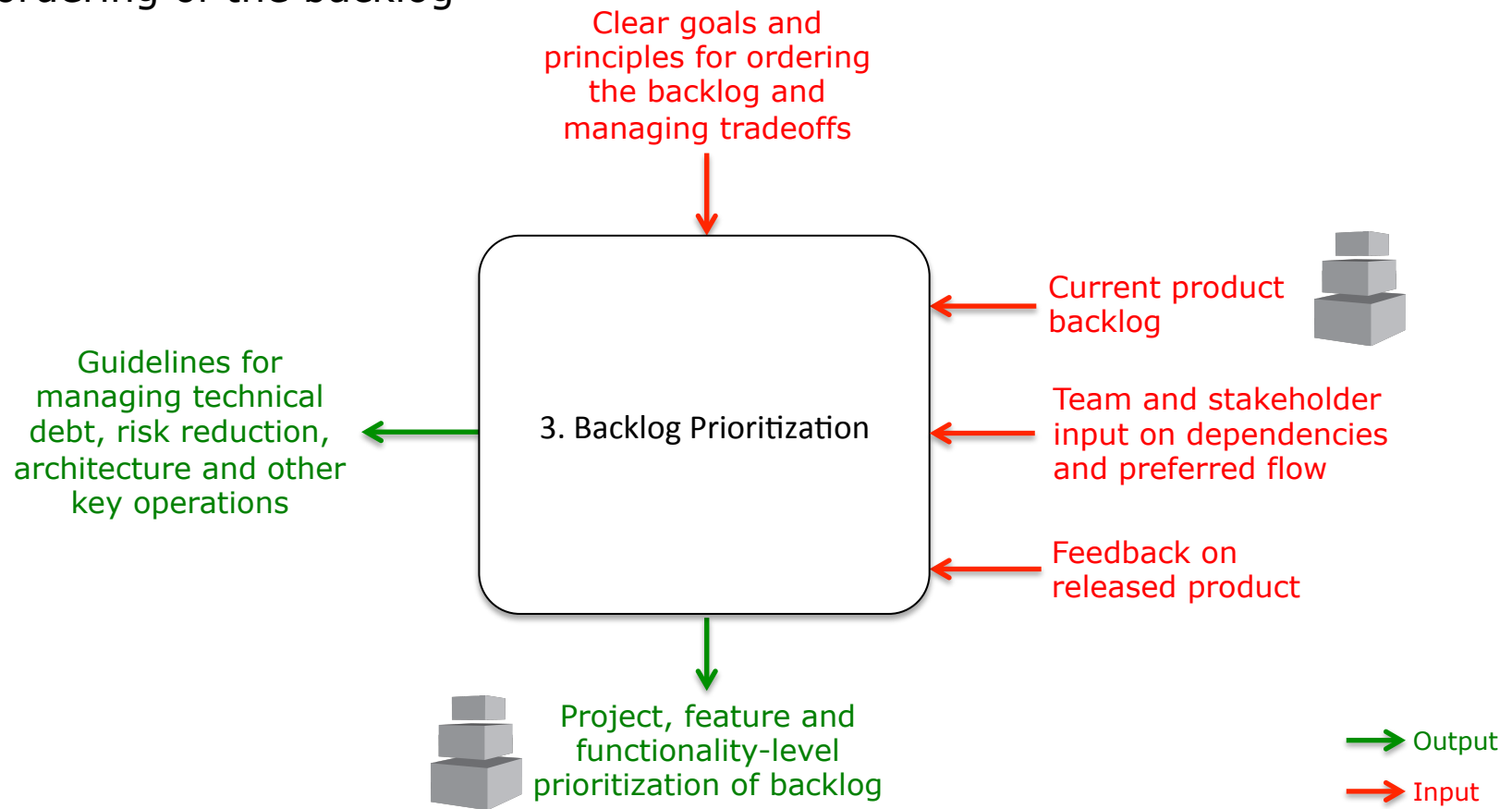
**Con:** Stronger potential for conflicting views on how to achieve objectives; Risk of sub-optimizing vision at component level



# 3. Backlog Prioritization

## Module Goals:

- Identify a clear ordering for products, features, and services to be delivered by the organization
- Reflect value creation, risk mitigation and internal dependencies in ordering of the backlog





# Alternate Approaches to Satisfy the “Backlog Prioritization” Module

A



## Contract Mgmt. Team

- Dedicated contract management team converts initial contract requirements into backlog and prioritizes to reduce risk and meet contract milestones
- Additional emerging requirements vetted and inserted at appropriate point in backlog

**Pro:** Works with government contracting requirements; provides centralized control over highly-interconnected product

**Con:** Much slower and less responsive to change; does not harness knowledge of working teams in prioritization

B



## PO Team

- One Chief Product Owner ultimately owns results for whole product, but works with POs for each team and component as well as stakeholders to prioritize backlog.
- Regular “Meta Scrum” meeting to assemble all stakeholders and align on priorities.

**Pro:** Provides a degree of centralized vision, while also responding to change and leveraging team-level knowledge/autonomy

**Con:** Requires more overhead, discipline and buy-in from stakeholders than empowered POs

C



## Empowered POs

- Leadership articulates “objectives & key results”
- Components independent enough for component POs to decide priorities for their teams w/only informal cross-components coordination
- Projects with greater coordination need have regular meeting cadence

**Pro:** Can be Extremely fast; very little overhead; allows each component to deliver its value-maximizing backlog

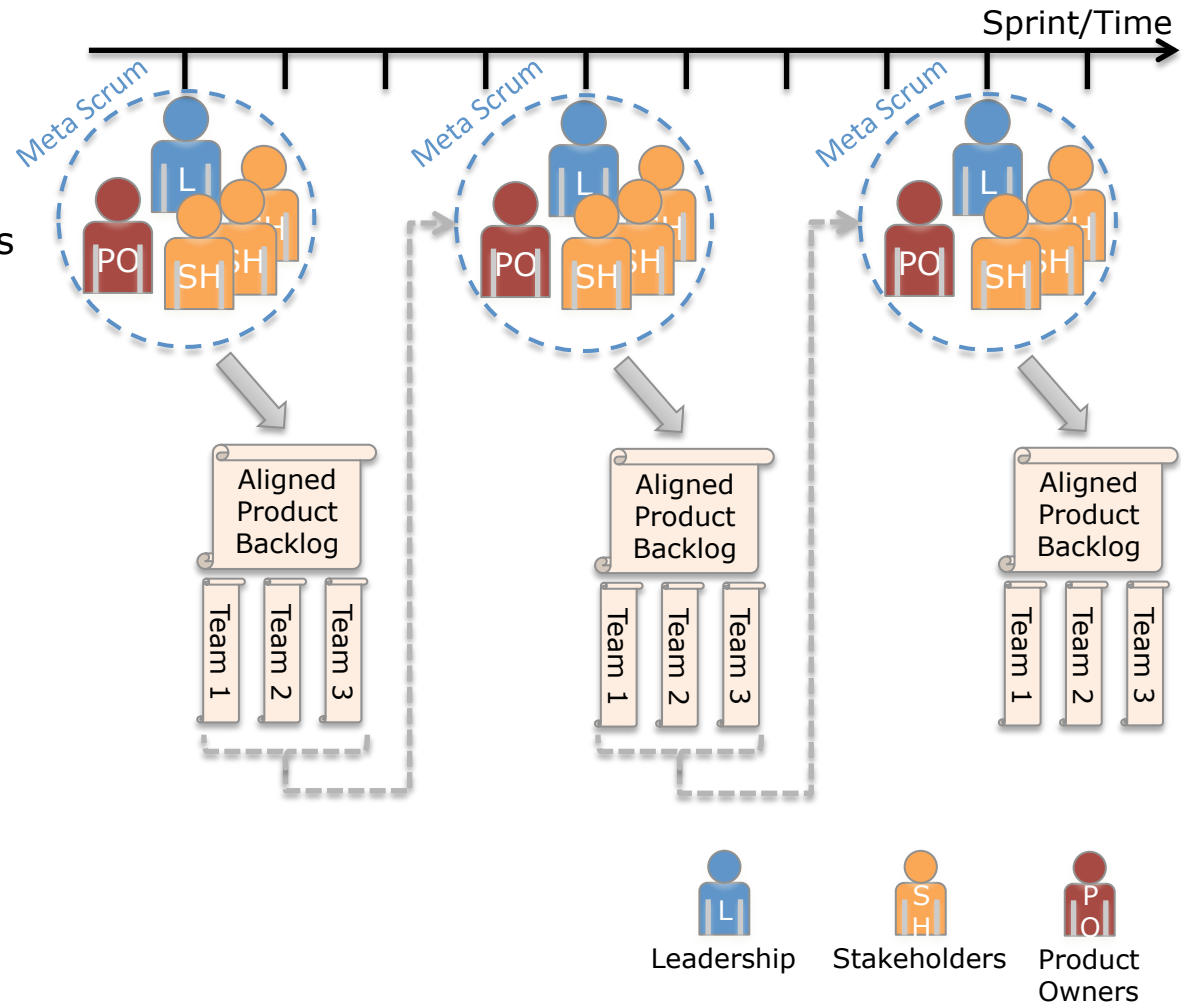
**Con:** Requires product and enterprise to be architected around independent modular components; some potential for divergent priorities



# The Meta Scrum:

## Used to Align Organizational Priorities

- A gathering of key Stakeholders, Leadership, and Product Owners
- Run by Chief Product Owner
- The forum for stakeholders to express preferences (they should not lobby teams directly or try to alter product vision between Meta Scrums)
- Can be held at regular intervals or on an ad-hoc basis
- Allows teams to progress efficiently down a single work path

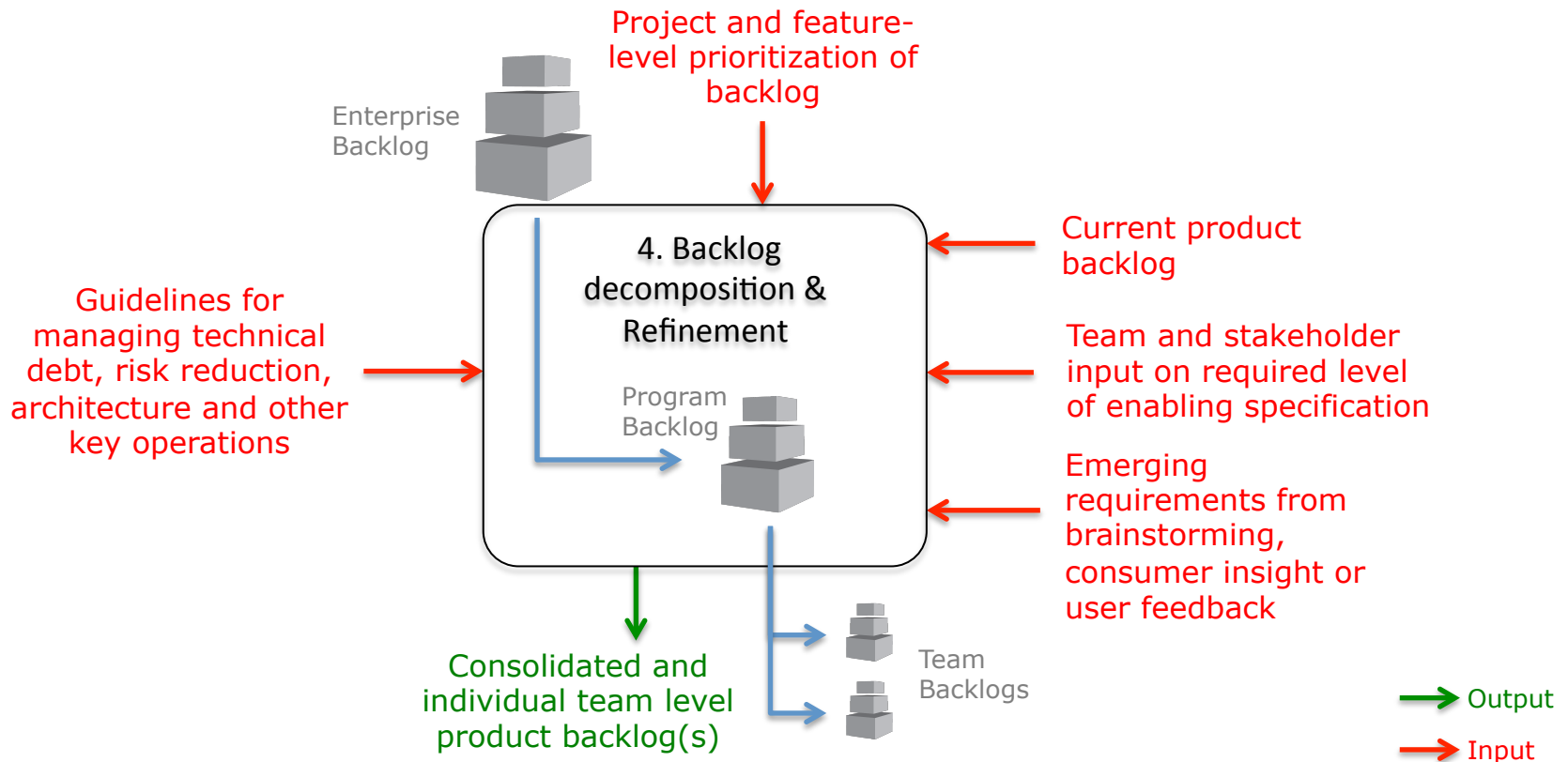




# 4. Backlog Decomposition & Refinement

## Module Goals:

- Break complex projects and products into manageable independent functional elements that can be completed by one team in one sprint
- Capture and distil emerging requirements and customer feedback
- Ensure all backlog items are truly "Ready" when they reach sprint backlog
- Parse backlog to individual teams







# Alternate Approaches to Satisfy the “Backlog Decomposition” Module

A



## Contract Mgmt. Team

- Contract management team subdivides contract-level features and epics into user stories in consultation with engineering and technical SMEs at regular refinement meetings
- Contracts team available to development teams to answer intent questions

**Pro:** Provides centralized control for contract compliance over highly-interconnected product; matches contract needs with team expertise

**Con:** Requires significant overhead structure; involves less input from working teams

B



## PO Team

- Product divided into logical “components” each with a PO team
- Chief PO articulates and signs off on Epic-level goals, and clear DoD
- Component PO teams subdivide & refine to team-level backlog
- Team POs own “Ready”
- Weekly grooming meeting

**Pro:** Structured and deliberate process that ensures stories flow from concept to execution and are ready for the team; accommodates and incorporates product feedback

**Con:** Requires more overhead and discipline to execute

C



## PO/Team Partnering

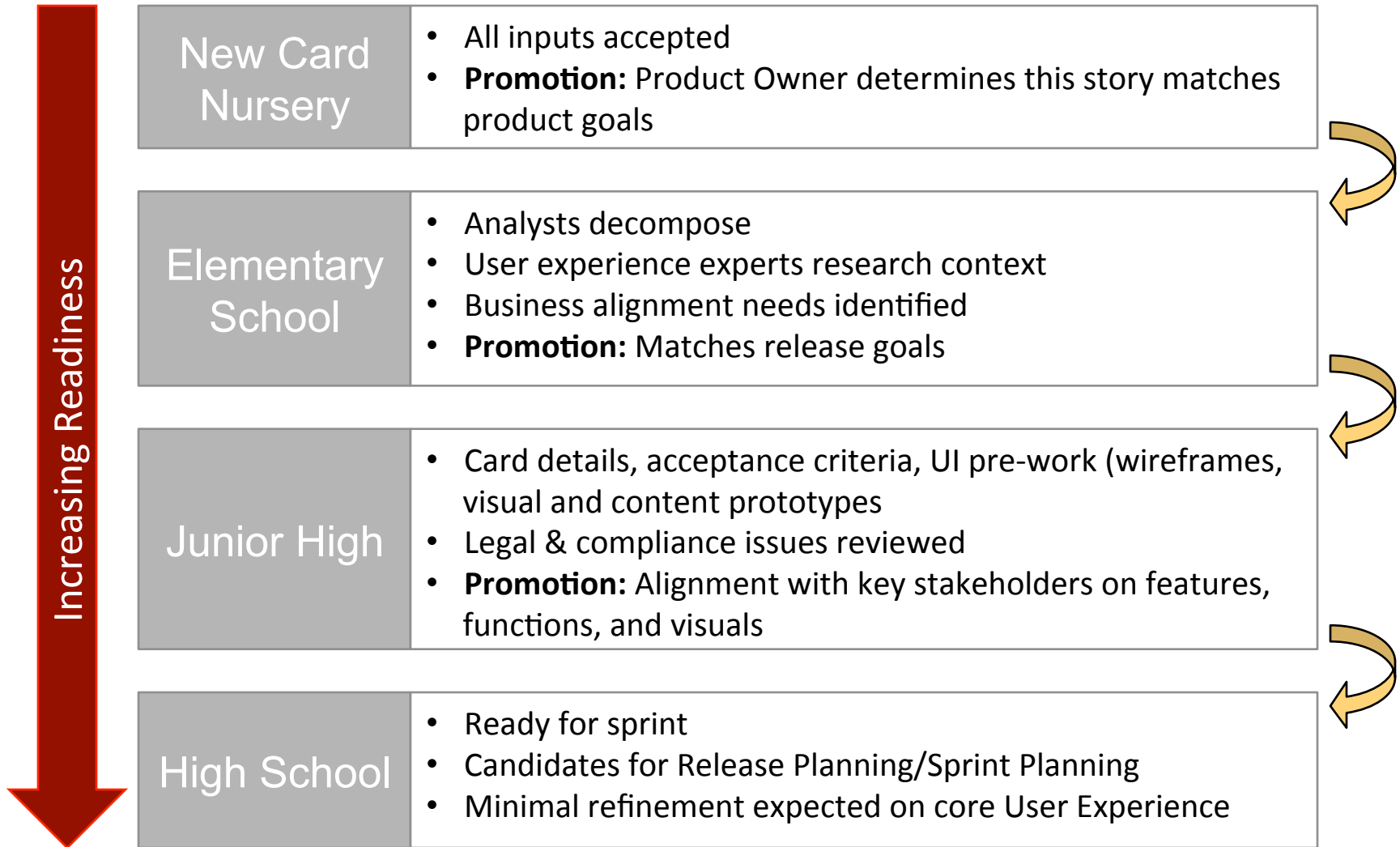
- New stories created at the component “Tribe” level
- PO team works closely with team to create, segment and refine stories to “ready”
- PO notionally responsible for ready backlog, but Team does most of the work

**Pro:** Can be relatively fast if consensus can be achieved; really empowers Team; largely eliminates team confusion about what is needed

**Con:** Greater risk of divergent stories between components; relies on strong culture of collective ownership



# User Story Readiness Progression

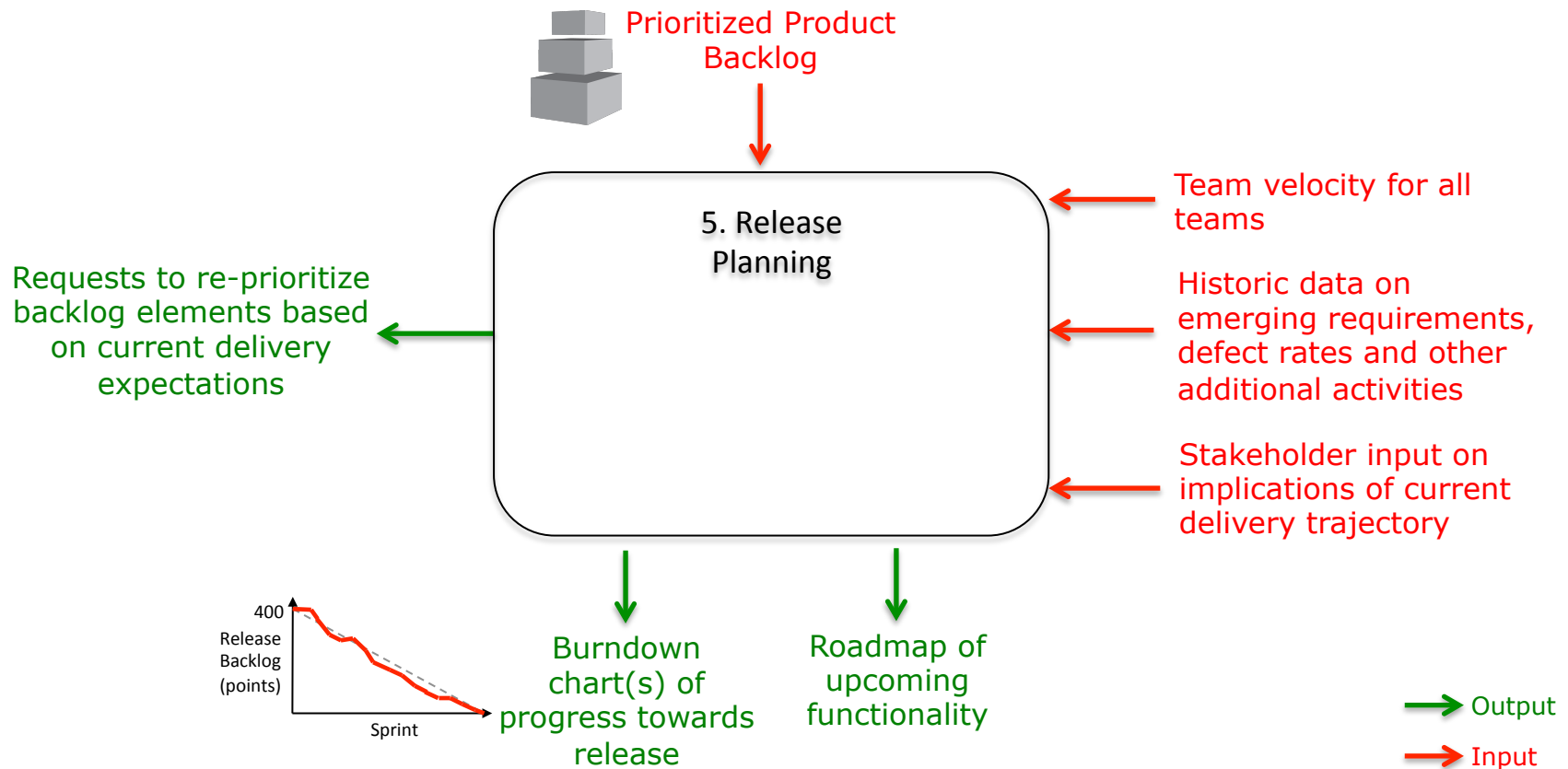




# 5. Release Planning

## Module Goals:

- Forecast delivery of key features and capabilities
- Communicate snapshot of delivery expectations to stakeholders
- Inform updated prioritization, as needed, based on stakeholder input





# Alternate Approaches to Satisfy the “Release Planning” Module

A



## Tightly Managed Deliverables

- Contract management team outlines and verifies feasibility of meeting contractual release milestones
- Monitors burndown progress and emerging requirements
- Identifies “at risk” deliverables early and negotiates responses

**Pro:** Better than traditional waterfall planning because forecasts based on actual progress, and interventions can happen much earlier.

**Con:** Still relatively rigid, hierarchical, and not as responsive to new learnings

B



## Release Train Burndown

- Product Owner team meets regularly to:
  - Discuss progress
  - Update release plan
  - Re-prioritize backlogs as needed to align complementary functions for quarterly releases
- Stakeholders updated of any changes

**Pro:** Straightforward way to plan releases that align key dependencies across teams and provide transparency to all teams and stakeholders

**Con:** Process not automated; Requires more overhead than independent release approach

C



## Stakeholder Transparency

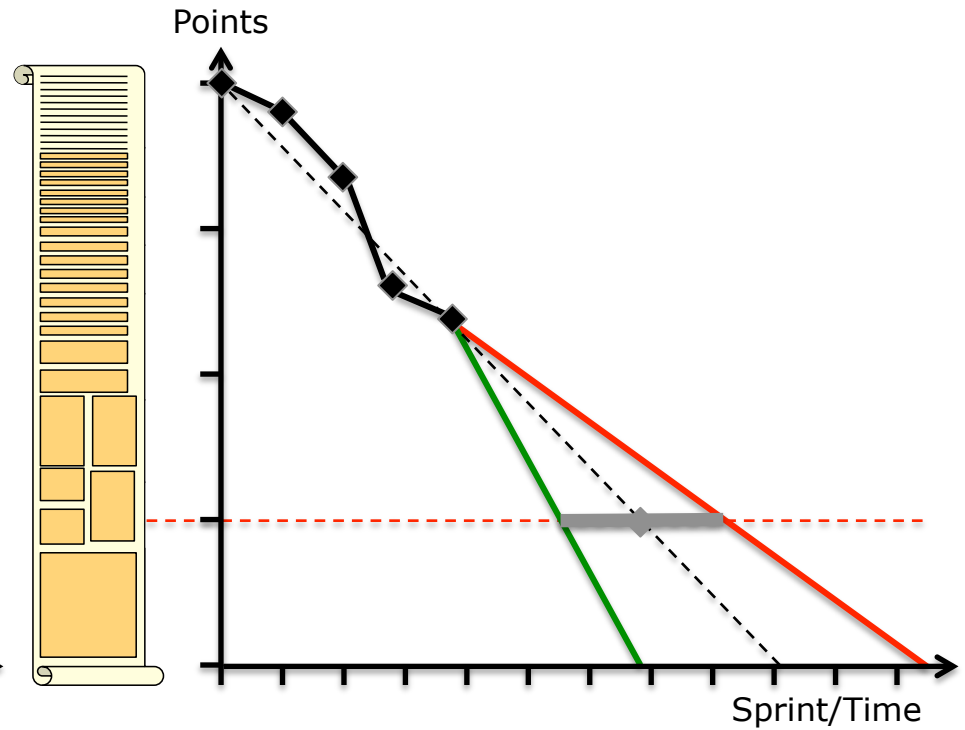
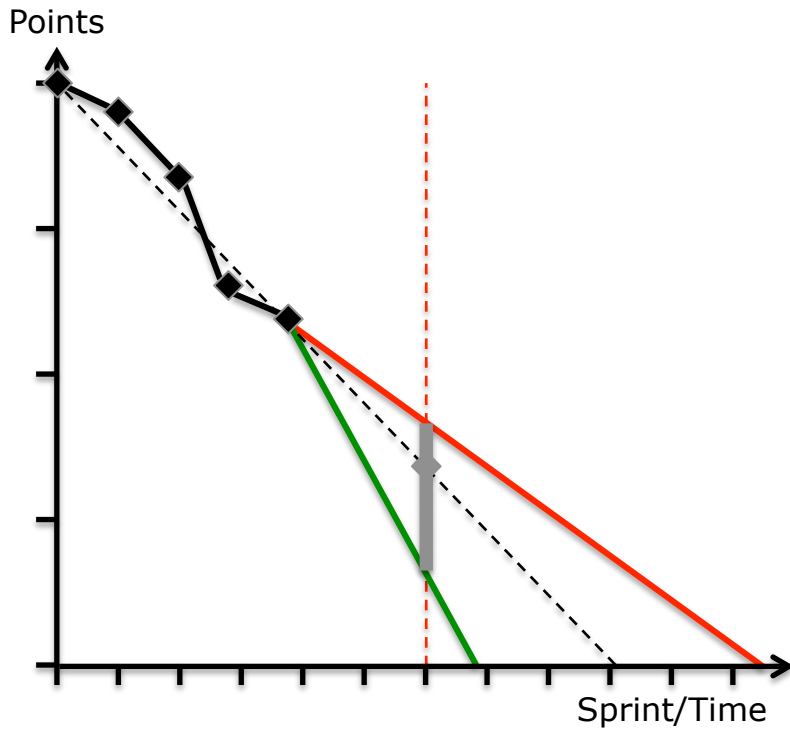
- Team Product Owners update metrics and backlog at end of each sprint
- Individual team tools and information radiators available to anyone
- Provides visibility, if stakeholders disagree with current plan, they can raise concerns

**Pro:** Provides transparency for all stakeholders; low overhead for teams and POs

**Con:** Requires product modules to be largely independent; not systematic across all teams; burden of proof for identifying conflicts falls on stakeholders



# Release Burndown Chart Makes Team's Velocity and its Implications Visible

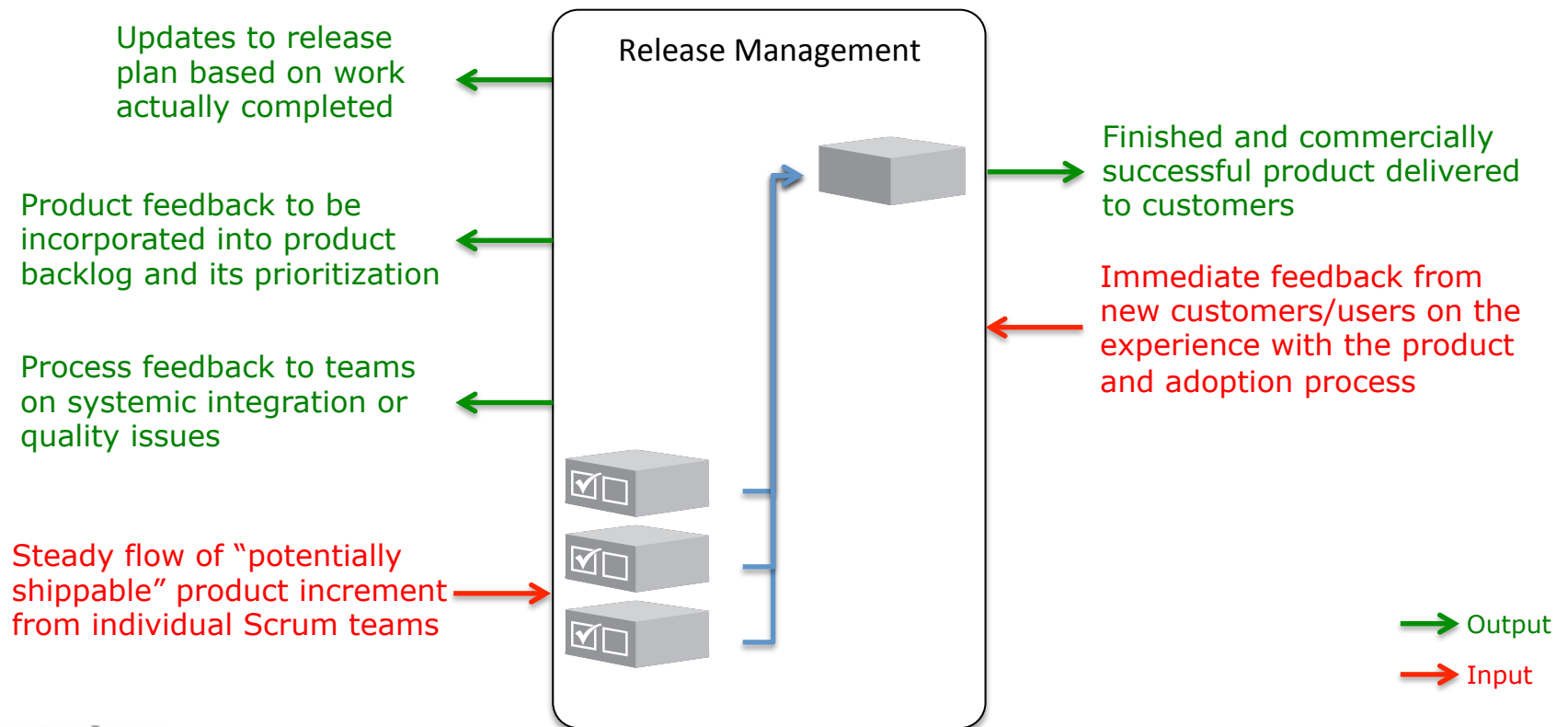




# 6. Release Management

## Module Goals:

- Deliver a consistent flow of valuable finished product to customers
- Integrate the work of different teams into one seamless product
- Ensure high quality of the customer experience
- Capture and communicate feedback on product, process and schedule





# Alternate Approaches to Satisfy the “Release Management” Module

A



## Milestone Based

- Release is based on a pre-defined feature set
- Often driven by a set target delivery date
- Larger clusters of functionality delivered at once
- Product is not released until all required features are available
- “Hardening” sprints

**Pro:** Necessary for certain contract types, tightly-integrated product designs, or difficult customer adoption processes

**Con:** Less responsive to new learning or minor setbacks. Stressful to try and constrain both scope and delivery date.

B



## Release Train

- Same dev teams release
- Product release internally each month, big internal releases quarterly
- Features that are ready in time for the release are included, otherwise they wait for the next release
- Release to customers on annual cadence, with goal to move to quarterly

**Pro:** Straightforward way to manage releases that removes the stress of deadlines and more manageable process for customer

**Con:** Harder to do with tightly coupled products. Requires more overhead than independent releases

C



## Independent Releases

- As new independent functionality is judged “ready” it is released directly to customers
- Releases can happen multiple times a day
- All features have on/off toggle to allow rapid roll-back in case of issues.

**Pro:** Allows for extremely rapid product development and low overhead for product releases

**Con:** Requires product modules to be independently defined with little need for integration with other team’s product (e.g. web pages)



# Three Common Approaches to Release Management



- **Deadline-based**
  - External deadline specified for team, they must complete as much of a given backlog as possible before that date



- **Regular-Departure**
  - Set cadence of product releases. (e.g. quarterly)
  - Ready features are included in the release, non-ready ones wait for next release



Incremental Product Release

- **Value-Based**
  - Team produces incremental potentially-shippable product each Sprint
  - When PO decides enough new value has been created, features are released to customers

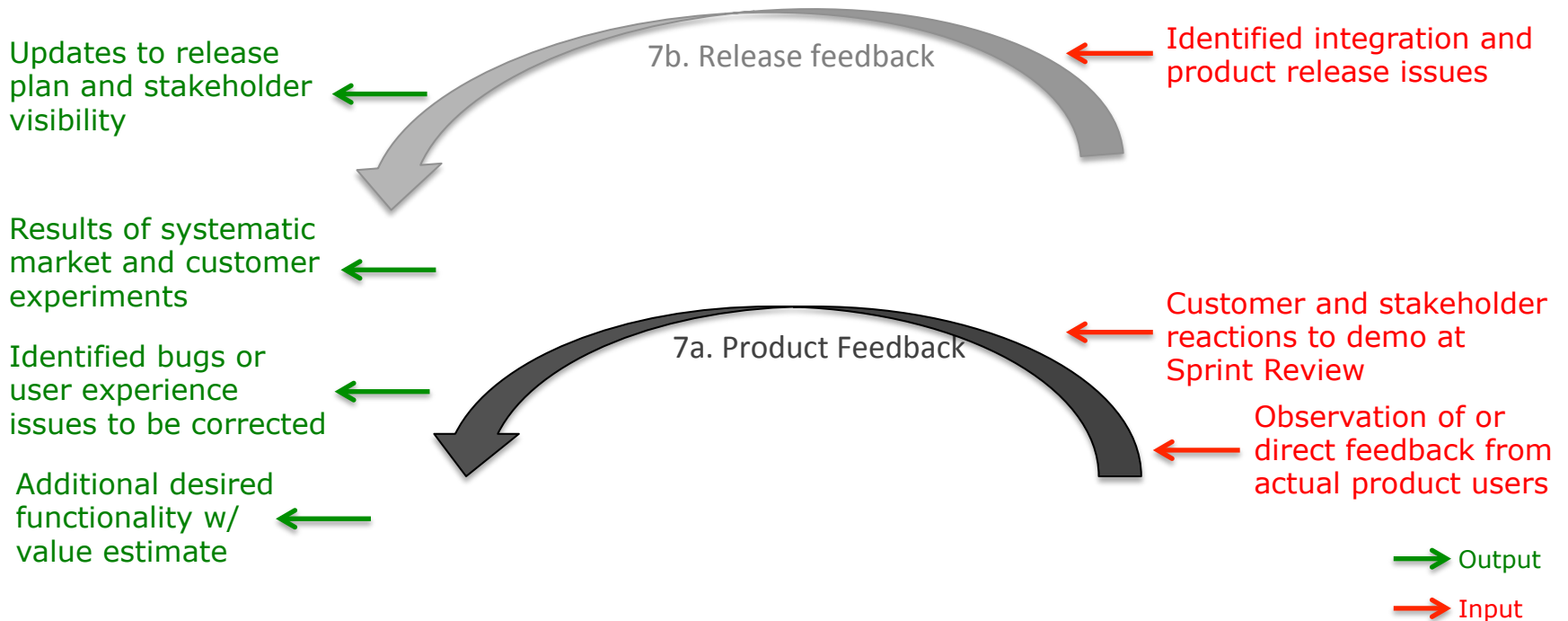




# 7. Feedback

## Module Goals:

- Understand how customers actually use and interact with the product
- Define improvements to existing functionality
- Distil actionable changes in direction from the noise of all responses
- Capture ideas for new features and functionality not previously identified
- Update progress towards product/project completion to refine release planning and stakeholder alignment





# Alternate Approaches to Satisfy the “Feedback” Module

A



## Structured Feedback

- Representatives from single customer invited to view intermediate internal release product and provide feedback
- Customer relationship team captures feedback and works with contracts team to determine how best to incorporate into backlog

**Pro:** Provides regular and clear feedback channel for customer to register feedback; works with contract requirements

**Con:** Hard to scale beyond a single customer; feedback has limited impact on enterprise vision or product design

B



## PO Filtration

- Product feedback gathered and categorized from customer service, test customers at demo meetings, customer discussions, stakeholders, trade press
- ALL feedback flows through Chief PO, who is charged with distilling product insight

**Pro:** Single-point of integration helps to resolve conflicting feedback or teams pulled in different directions; maintains an integrated product view

**Con:** Heavy burden on CPO, who must be skilled to understand all product, market and technical needs

C



## Direct Feedback

- In-App feedback button, online product reviews and bug ticketing system feed directly back to right component team
- Teams use different tools to collect, process and pareto feedback
- Teams review frequently with PO in determining new component backlog

**Pro:** Streamlined & lightweight system for channeling feedback; lets each team use an approach that work for their needs

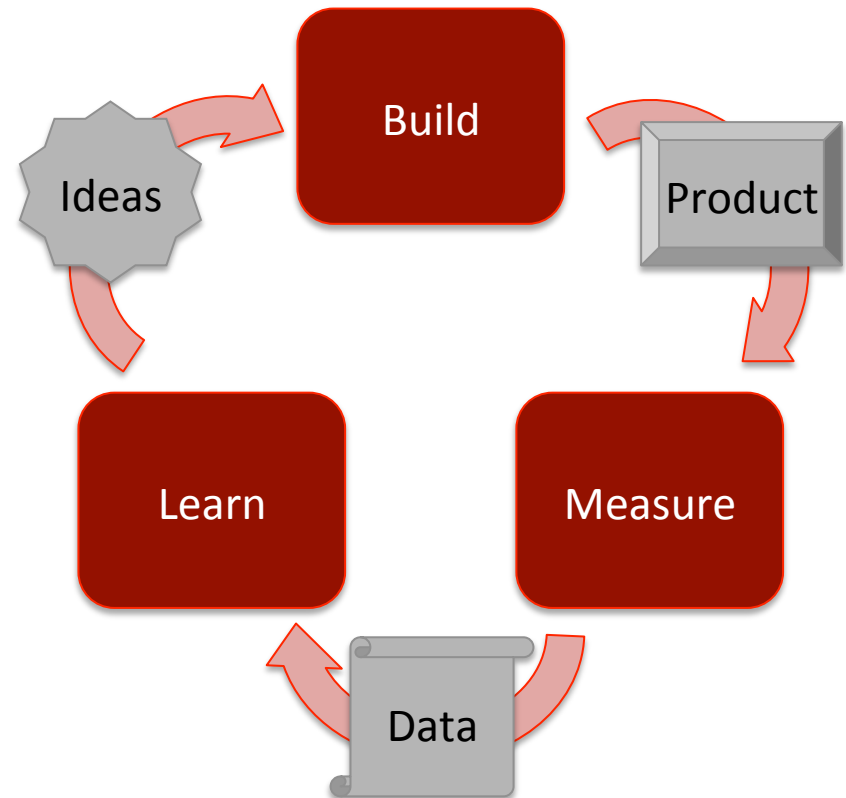
**Con:** May miss systematic feedback across multiple components; Does not necessarily seek out input on totally new functionality



# Feedback is About Distilling “Validated Learning”

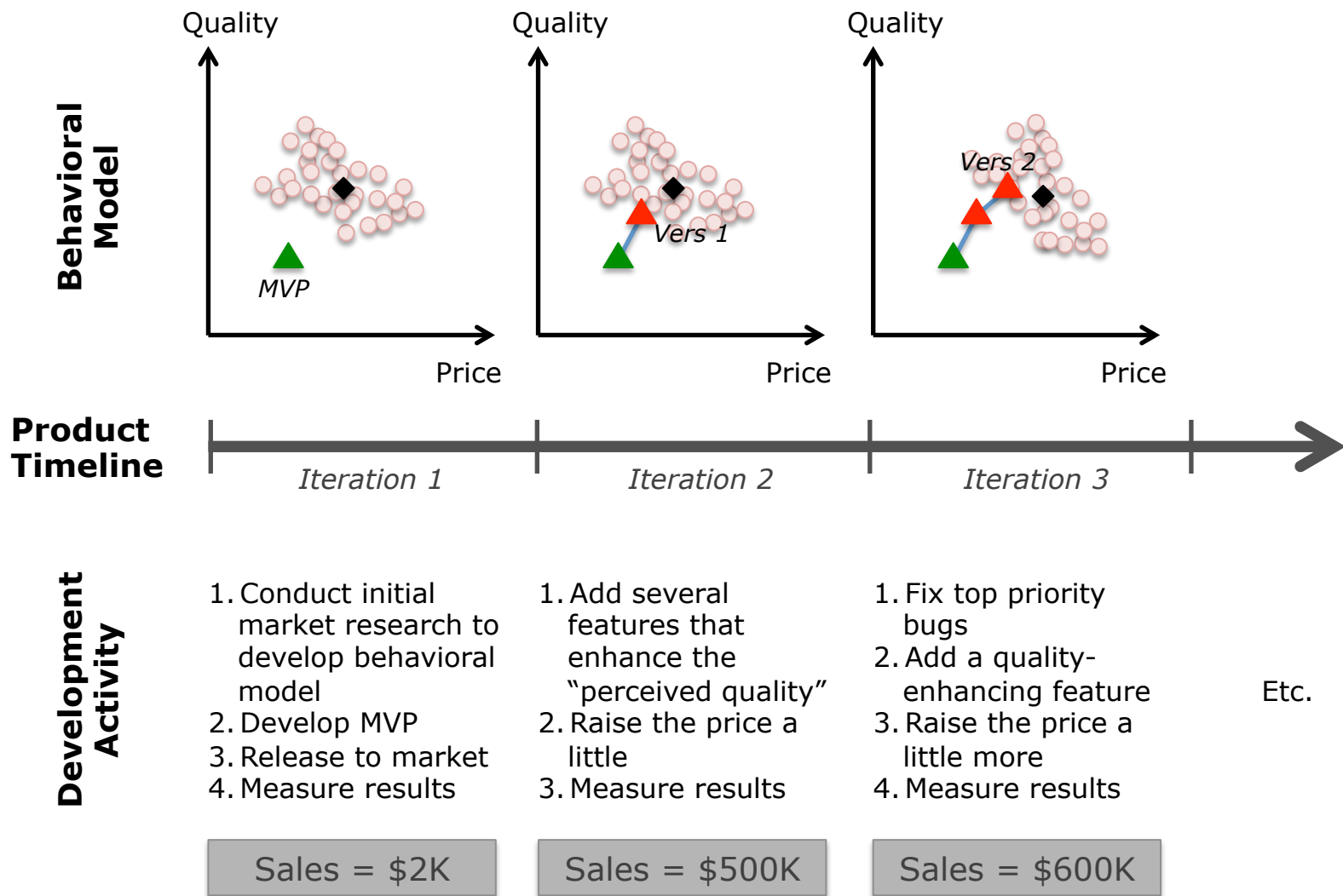


- Cast your business case as a set of **assumptions**
- Rapidly build prototypes for early adopters to **validate those assumptions**
  - *“Get out of the building.”*
- “Pivot” releases based on both **qualitative & quantitative feedback**
- Deliver quickly, often & with high quality using **agile methods**





# Use Feedback Loop to Update Strategic Vision

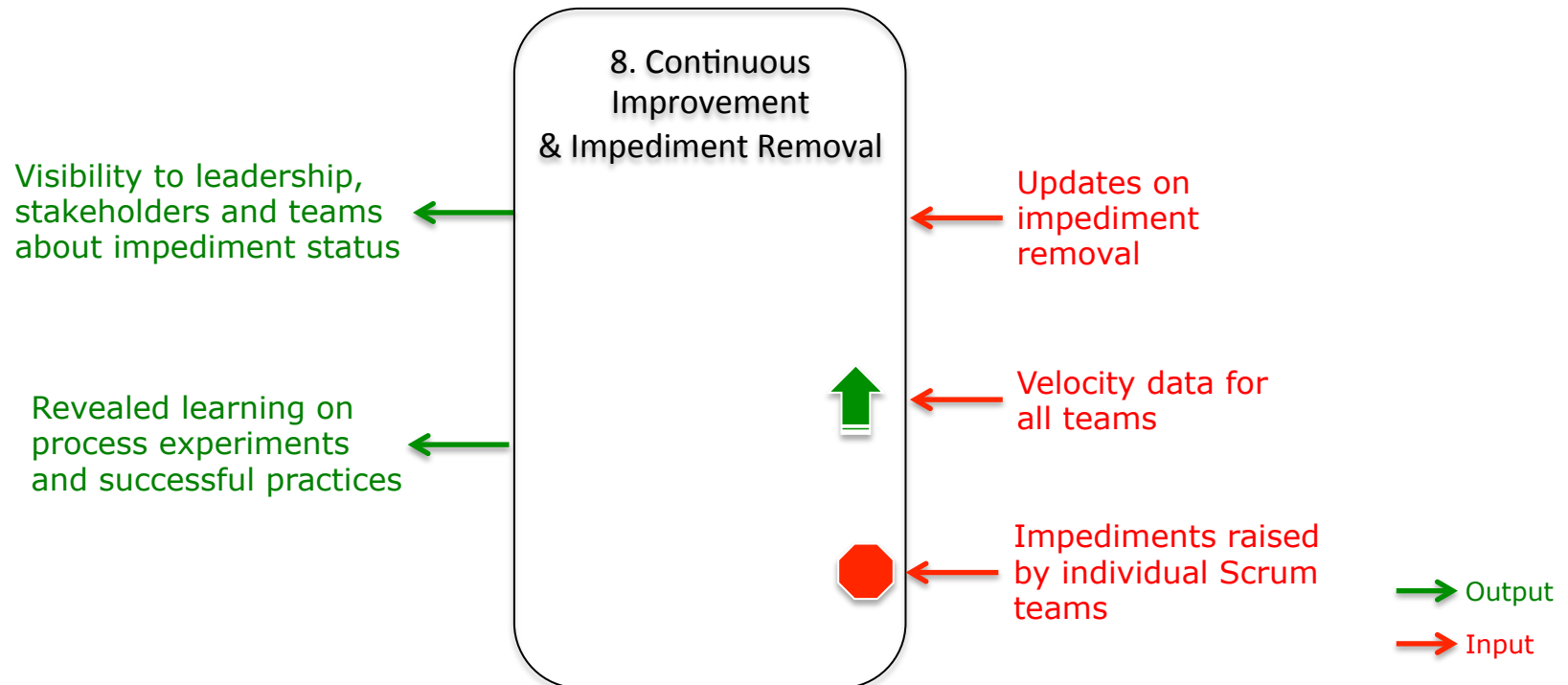




# 8. Continuous Improvement and Impediment Removal

## Module Goals:

- Identify impediments that slow teams down and reframe them as opportunities to get faster
- Maintain a safe and structured environment for prioritizing and removing impediments, and then verifying the resulting improvement
- Ensure visibility at the right level(s) in the organization to effect change





# Alternate Approaches to Satisfy the “Continuous Improvement” Module

A



## Agile PMO

- Individual teams identify impediments
- Impediments discussed at regular Scrum of Scrums, and escalated if needed
- “Agile PMO” is available to support removal of corporate, contract, or systematic impediments
- Agile PMO logs and tracks impediments

**Pro:** Structured process to provide teams with support to remove impediments; provides audit trail for ISO and contract requirements

**Con:** Involves greater overhead; in practice, has a mixed record removing impediments in a timely way

B



## Escalation with Exec. Support

- Individual teams identify impediments
- Impediments discussed at regular Scrum of Scrums, and escalated if needed
- Executive “sponsor team” tasked with removing major impediments fast
- Systemic impediments referred to functional “Centers of Excellence”

**Pro:** Traditional escalation model for removing impediments; teams get support, but impediments removed at lowest level possible

**Con:** Requires greater overhead in terms of meetings and staffing; can take time for impediments to percolate up

C



## Flexible

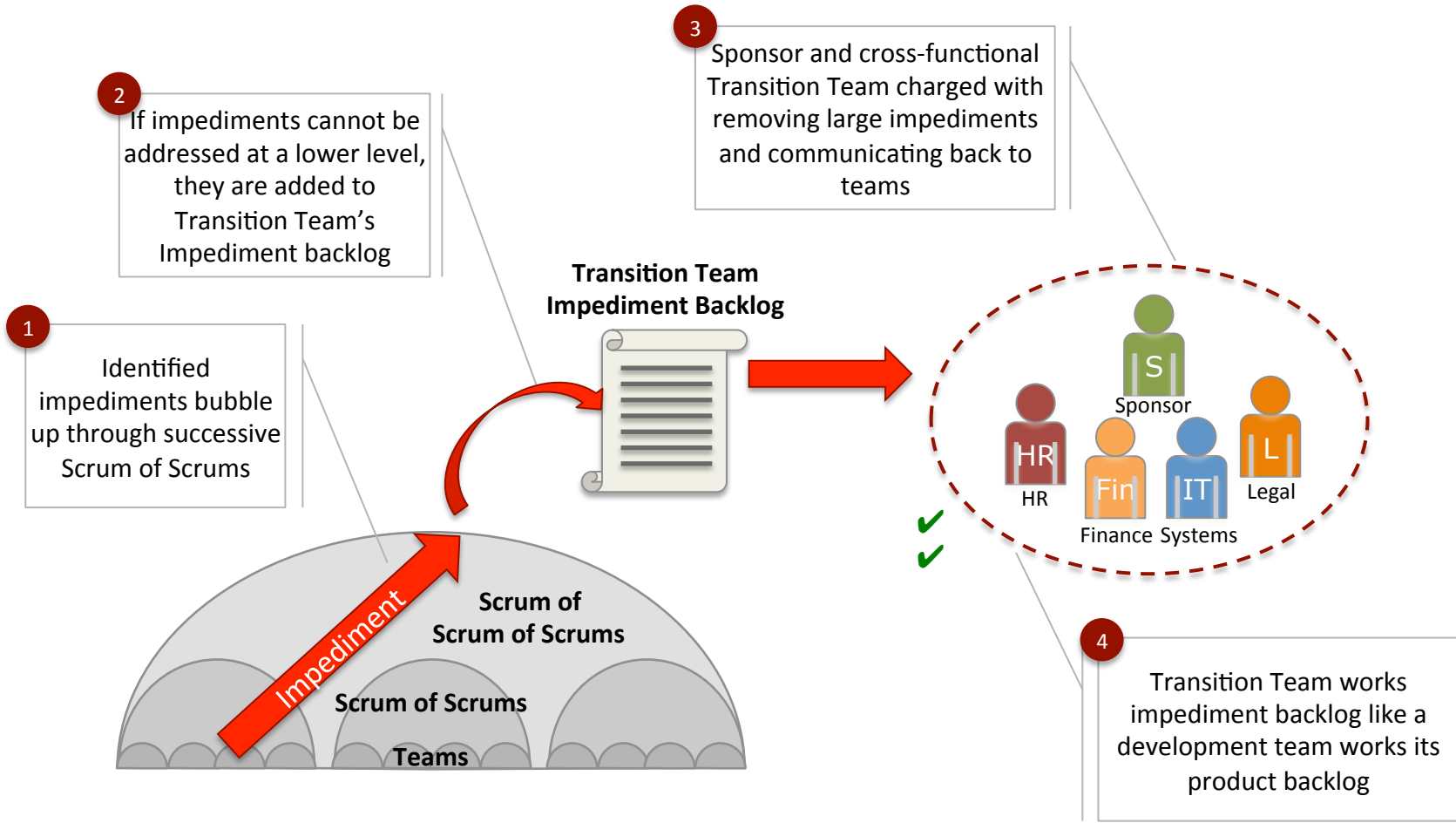
- Individual teams identify impediments
- Cross-cutting issues can be discussed in “chapters,” “guilds”, ad hoc, or with team’s executive mentors
- Culture of continuous improvement encourages employees to help resolve team impediments

**Pro:** Very informal approach allows for different solutions to different impediments; reinforces culture of collaborative empowerment

**Con:** Little formal structure can make it difficult to recall what was or wasn’t done; depends on supporting culture



# How the Sponsor Team Works

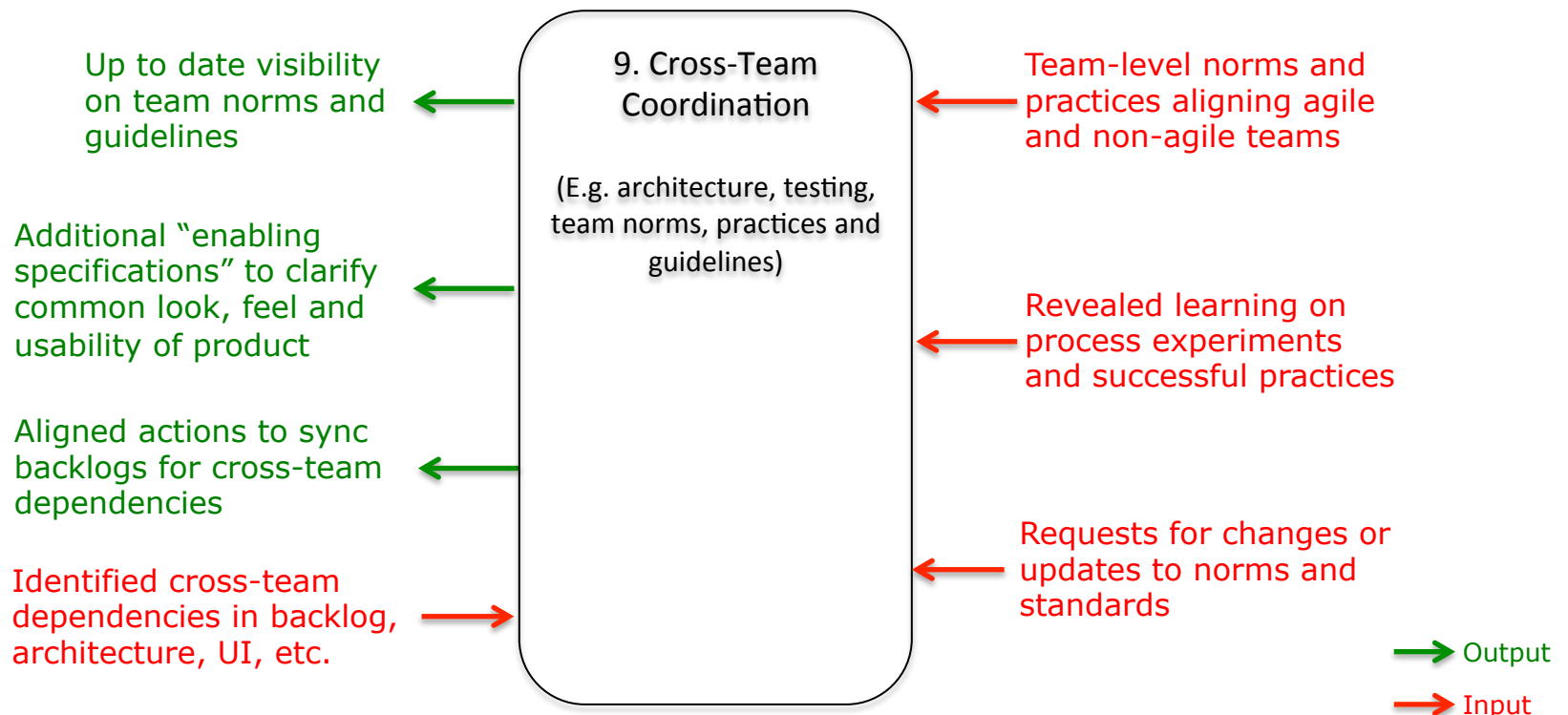




# 9. Cross-Team Coordination

## Module Goals:

- Coordinate similar processes across multiple related teams
- Manage cross-team dependencies to ensure they don't become impediments
- Maintain alignment of team norms and guidelines for consistent output







# Alternate Approaches to Satisfy the “Cross-Team Coordination” Module

A

## Scrum of Scrums

- Regular coordination meeting on a cadence agreed by participants
- All participants are peers in a Scrum of Scrums
- Not just for SMs! UX, architects, testing hardware, writing, etc. can also hold regular SoS

**Pro:** Lightweight and flexible to accommodate a range of different needs. Good for day-to-day coordination

**Con:** Does not provide sufficient resources for major issues or sustained coordination work

**Ongoing “light-touch” coordination**

B

## “Guilds” or “Scrumlets”

- Temporary team formed across other teams to address a specific issue
- Teams are cross-functional, and draw needed expertise from across wide range of skillsets

**Pro:** Very helpful for tackling important but short-lived issues or challenges. Does not commit resources in long term

**Con:** Significant time commitment for duration of Scrumlet. Not suitable for sustaining long term standards

**Specific near-term issues**

C

## Communities of Practice

- Standing overlay organization of team members with related functional experience
- CoP maintains shared norms, guidelines and standards
- At least one identified “owner” of the CoP

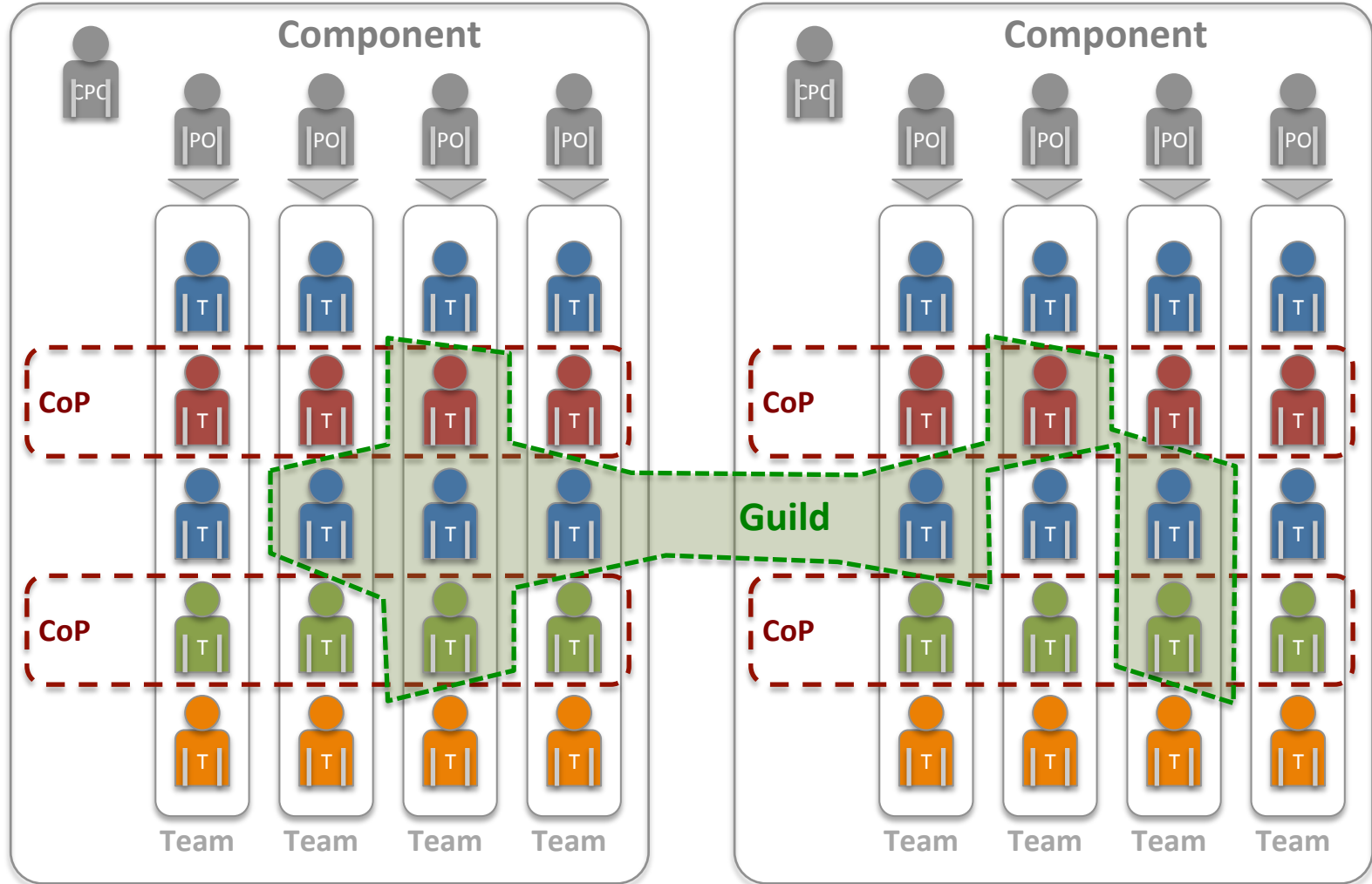
**Pro:** More formal, long-lived and resourced organization useful for maintaining key standards used by many groups

**Con:** More resource-intensive than Scrum of Scrums. Adds more hierarchy to organization

**Maintaining important standards**



# Different Cross-Team Coordination Mechanisms Serve Different Purposes



Adapted from: Scaling @ Spotify, Anders Ivarsson & Henrik Kniberg, Scrum Alliance Gathering Paris, 6 Feb 2013



# Alternate Approaches to Coordinate Agile and Non-Agile Teams

A

## Dedicated cross-coordination team

- Defined team of key stakeholders and Product Owners (Project Managers) from relevant groups
- At least 50% of their time allocated to ensuring smooth coordination
- Team self-organizes to decide how to achieve coordination (meeting frequency, agenda, etc.)

**Pro:** Clear responsibility, focus and accountability

**Con:** More resource intensive and time consuming

**Process Focus**

B

## Addressed at Regular Meta-Scrum meeting

- Regularly scheduled meeting of all key stakeholders
- Cadence determined by stakeholders
- All strategic, alignment and prioritization decisions made in the meeting (otherwise wait to the next meta-scrum)

**Pro:** Less resource intensive, aligns with sprint cadence

**Con:** Less familiar for non-agile stakeholders, lower emphasis on agile/non-agile coordination

**Vision Alignment Focus**

C

## Automated and ad hoc coordination

- Effective dashboard of progress metrics, release plans, impediments automates transparency
- POs, SMs non-agile Project Managers and Stakeholders know who their counterparts are
- Individual teams responsible for reaching out with announcements, impediments, as needed

**Pro:** Very quick and efficient

**Con:** Requires more tooling, and high-performing agile implementation to succeed

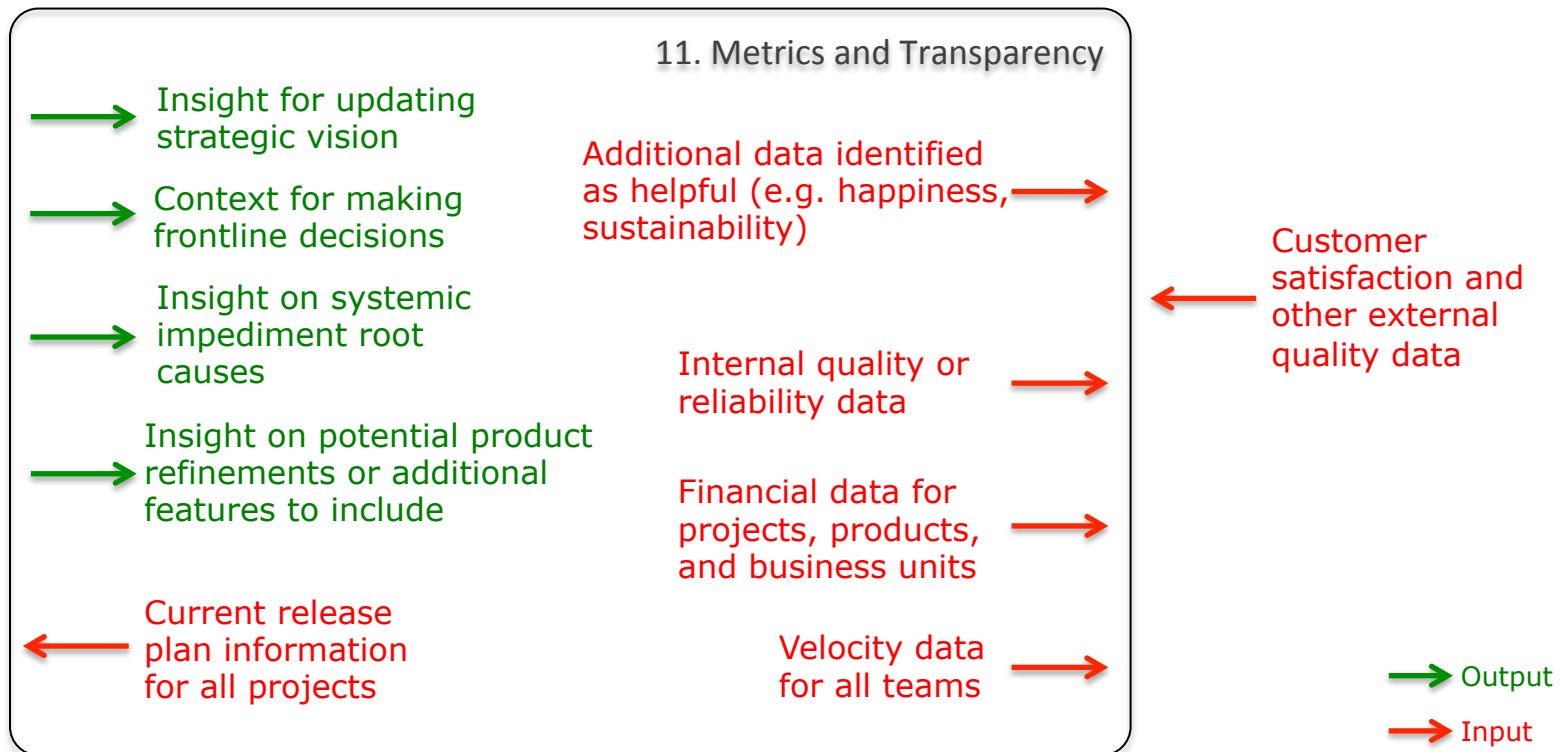
**Transparency Focus**



# 11. Metrics and Transparency

## Module Goals:

- Provide all decision makers including team members with appropriate context to make good decisions
- Shorten feedback cycles as much as possible to avoid over-correction
- Accomplish all of this with minimal additional effort by teams, stakeholders or leadership





# Alternate Approaches to Satisfy the “Metrics and Transparency” Module

A



## Agile PMO

- Agile PMO tracks team velocities, project burndown, actual vs. committed points, impediments, and defect rate centrally
- Metrics available to all leadership, POs and SMs via online dashboard
- All data pulled automatically from tools

**Pro:** Transparent and real-time data available to most decision-makers; consistent metrics across all teams; little effort by teams to produce

**Con:** Significant setup effort to establish system; not as flexible if different teams want to track different metrics

B



## Backlogs & Dashboards

- All teams use same backlog tracking tool and have access to each other’s backlogs, velocity, and burndown
- Component level groups may produce a regular dashboard of additional metrics (bugs, happiness, impediments, etc.) specific to their area

**Pro:** Relatively consistent system for sharing core metrics, with room for variation by team; requires little team overhead

**Con:** Although accessible and consistent, team data requires legwork to access and aggregate by data-user

C



## Ad Hoc

- Enterprise tracks financials, objectives & key outcomes and shares broadly
- Each team chooses its own tools, metrics and methods to display
- All teams have access to every other team’s tools and space, if desired
- Cross-team events

**Pro:** Lightweight; Allows each team to experiment with what works best for them

**Con:** No central and easily accessible source for information; can be very cumbersome to access data only posted in team room



# Automatic Reporting Via Scrum Tools

1. Tap into data the team should already collect to manage their process

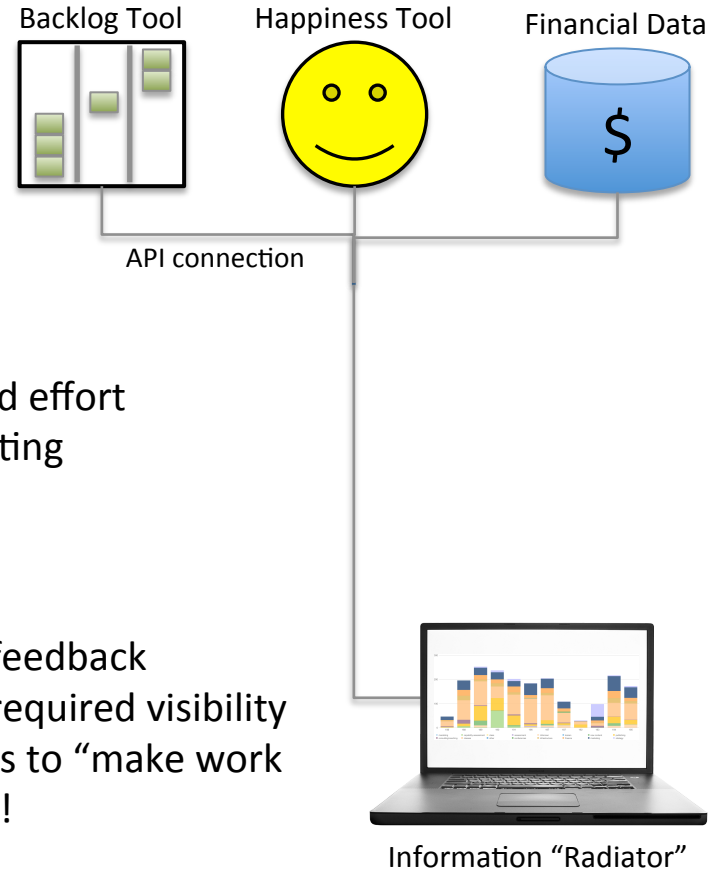
- No additional work

2. Pull and aggregate it automatically

- API interfaces
- “The Cloud”

3. Make it available to everyone to drive radical transparency

- Team gets clear feedback
- Leadership gets required visibility
- Creative solutions to “make work visible” welcome!

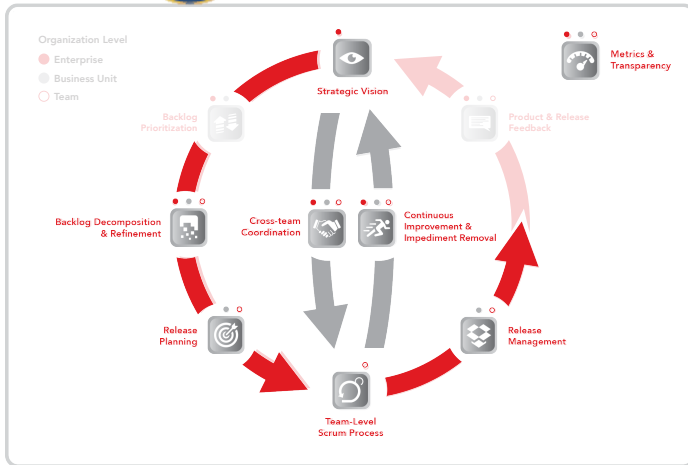


# Modularity Supports Different Implementation Paths

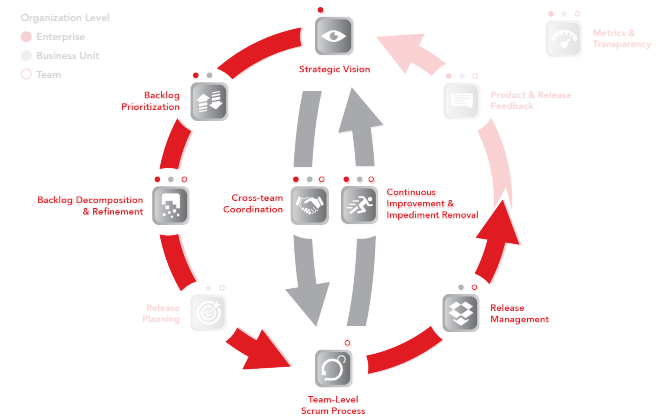
Start  Current



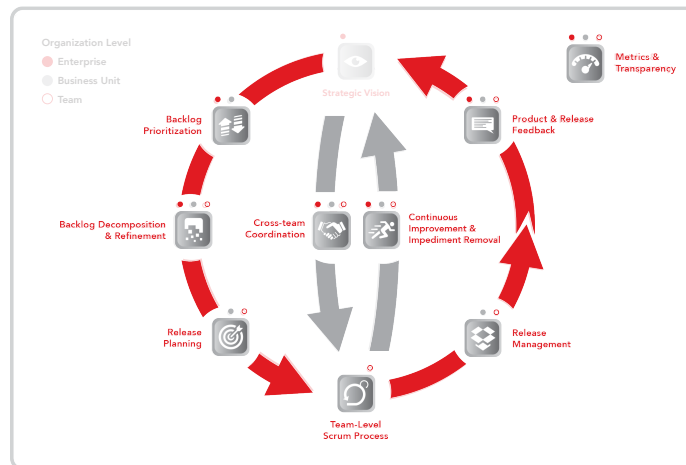
**Name Classified**



**Spotify**



**Autodesk**



# Conclusion

- Scrum has matured to the point that many companies have successfully implemented it at scale
- But it is not a “one size fits all” success story, context is vital
- We need, and have tried to present a language for discussing scaling issues in context
- Now we need to start building a library of successful alternative practices for each module under different organizational contexts

[www.scruminc.com/agile2014](http://www.scruminc.com/agile2014)



# Questions?



# Stay Connected

## E-mail

- [Alex@scruminc.com](mailto:Alex@scruminc.com)

## Twitter, Facebook, and G+

- @Scruminc, #Scrum, #Agile

## Our Website

- [www.scruminc.com](http://www.scruminc.com)
- check in for announcements, new content and services, book releases, and more!

## ScrumLab

- [Scrumlab.scruminc.com](http://Scrumlab.scruminc.com)
- Articles, videos, papers on all things scrum

## Online Courses

- Advance your learning with our interactive online courses. Visit [scrumlab](http://scrumlab) to view upcoming topics.

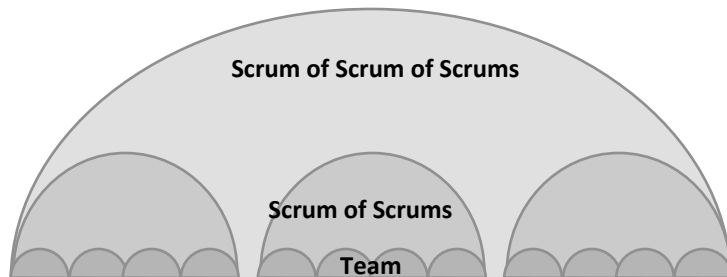
## Scrum at Scale Pattern Library

- [www.scruminc.com/agile2014](http://www.scruminc.com/agile2014), #ScrumAtScale

# Scrum Master and Product Owner Functions Scale Differently

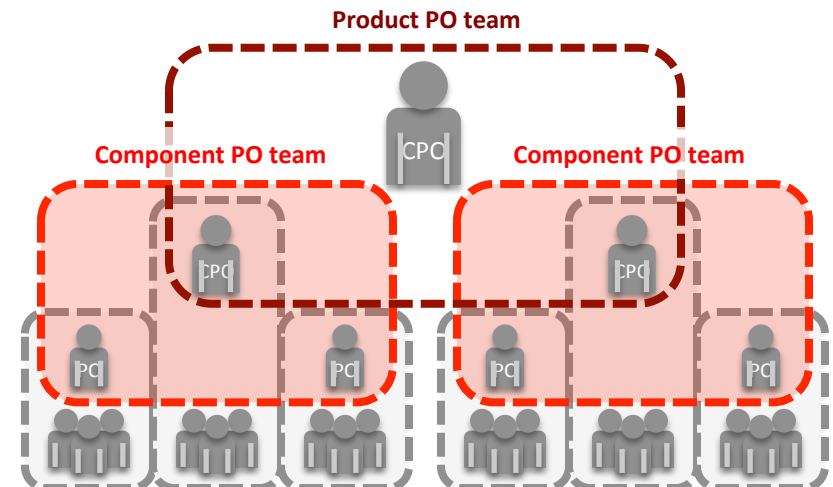
## Scrum Master

- Share best practices
- Collectively solve problems & remove impediments



## Product Owner

- Maintain clear and consistent product vision
- Optimize business value
- Respond decisively to changing market



# Example: Scaled Agile Framework

## Scaled Agile Framework® Big Picture

