

Feature vs. Component Teams

January 15, 2013

Raleigh, North Carolina

by Kenny Rubin

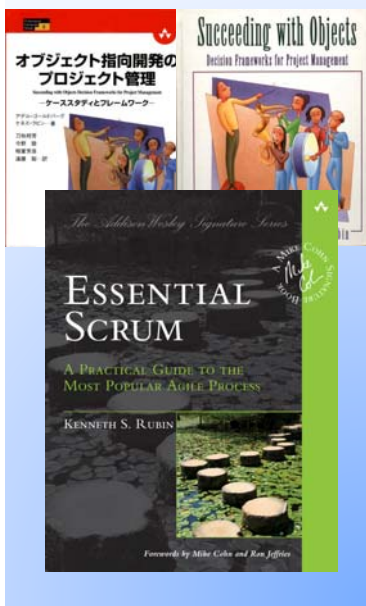
www.innovation.com

Copyright © 2007-2013, Innolution, LLC. All Rights Reserved.

1

Background of Kenny Rubin

Author



Trainer/Coach

Trained more than
18,000 people in
Agile/Scrum, SW
dev and PM

Provide Agile/
Scrum coaching to
developers and
executives



Experience

Former Managing
Director



My first Scrum project was
in 2000 for bioinformatics

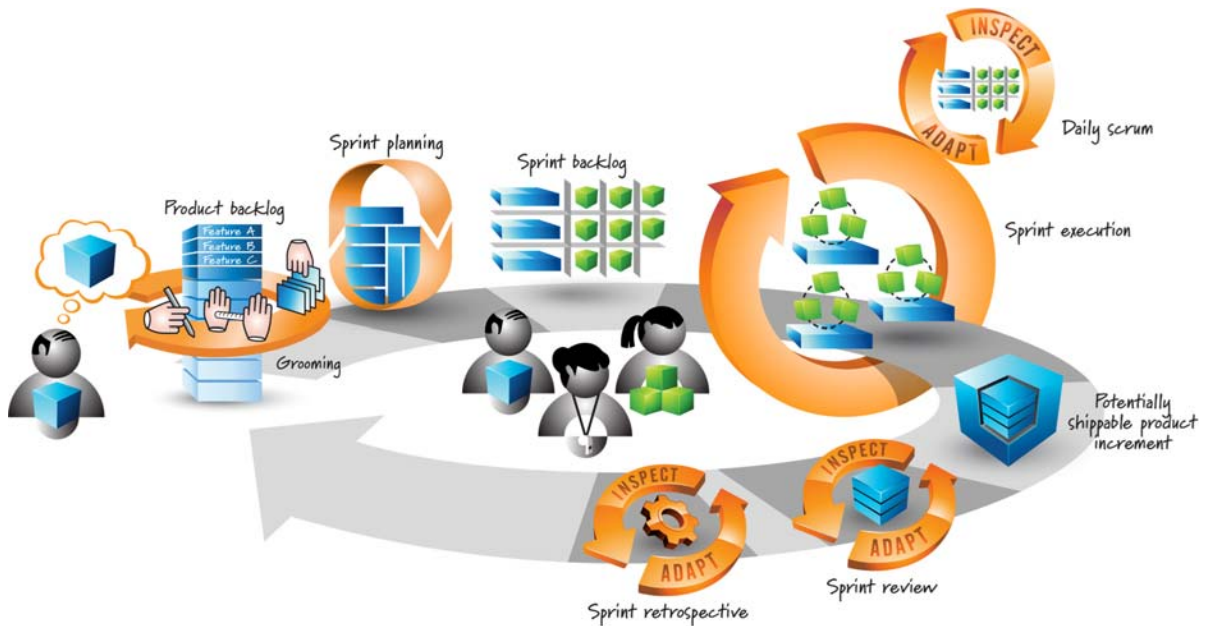
GENOMICA



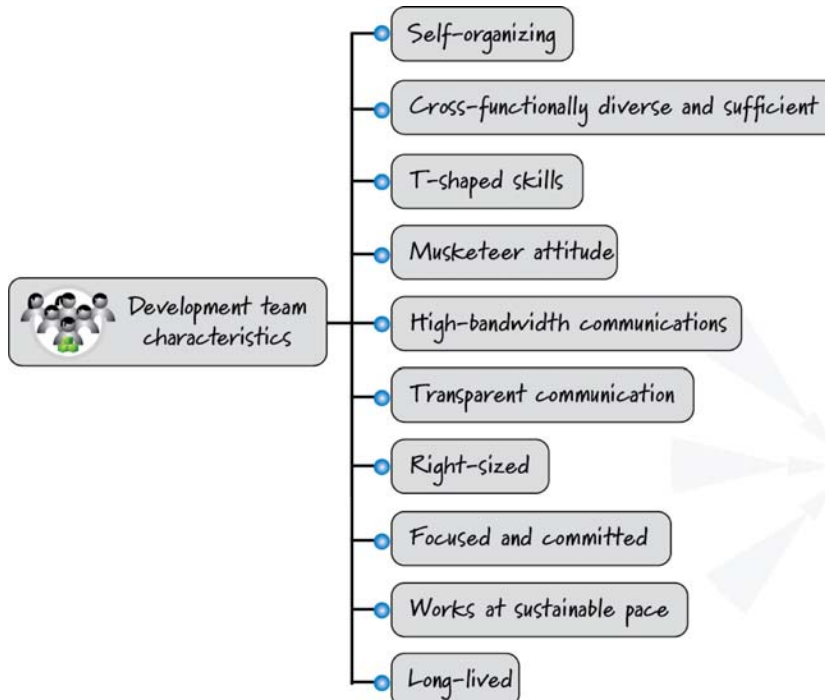
Executive



Simple Agile Has One Product Backlog and One Team



Characteristics of a Single Development Team

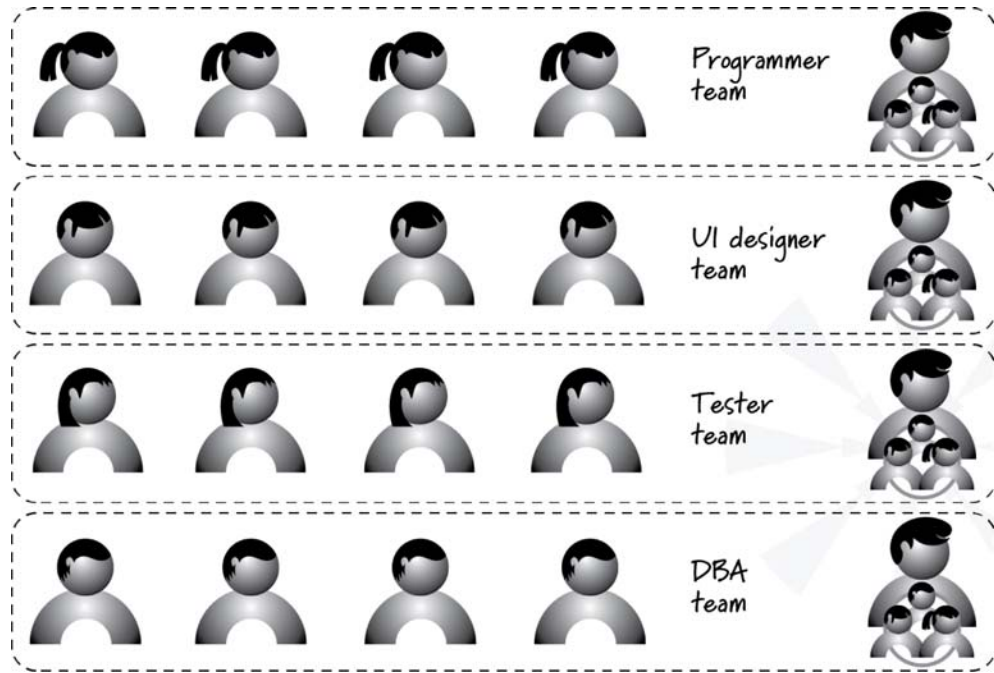


✦ Scaling Question #1

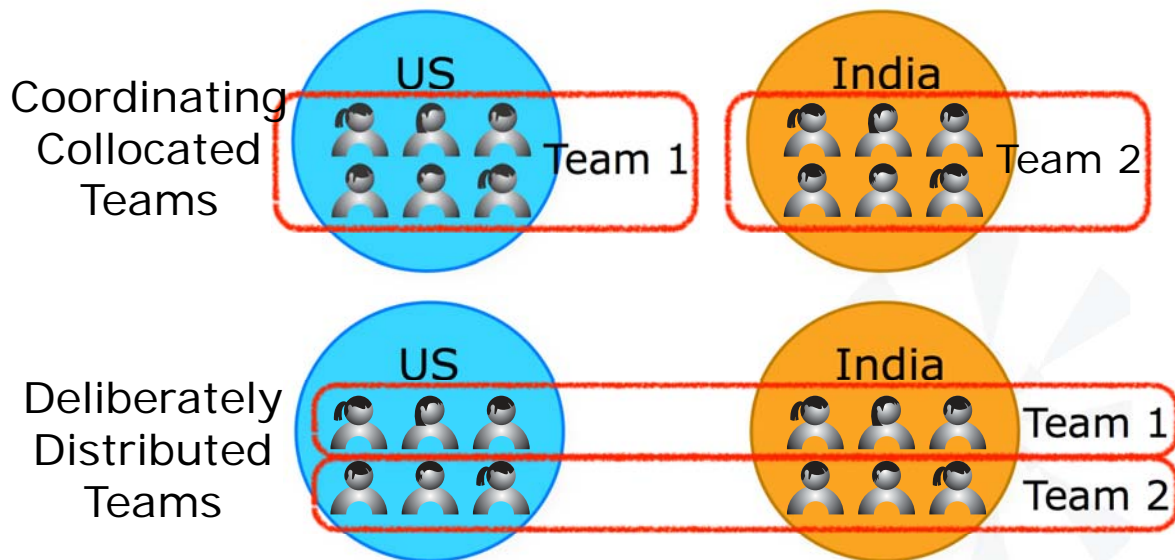
As the scope of work gets larger and one team is no longer sufficient, what is your scaling strategy?



Discipline Teams



Location Teams



Architectural Layer Teams



GUI



Middle Tier



DB



Component Teams

Component team 1



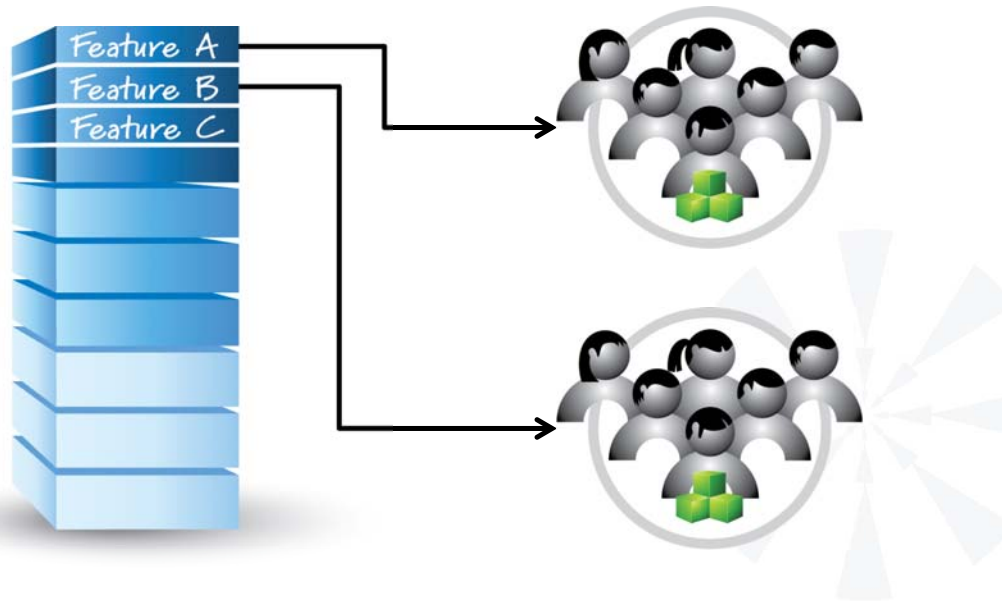
Component team 2



Component team 3



✦ Feature Teams

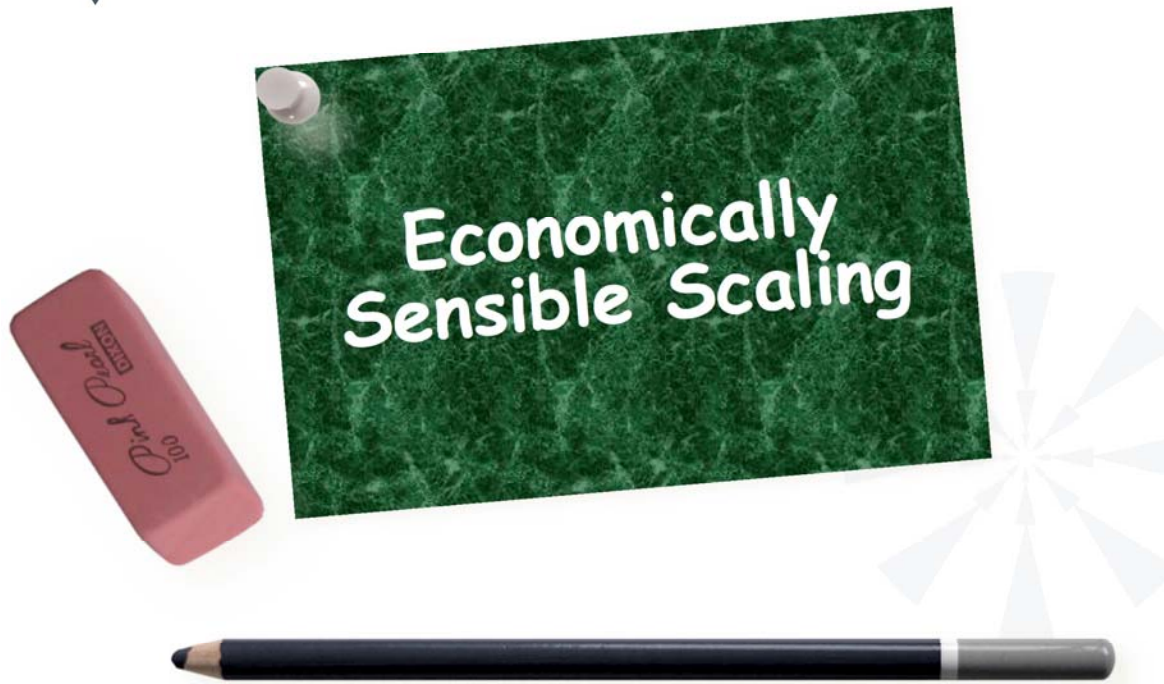


✦ Scaling Question #2

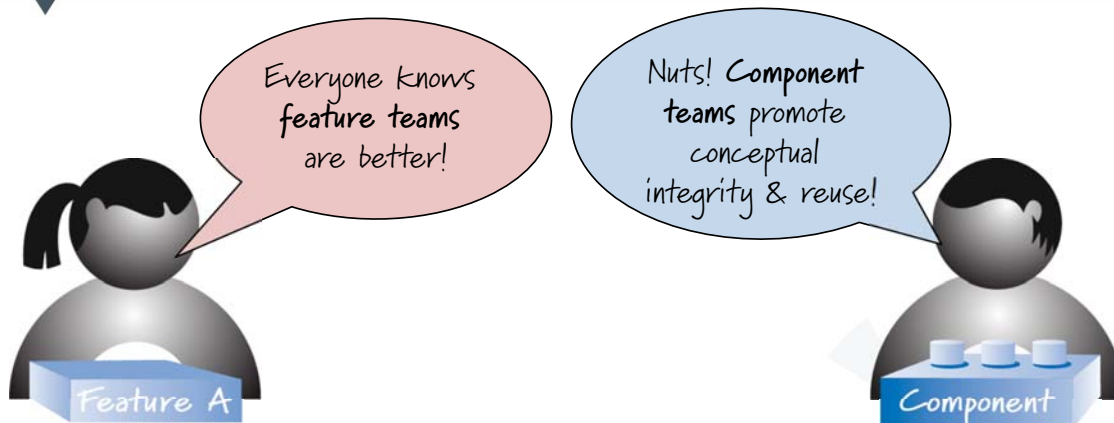
So, which approach do you prefer?

What criteria are you using to decide?





✱ Don't Scale Based on Dogma!

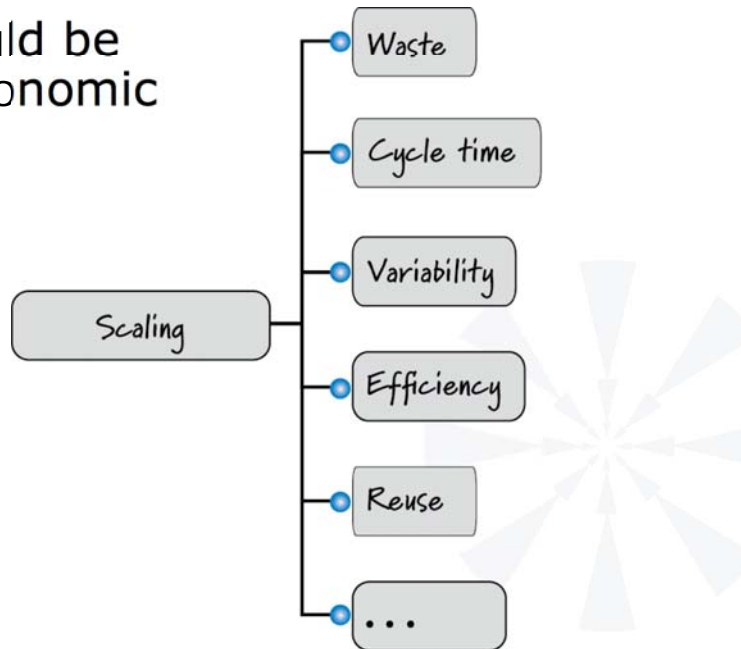


Do you honestly think there is a single answer to scaling that universally applies to all situations (sizes and types of organizations)?

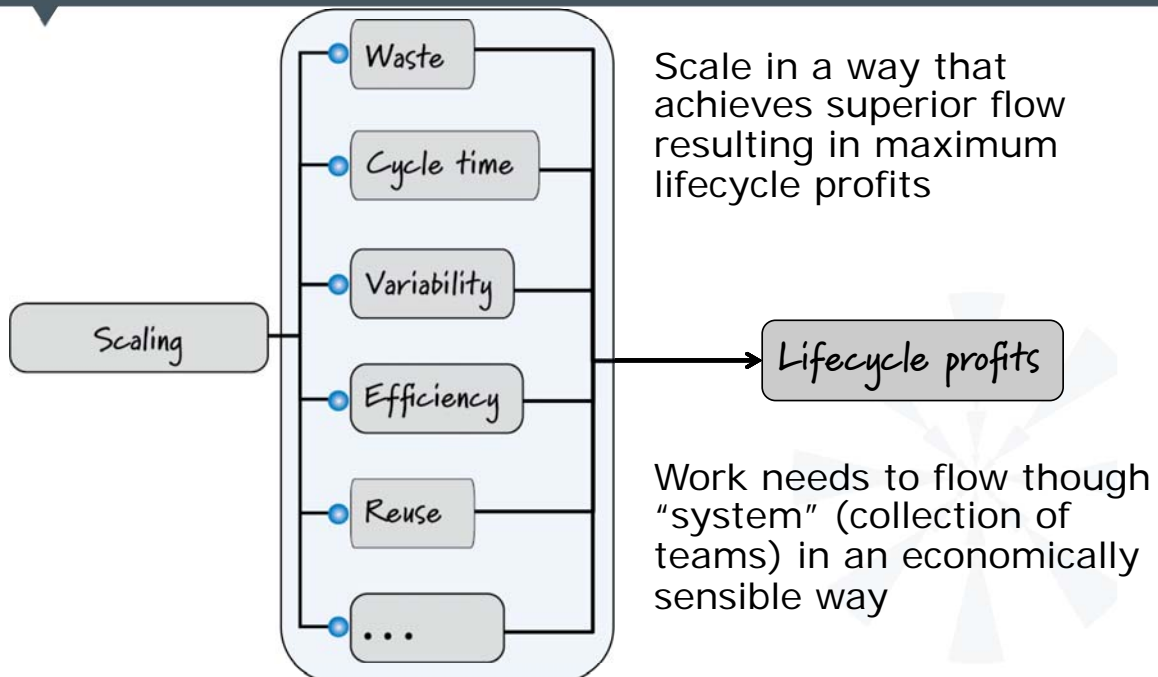


Scale Based on Economic Tradeoffs

- Scaling should be based on economic factors



Scale to Maximize Lifecycle Profits

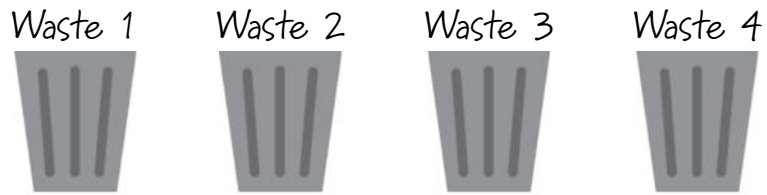


Based on Reinertsen 2009.

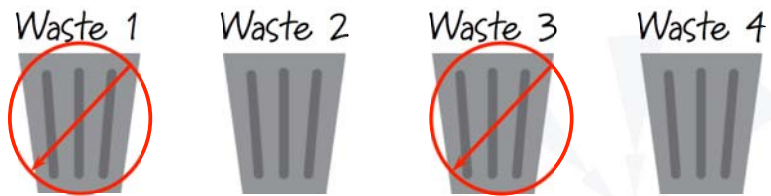


Waste

Multiple forms of waste



Can't eliminate them all



Determine which cause most economic damage

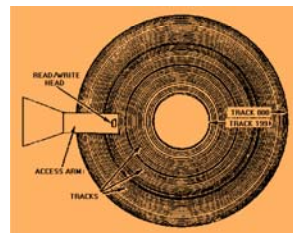


Recognize Inventory (WIP) Waste

Manufacturing inventory is both physically and financially visible

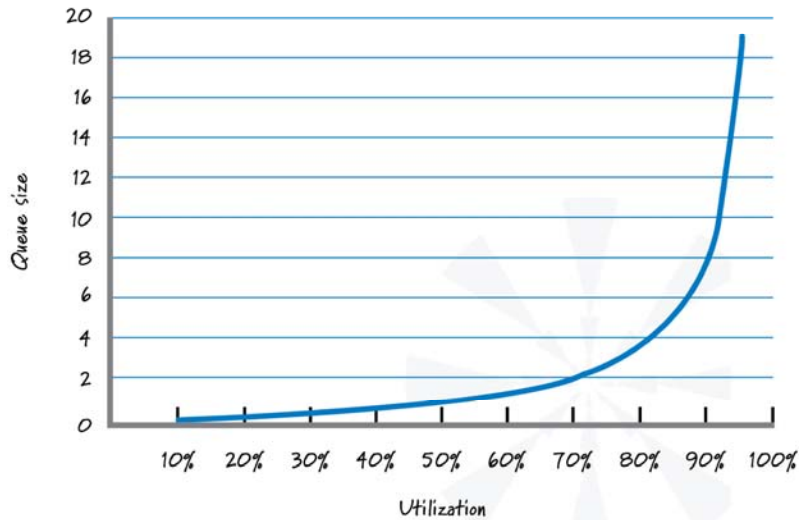


Product-development inventory are knowledge assets that aren't visible in the same way as physical parts

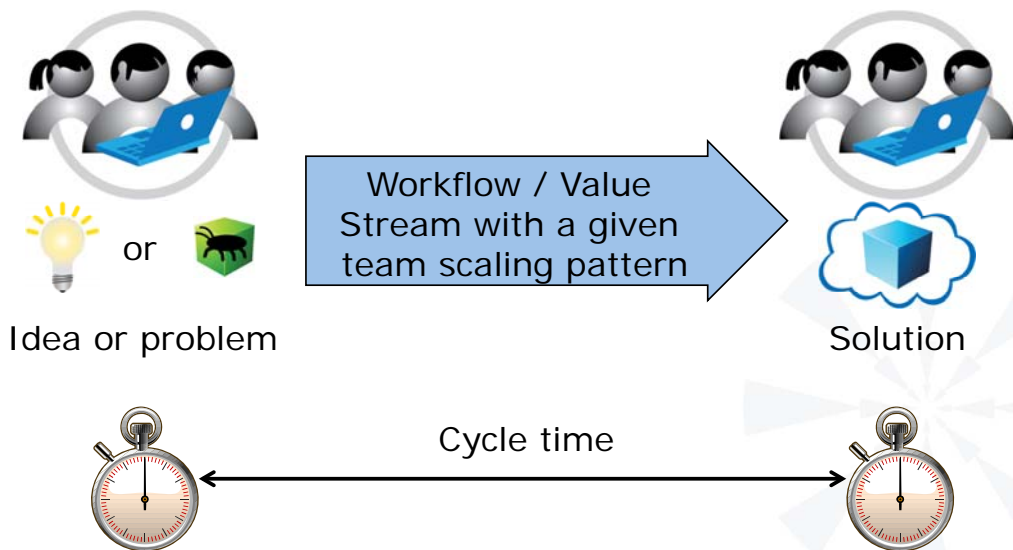


Focus on Idle Work Not Idle Workers

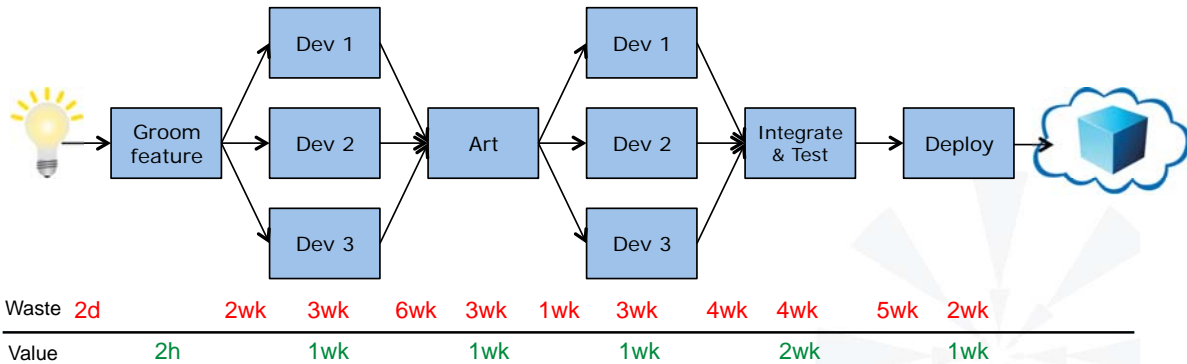
Watch the Baton Not the Runners



Cycle Time (Lead Time)



Example Workflow / Value Stream



$$\frac{6 \text{ wk value-adding time}}{39.4 \text{ wk cycle time}} = 15\%$$
 Process cycle efficiency

Improve team efficiency 10% yields 1.5% improvement
 Eliminate 10% waste yields 8.5% improvement



Cost of Delay

If you have to wait 6 weeks for the Art team to draw your art, and that delay could be eliminated by having artists on your team, what would be the cost of the Art-team delay (in lifecycle profits)?

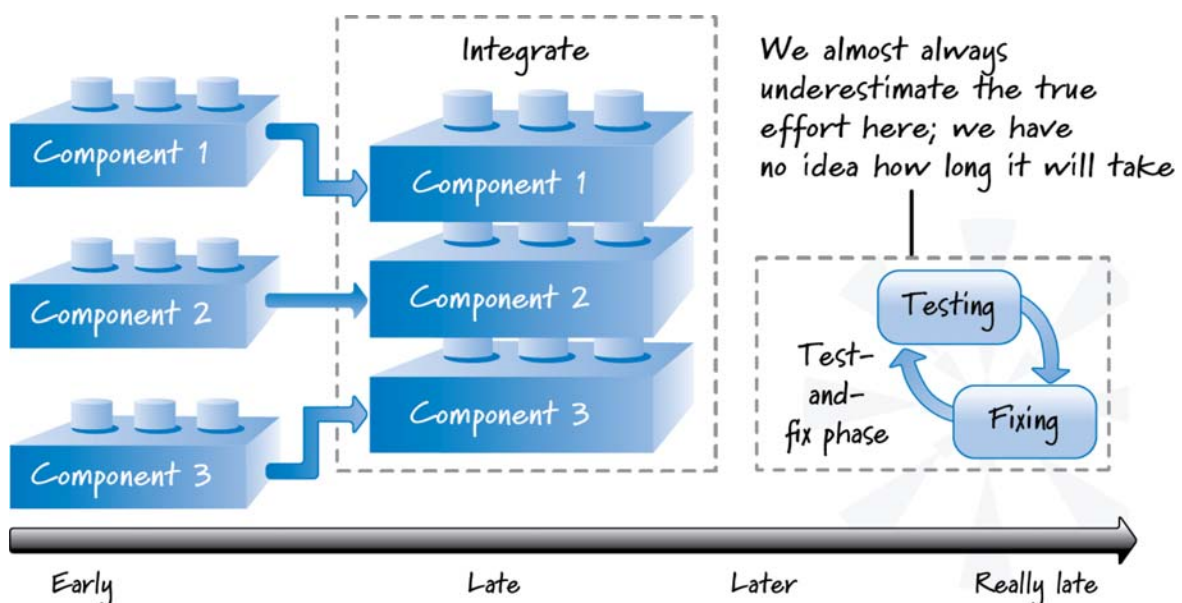


Team Structure Can Effect Predictability

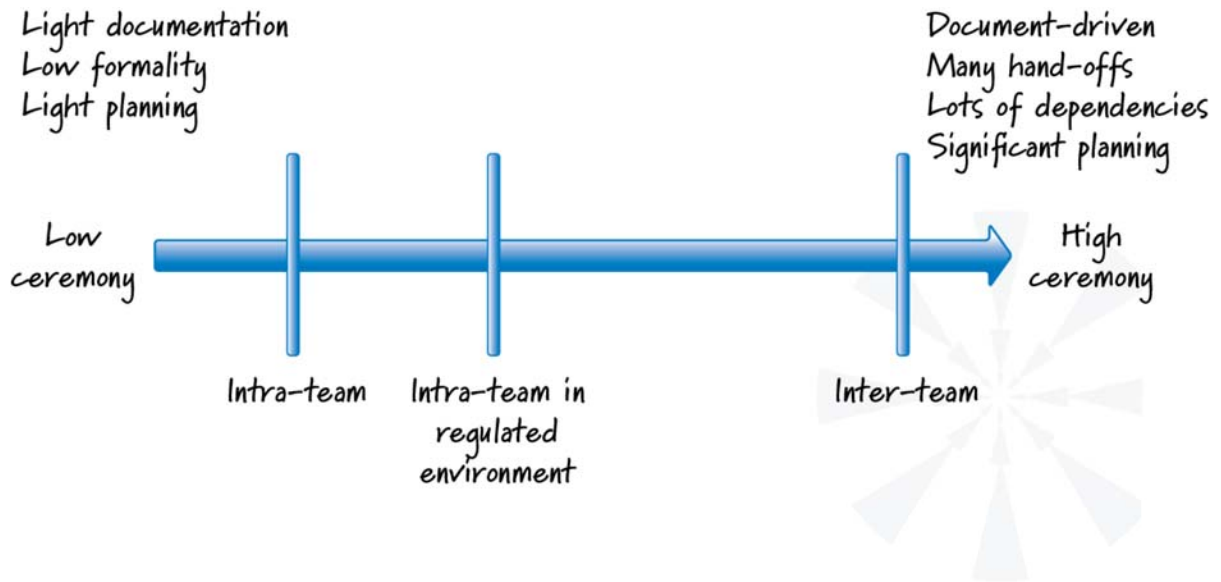
What if we organized our teams in a way that made the product-development process inherently more predictable?



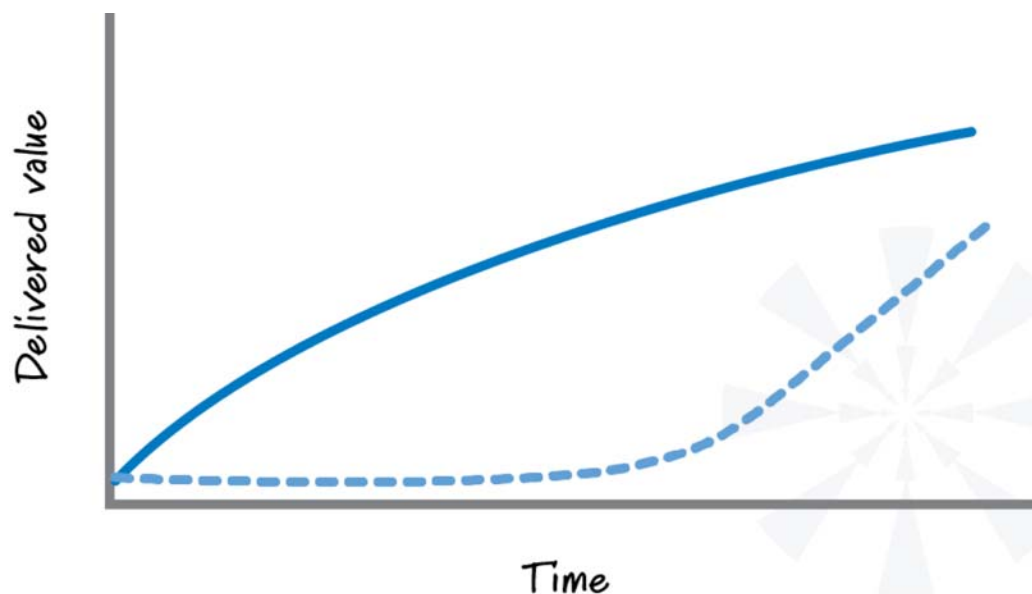
Validate Important Assumption with Fast Feedback



✶ Eliminate Unnecessary Ceremony

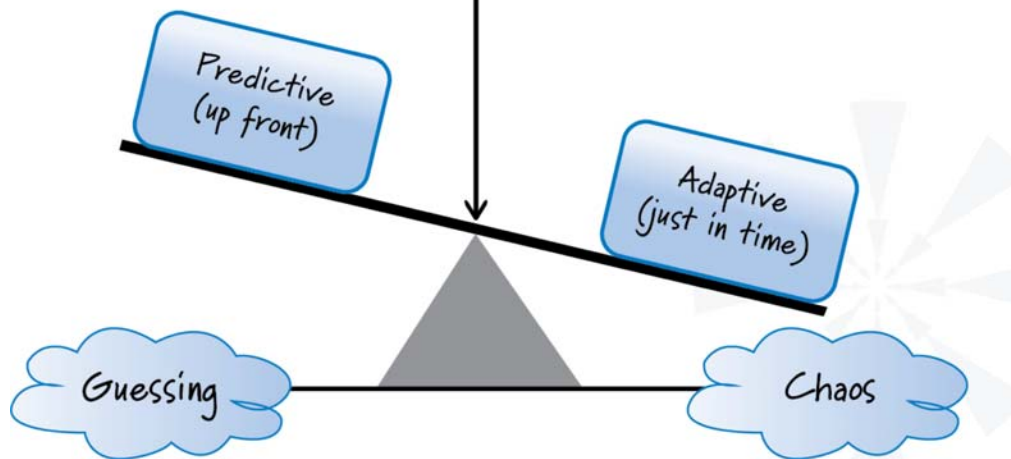


✶ Value-centric Deliverables

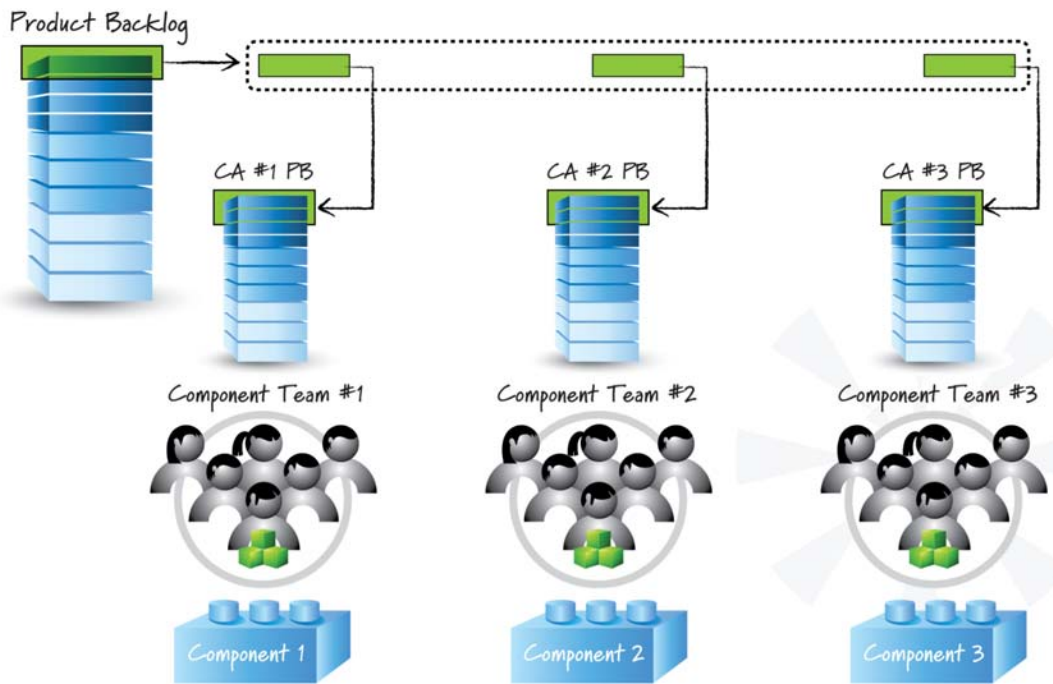


Team Pattern Should Balance Predictive and Adaptive work

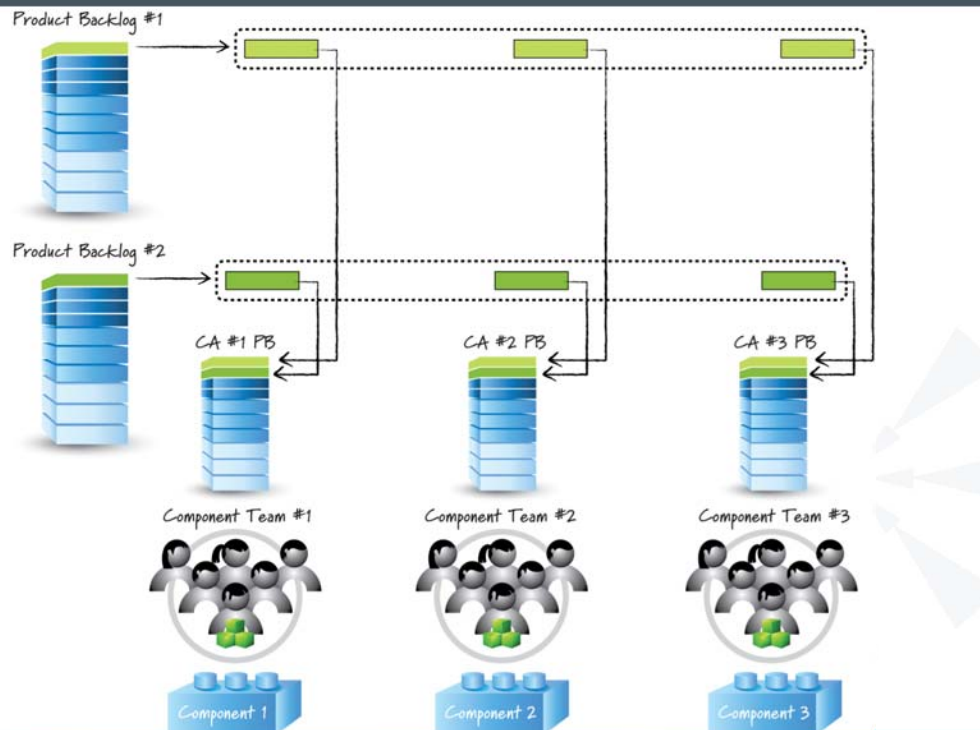
How teams are organized can influence balance point



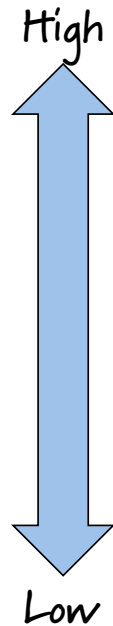
Component Teams (Single Source)



Component Teams (Multiple Sources)

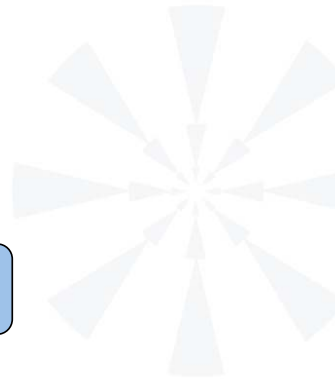


* Issue – Blocked Work



Cost of delay

Process efficiency



* Issue – Prioritization

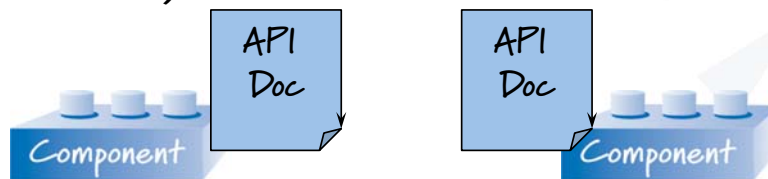


- * Localized prioritization decisions based on things like:
 - * Technical priorities of component
 - * Whatever is fun or easy
- * Feature prioritization can be driven by component team availability
- * NPF (Nosiest Person First) can dominate work order

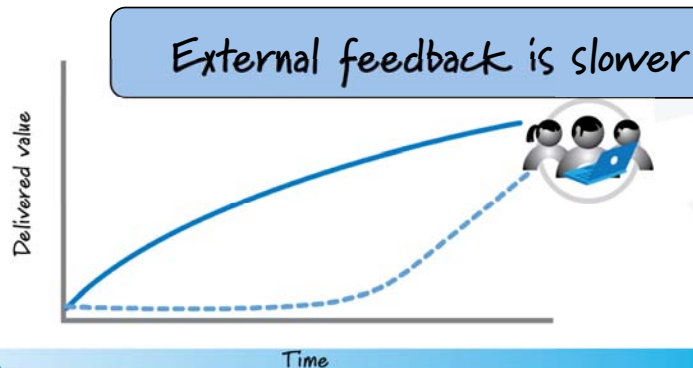
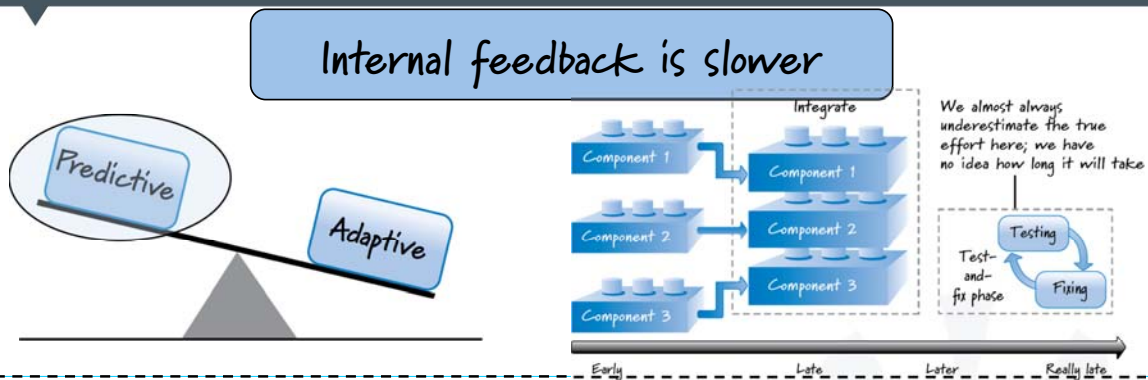


Issue – Coordination Cost

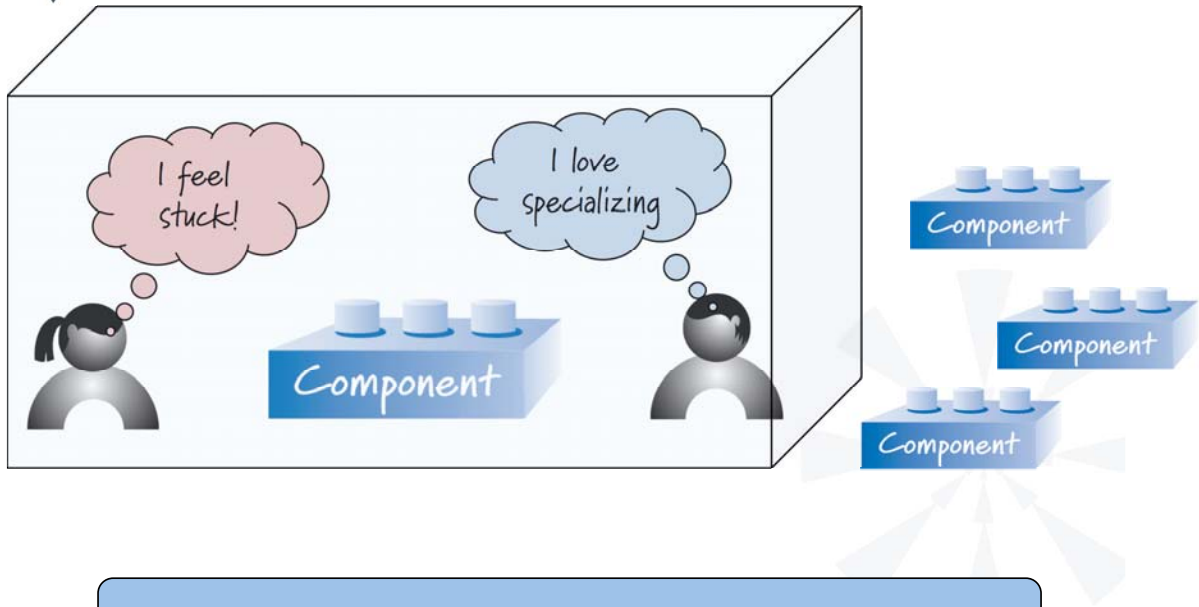
- ✱ Requires significant and on-going planning, handoffs, and dependency management
- ✱ At scale dependency management becomes economically intractable
- ✱ Favors low-bandwidth means of communication (e.g., interaction by contracts)



Issue – Slower Feedback



Issue – Limits Learning



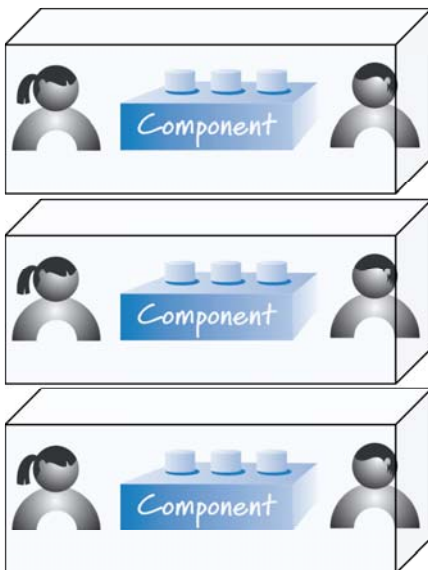
Risky: specialty knowledge in only a few heads



Issue – Harder to See the Whole

Best components ever!

But still a poor product



Alignment trumps local excellence



Desirable Property – Conceptual Integrity

- * Knowledgeable and trusted people work in the code
 - * Ensure conceptual integrity
 - * Low technical debt
- * Conceptual integrity provides:
 - * Congruity; consistency; logical interconnectivity, overall cohesive and understandable
- * Want conceptual integrity both at the component and the full system/product level
 - * NOTE: conceptual integrity at the component level doesn't guarantee conceptual integrity at the product level



Desirable Property – Asset Reuse

- * Build it once, use it often
- * Avoid building the same capability in multiple, potentially inconsistent ways
- * Would otherwise appear in many places in the codebase, complicating maintenance and testability
- * Economically a sensible concept, but need to consider the full cost of reuse

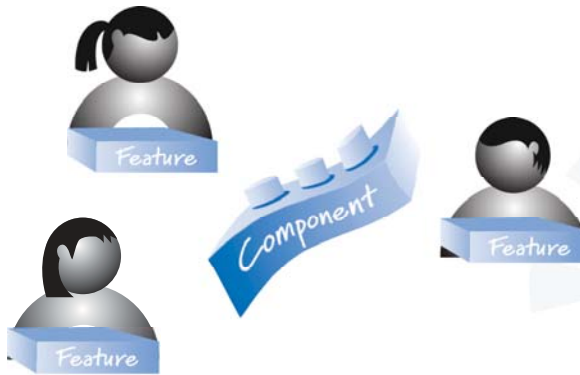




Issue – Lack of Conceptual Integrity

Incompatible changes

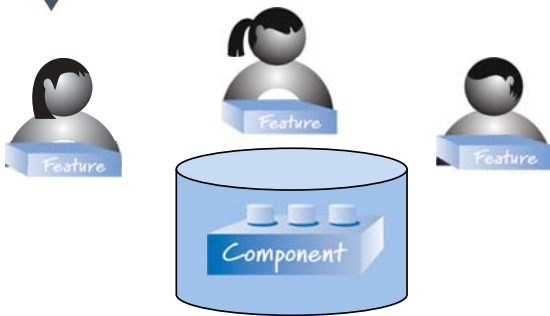
Shared design



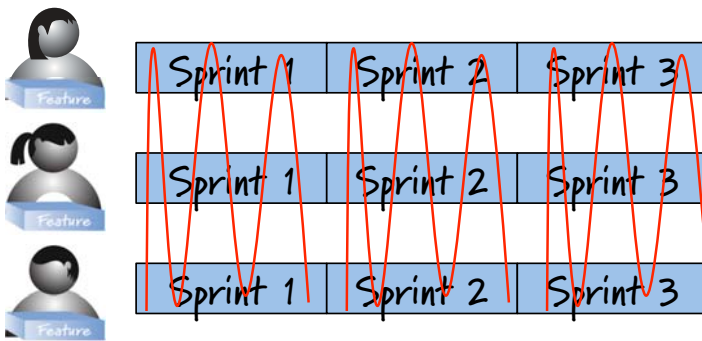
Who owns it?



Issue – Technical Practices



Manage concurrent access



Continuously integrate work



Issue – Lack of Knowledge



Need deep domain skills

Need deep technical skills

Need to understand large system



Issue – Non-functional Requirements

Who ensures the non-functional requirements?

As a customer, I want to be one of 10,000 customers who can use the system during peak usage periods.

As a user, I want the site to be available 99.999% of the time I try to access it.

As the CTO, I want the new system to conform to our established security policies.

As a user, I want an interface in English, a Romance language and a complex language.



Issue – Team Longevity

Product 1



PB 1



Feature Team A



Product 2



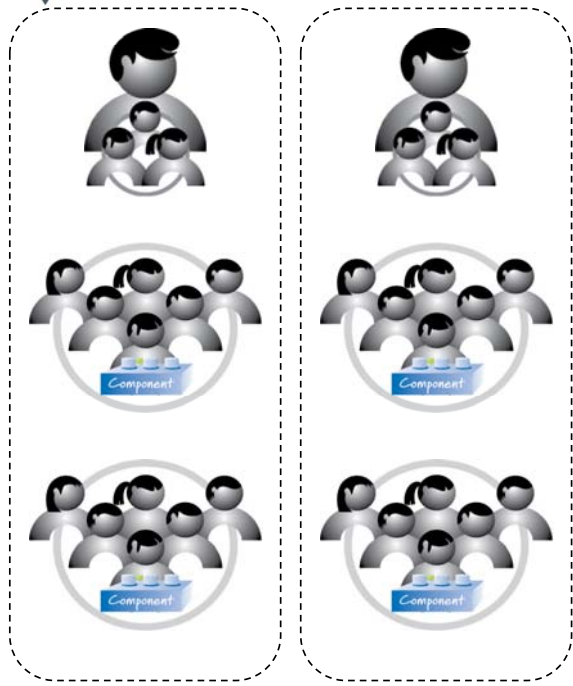
PB 2



Feature Team A



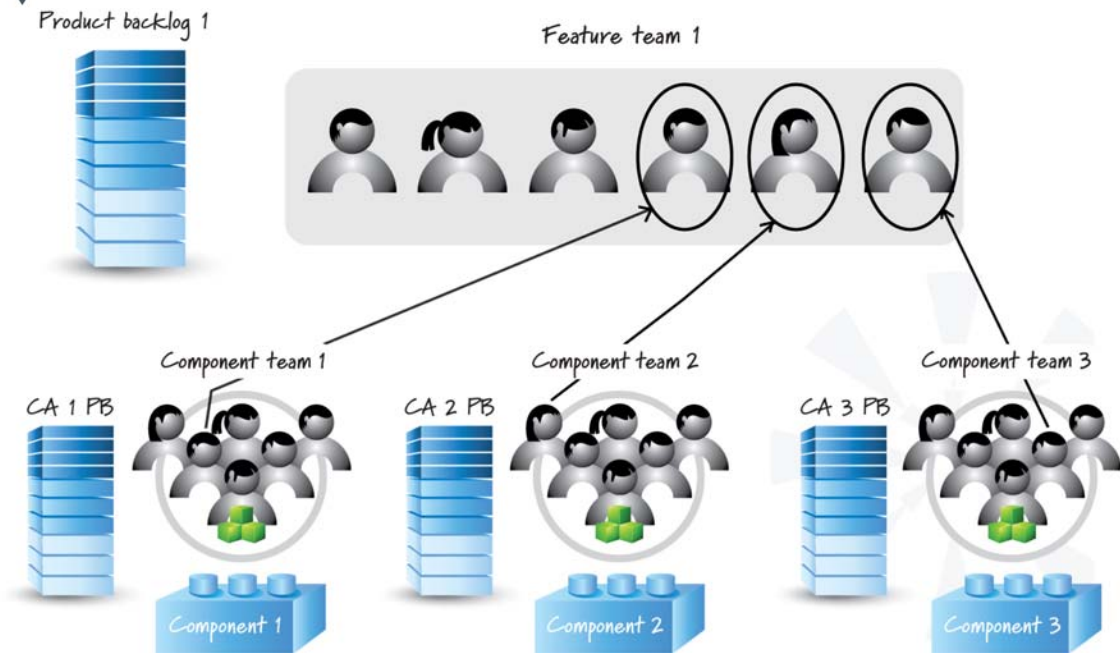
✦ Issue – Organizational Resistance



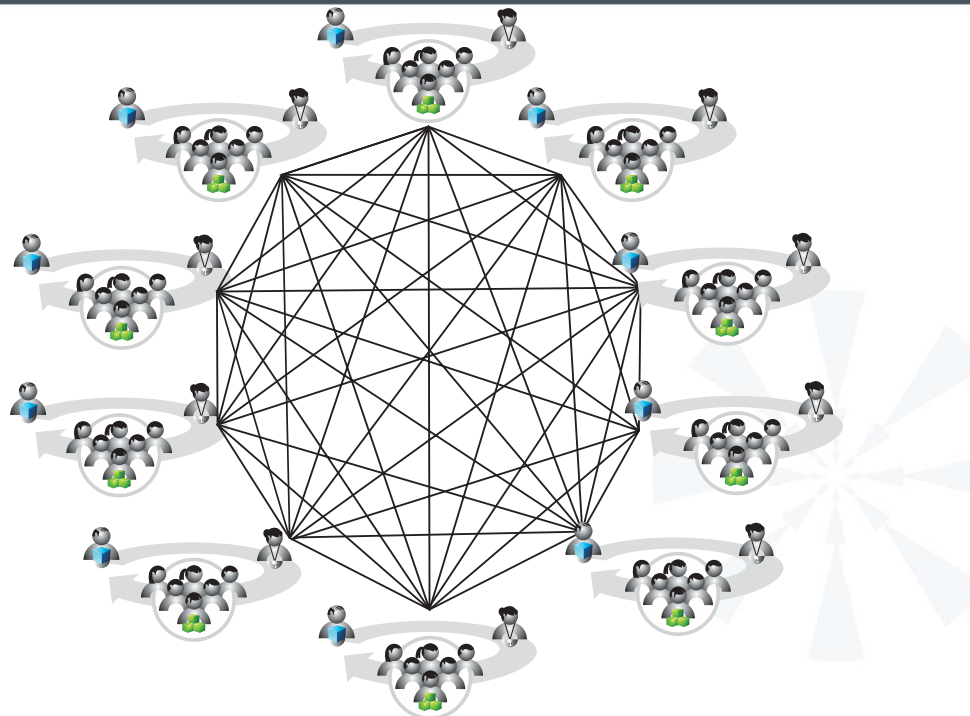
- ✦ Interferes with fiefdoms
- ✦ Too hard to reorganize into feature teams
- ✦ A general belief that feature teams will lead to significant technical debt



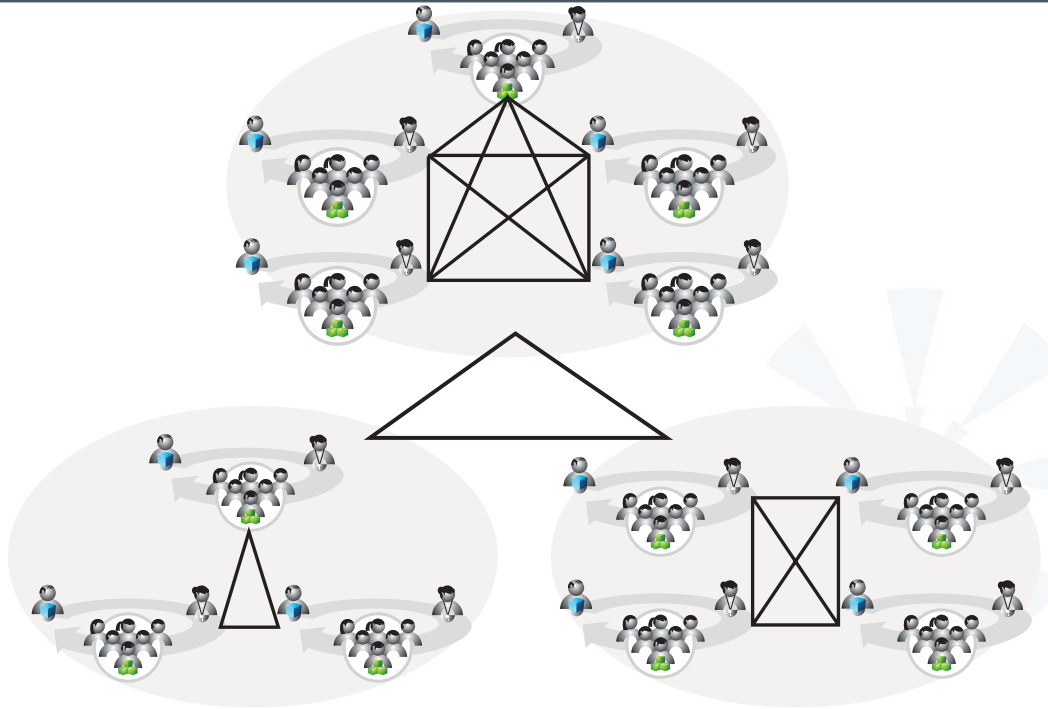
Combined Feature & Component Teams



Teams with Fully Connected Communication Channels



☀ Teams for Collaboration Clusters

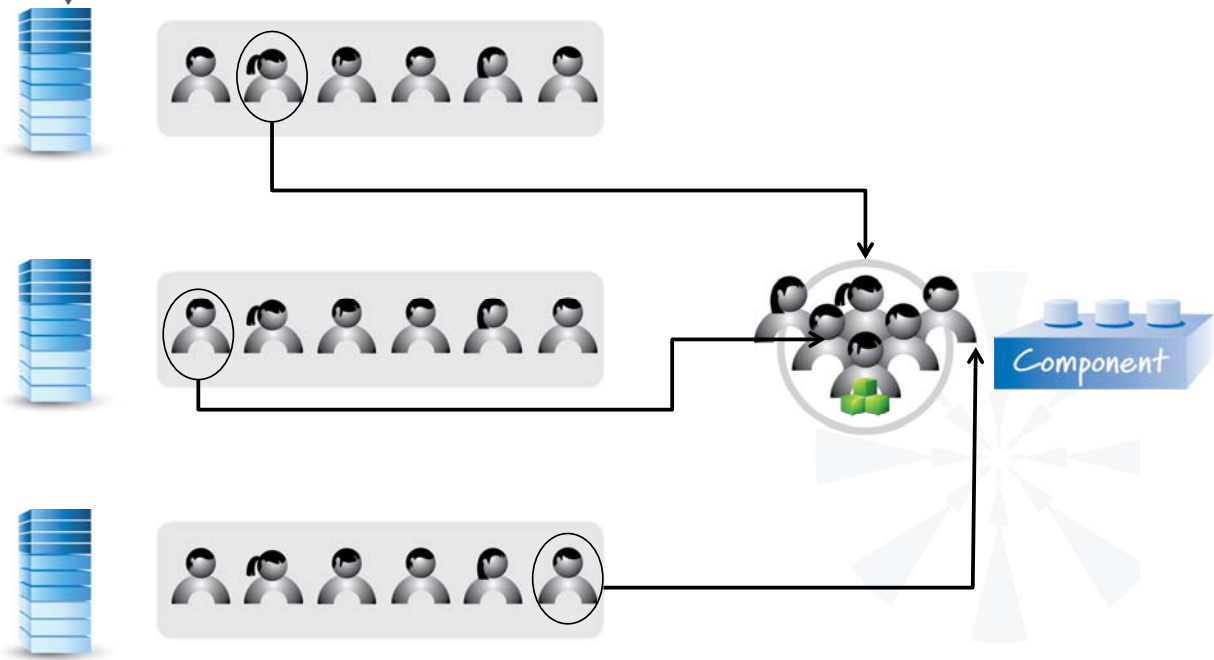


☀ Component Stewards/Guardians

- ☀ One or more people that teach others about the component
- ☀ Ensures changes maintain or improve conceptual integrity
- ☀ Not the owner of the component; feature teams make the changes
- ☀ Can also take a leadership role in promoting reuse

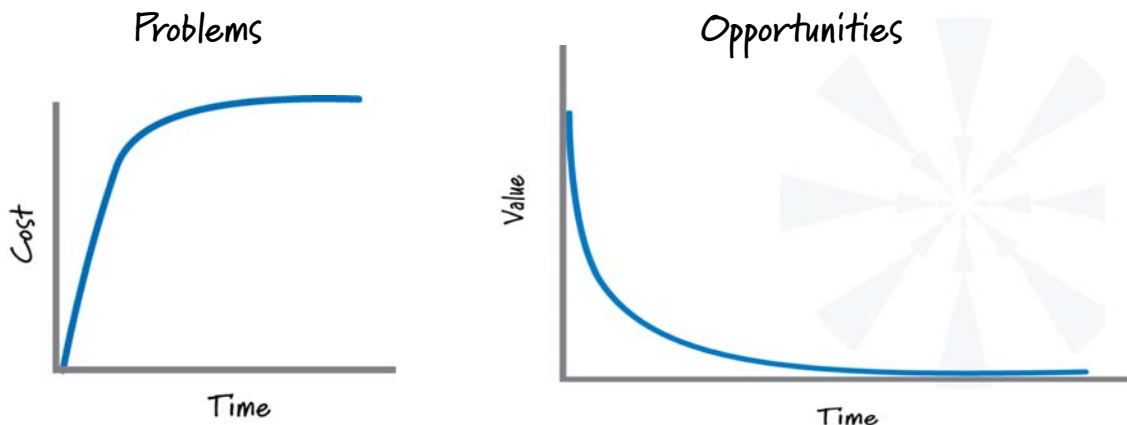


✦ Create a Community of Practice from Feature Team Members



✦ Deal with Problems/Opportunities that Age Poorly

- ✦ Structure teams so we can attack problems that escalate fast or opportunities that disappear quickly



☀ Scaled Agile Framework Recommendation

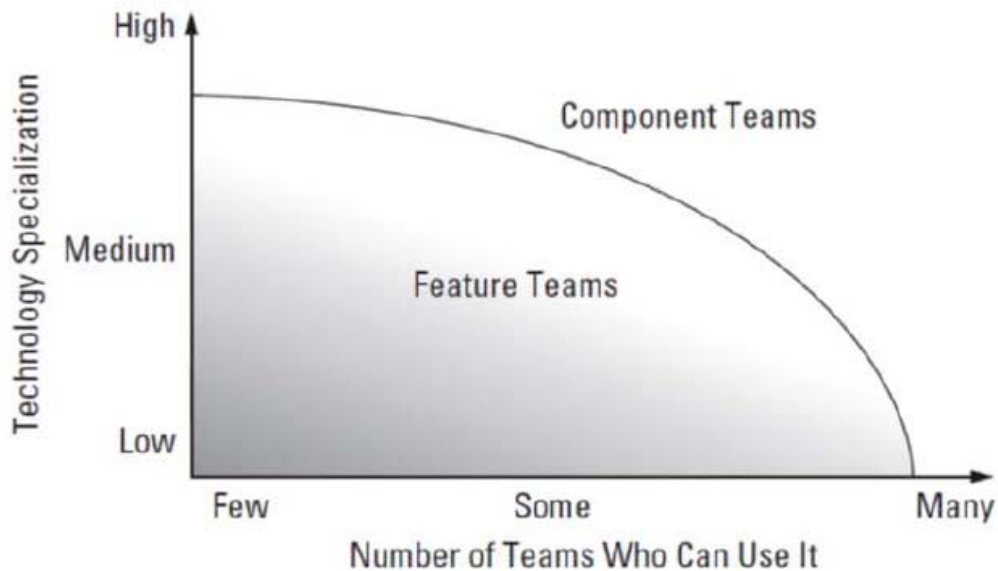


Figure 3. The feature and component teams power curve

Copyright © 2007-2013, Innolution, LLC. All Rights Reserved.

53

☀ Top Down System Level Approach

What are your products?

What are your product backlogs?

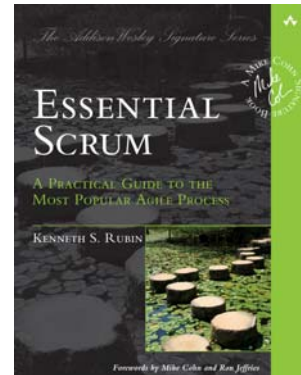
What teams do you need to deliver on your goals?

Copyright © 2007-2013, Innolution, LLC. All Rights Reserved.

54

Visual AGILExicon™

- ✦ Slides in this presentation contain items from the Visual AGILExicon™, which is a trademark of Innolution, LLC and Kenneth S. Rubin.
- ✦ The Visual AGILExicon is used and described in the book: **Essential Scrum: A Practical Guide to the Most Popular Agile Process.**
- ✦ You can learn more about the Visual AGILExicon and permitted uses at: <http://innolution.com/resources/val-home-page>



www.essentialscrum.com

Home | Contact Us

TRAINING ESSENTIAL SCRUM RESOURCES BLOG ABOUT US

ESSENTIAL SCRUM

View the Book Trailer

Read the Table of Contents

Read Reviews

Order on Amazon

Essential Scrum

Watch the Book Trailer

Introducing Essential Scrum, A Practical Guide to the Most Popular Agile Process by Kenneth S. Rubin.

▶ Watch the trailer now

Buy now from amazon.com

READ THE TABLE OF CONTENTS:

Browse through the Table of Contents to find out more about the how the book can help you:

Choose a chapter...

SEE WHAT PEOPLE ARE SAYING:

“Corporate IT leadership, which has been slow to embrace agile methods, would benefit immensely from giving a copy of this book to all of their project and delivery managers. Kenny Rubin has laid out in this book all the pragmatic business case and process materials needed for any corporate IT shop to successfully implement Scrum.”

— John F. Bauer III, veteran of technical solution delivery in large corporate IT shops

more reviews >



Contact Info for Kenny Rubin



Email:	krubin@innolution.com
Website:	www.innolution.com
Phone:	(303) 827-3333
LinkedIn:	www.linkedin.com/in/kennethrubin
Twitter:	www.twitter.com/krubinagile
Essential Scrum: A Practical Guide to the Most Popular Agile Process	www.essentialscrum.com
Comparative Agility Website	www.comparativeagility.com

