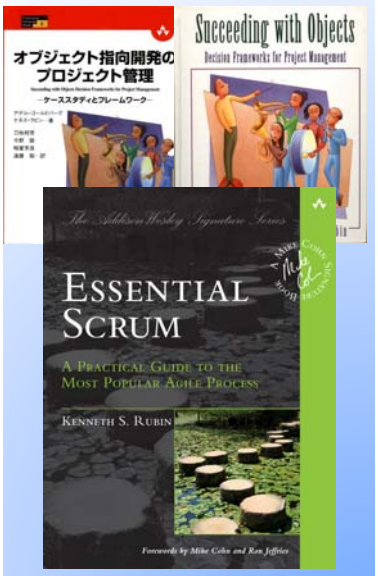
Scaling with Feature vs. Component Teams
Agile 2013
Nashville, TN
August 8, 2013
by Kenny Rubin

www.innolution.com

Background of Kenny Rubin

Author



Trainer/Coach

Trained more than 20,000 people in Agile/Scrum, SW dev and PM

Provide Agile/Scrum coaching to developers and executives

CST ScrumAlliance CERTIFIED ScrumTrainer

Experience

Former Managing Director

ScrumAlliance transforming the world of work

My first Scrum project was in 2000 for bioinformatics

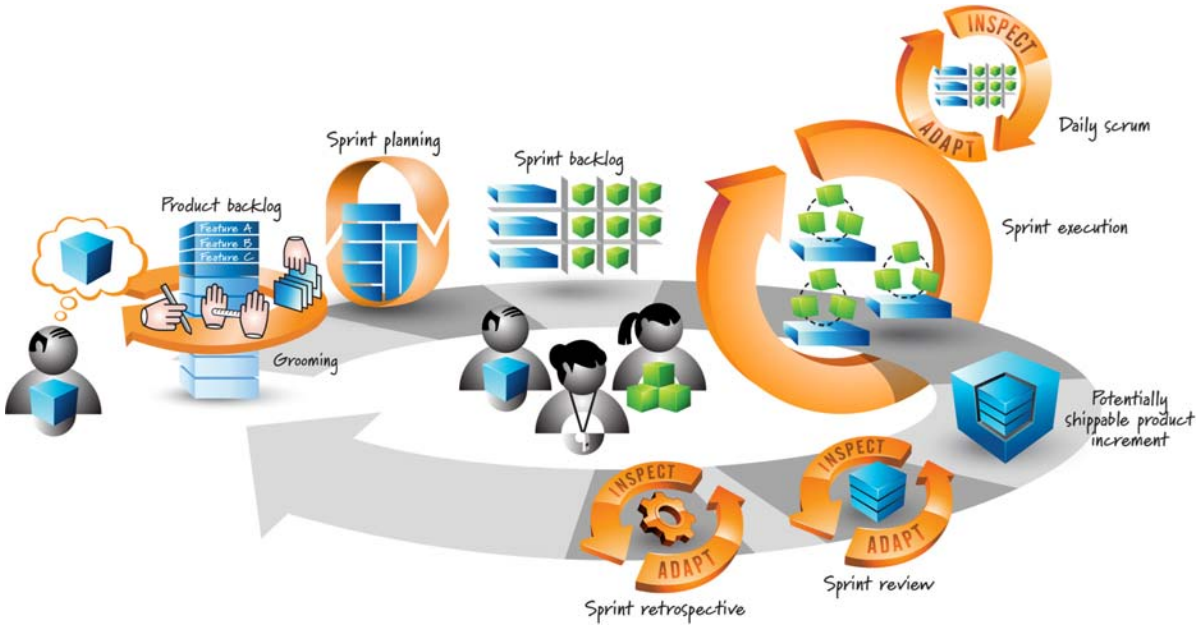
GENOMICA

IBM **Executive**

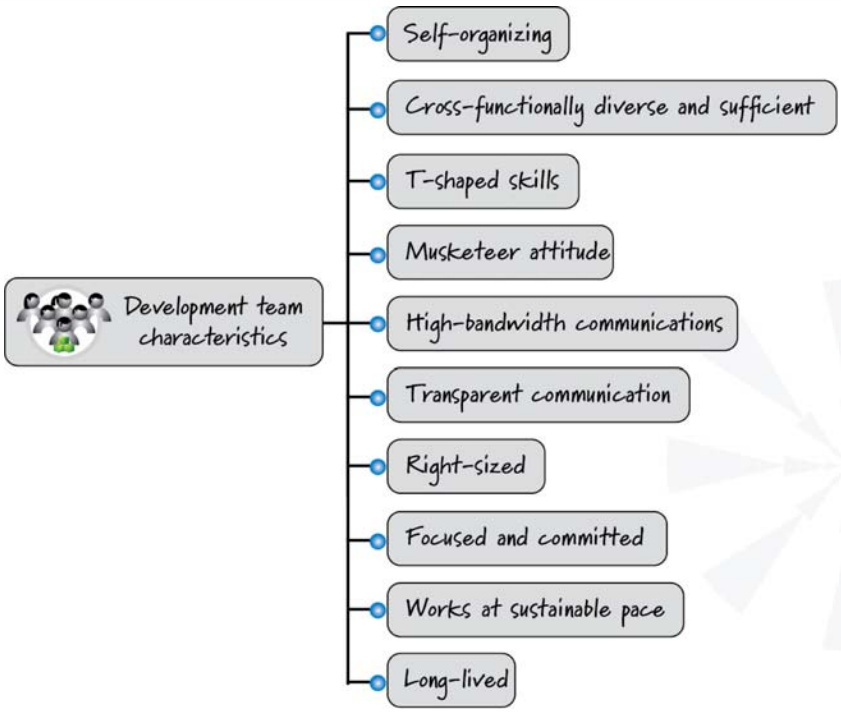
SMALLTALK

ParcPlace

Simple Agile Has One Product Backlog and One Team



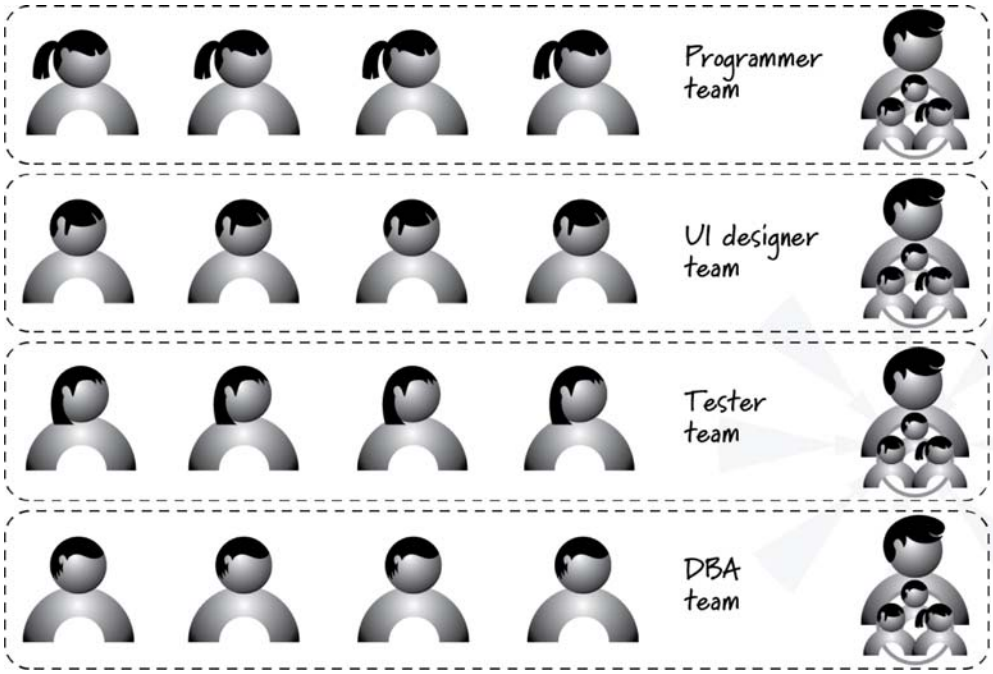
Characteristics of a Single Development Team



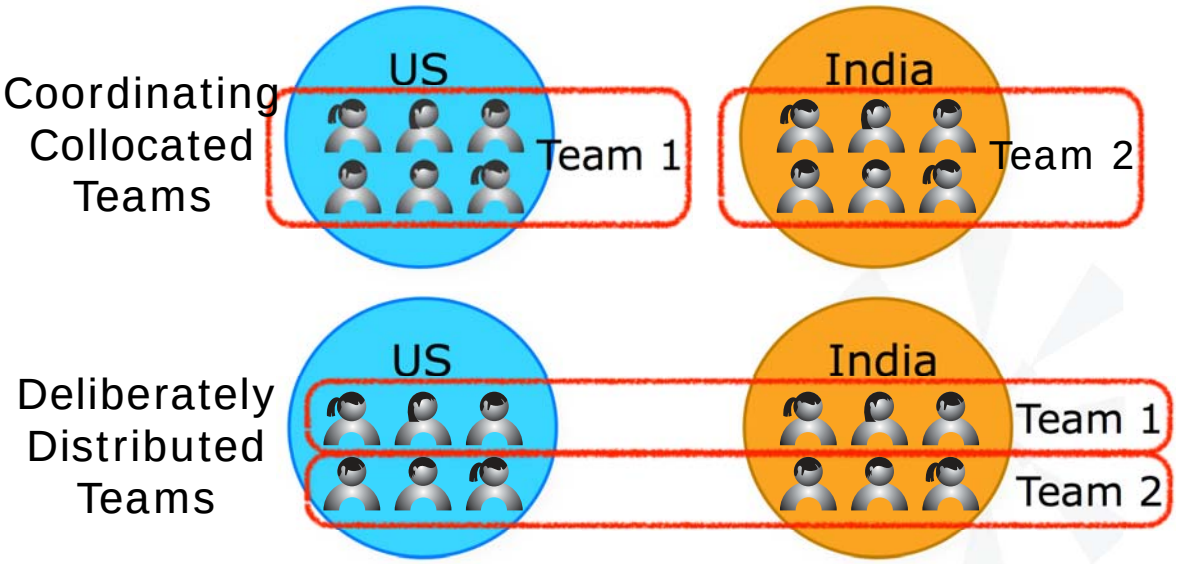
Scaling Question #1

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

Discipline Teams



Location Teams



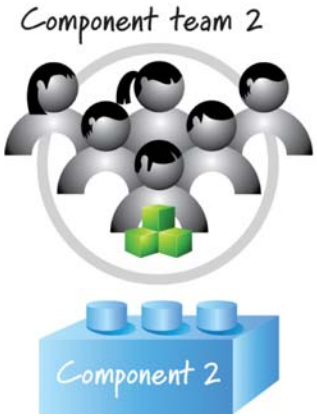
Architectural Layer Teams

GUI

Middle Tier

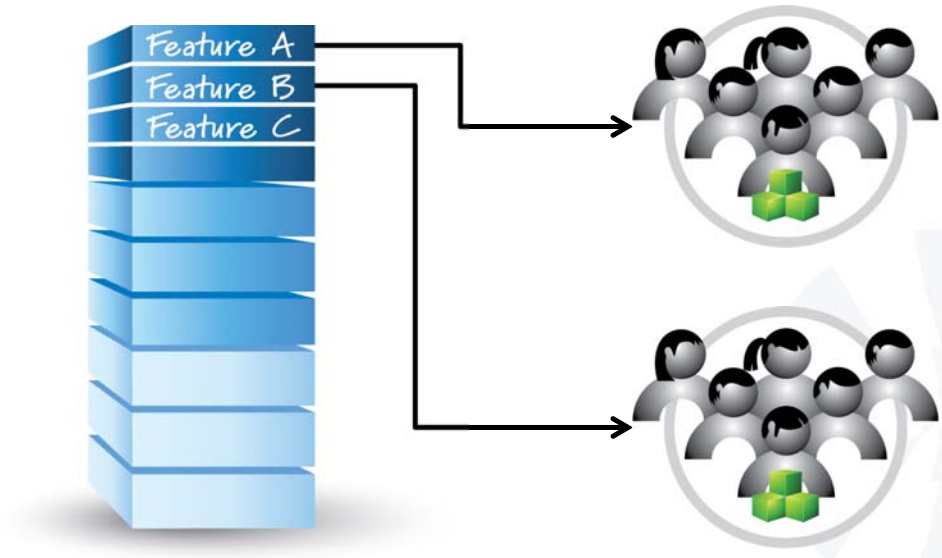
DB

Component Teams



Example component: Routing algorithms (component) inside of a GPS

Feature Teams

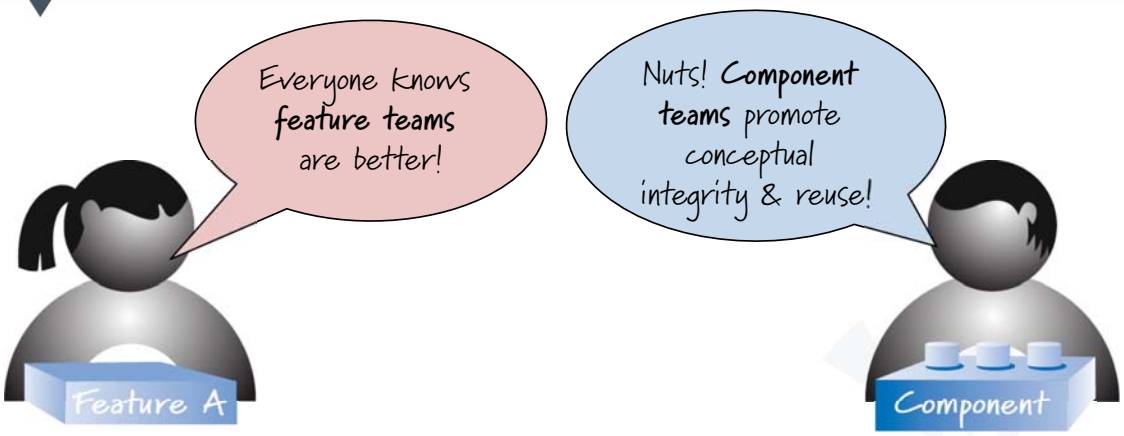


Scaling Questions #2

So, which approach do you prefer?

What criteria are you using to decide?

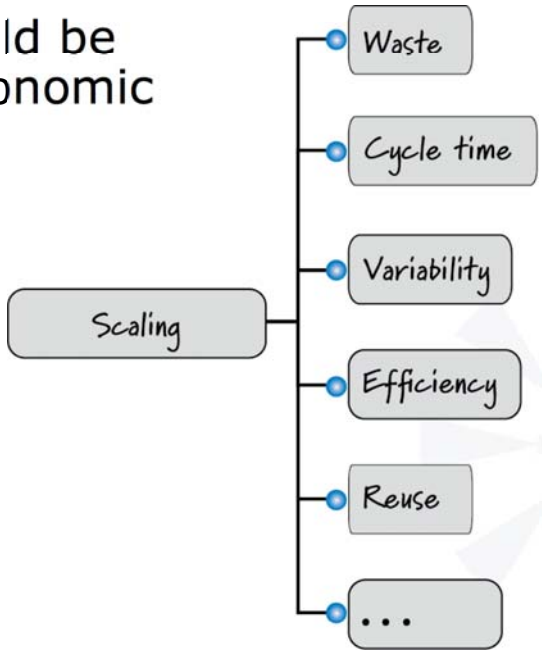
Don't Scale Based on Dogma!



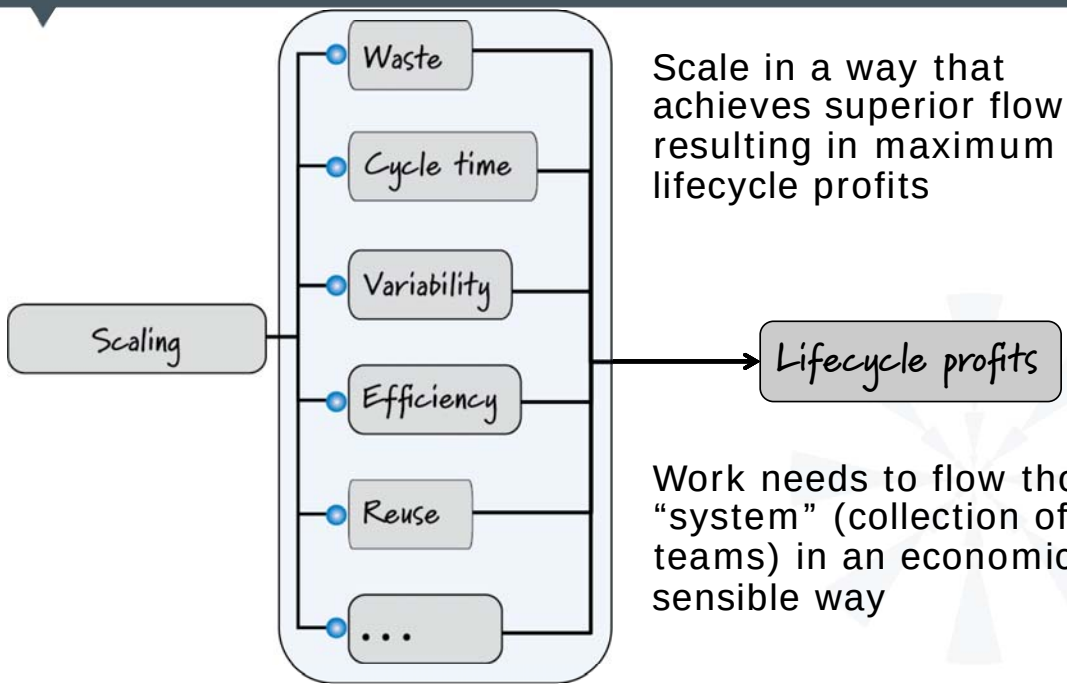
Do you think there is a single answer to scaling that universally applies to all situations (sizes and types of organizations)?

Scale Based on Economic Tradeoffs

- Scaling should be based on economic factors



Scale to Maximize Lifecycle Profits



Scale in a way that achieves superior flow resulting in maximum lifecycle profits

Work needs to flow though "system" (collection of teams) in an economically sensible way

Based on Reinertsen 2009.

Waste

Multiple forms of waste

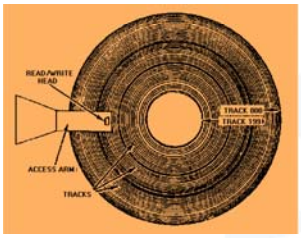
Can't eliminate them all

Determine which cause most economic damage

Recognize Inventory (WIP) Waste

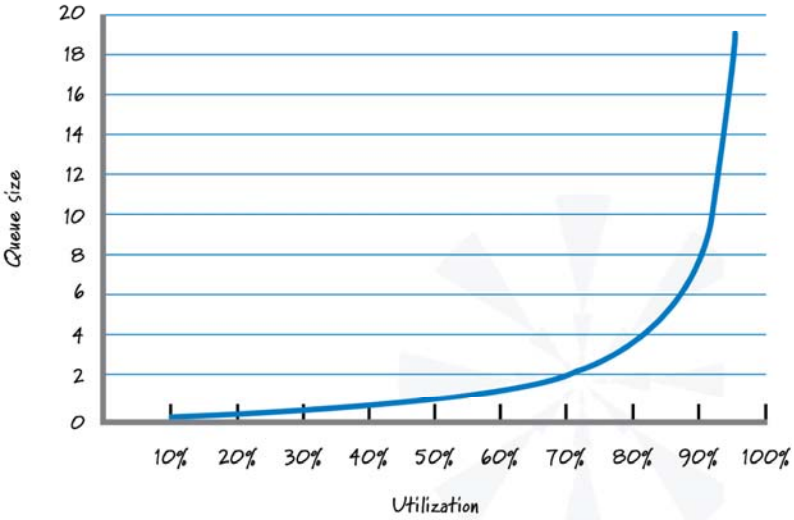
Manufacturing inventory is both physically and financially visible

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

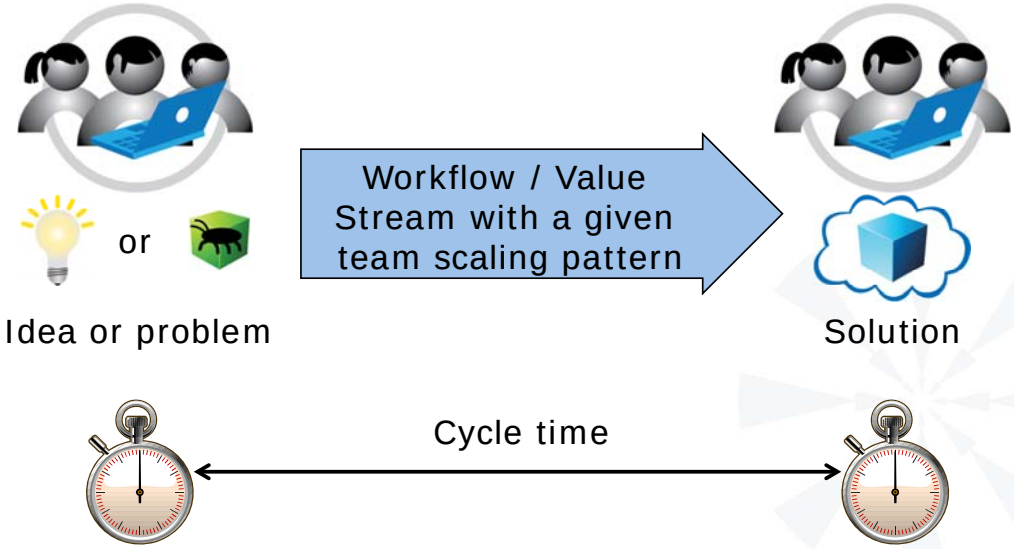


Focus on Idle Work Not Idle Workers

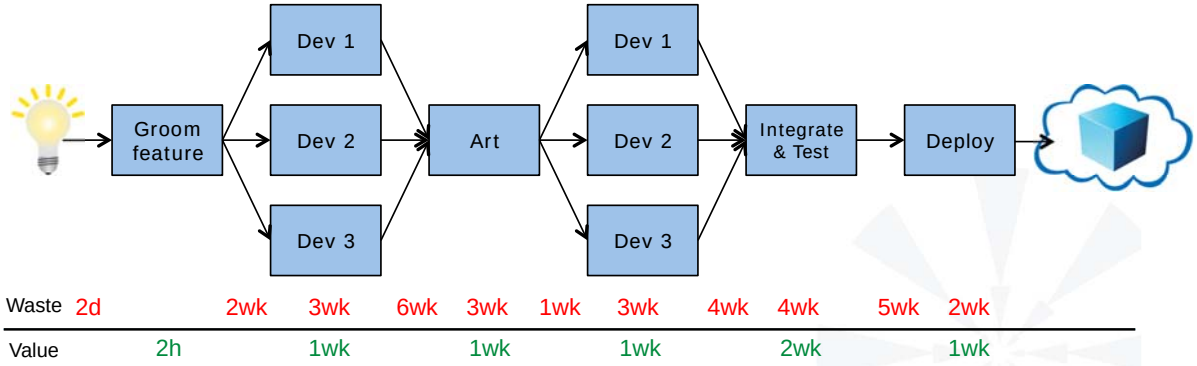
Watch the Baton Not the Runners



Cycle Time



Example Workflow / Value Stream



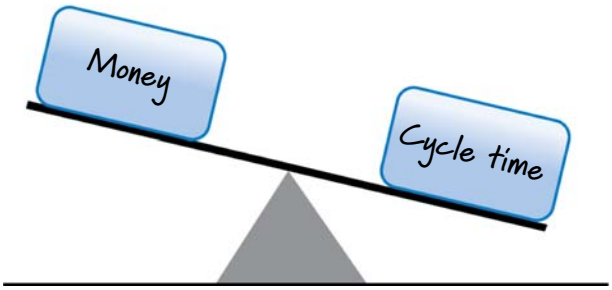
$$\frac{\text{6 wk value-adding time}}{\text{39.4 wk cycle time}} = 15\%$$ Process cycle efficiency

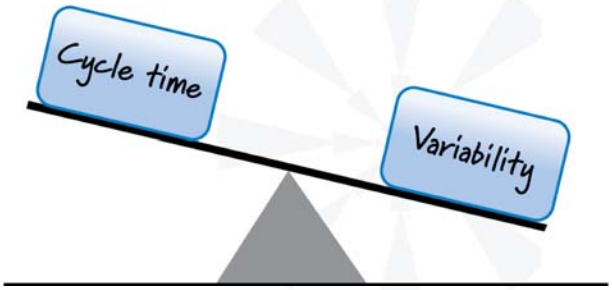
Improve team efficiency 10% yields 1.5% improvement

Eliminate 10% waste yields 8.5% improvement

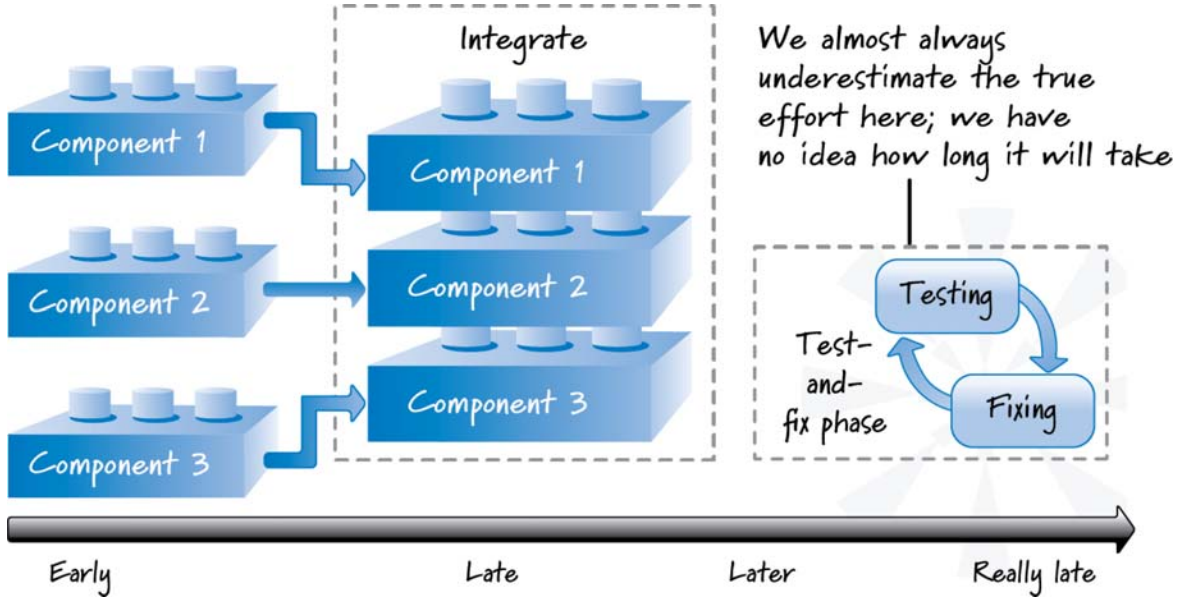
Cost of Delay

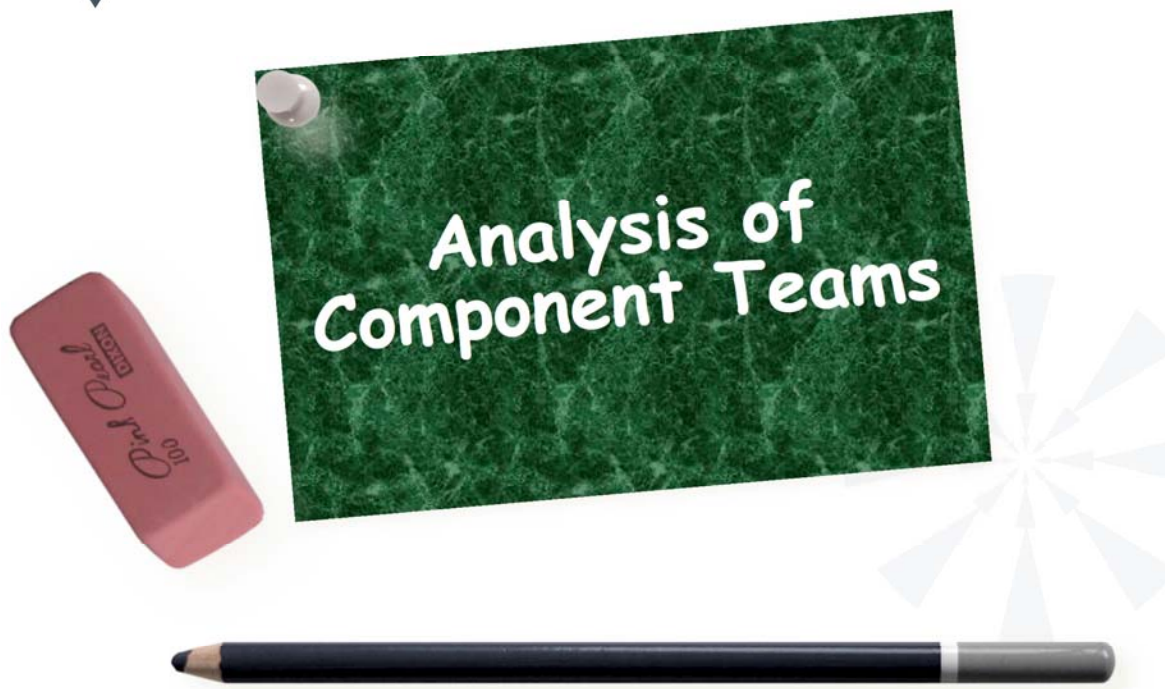
If you have to wait 6 weeks for the Art team to draw your art, and that delay could be eliminated by having an artist on your team, what is the cost of the Art-team delay (in lifecycle profits)?



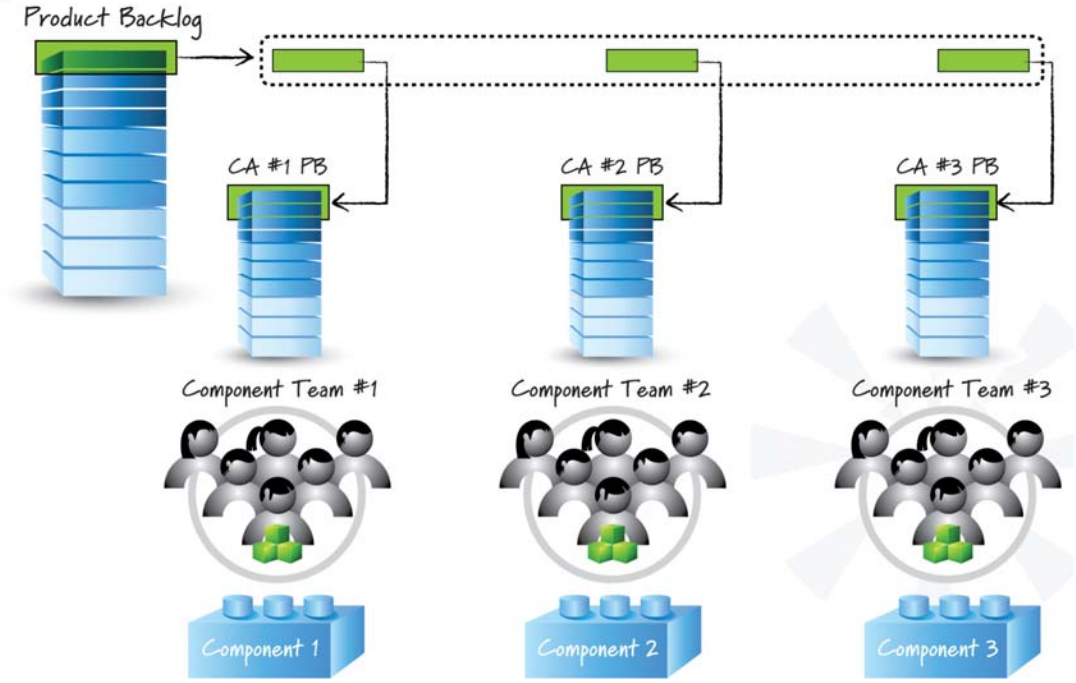


Organize Teams to Validate Important Assumptions Fast

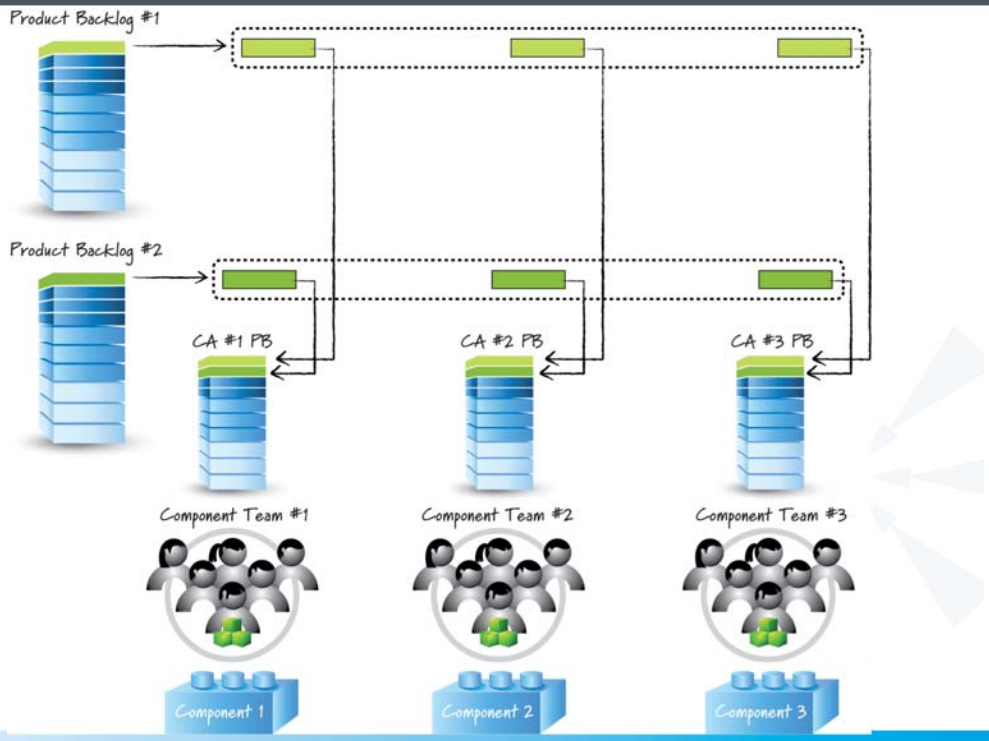




Component Teams (Single Source)



Component Teams (Multiple Sources)



Issue – Prioritization

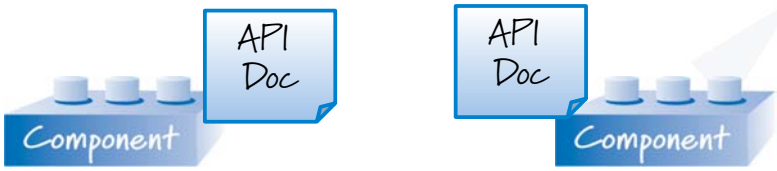
Localized prioritization decisions

Feature prioritization driven by component team availability

NPF (nosiest person first) dominates

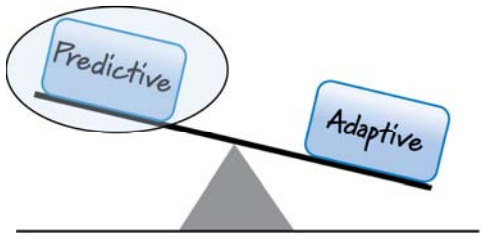
Issue – Coordination Costs

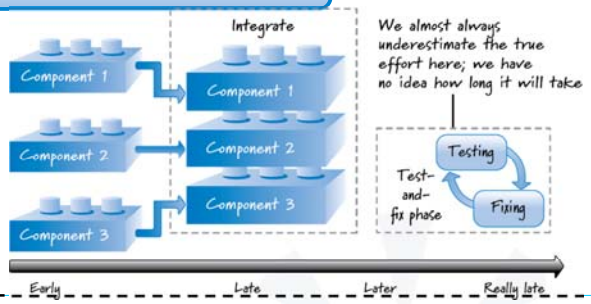
- Requires significant and on-going planning, handoffs, and dependency management
- At scale dependency management becomes economically intractable
- Favors low-bandwidth means of communication (e.g., interaction by contracts)



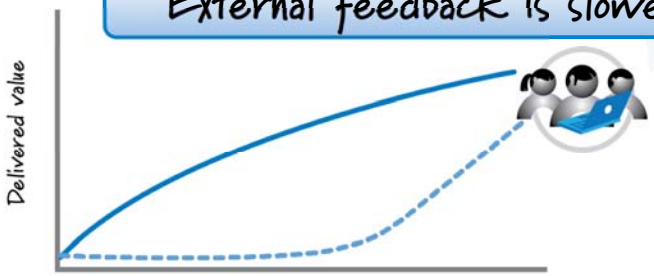
Issue – Slower Feedback

Internal feedback is slower

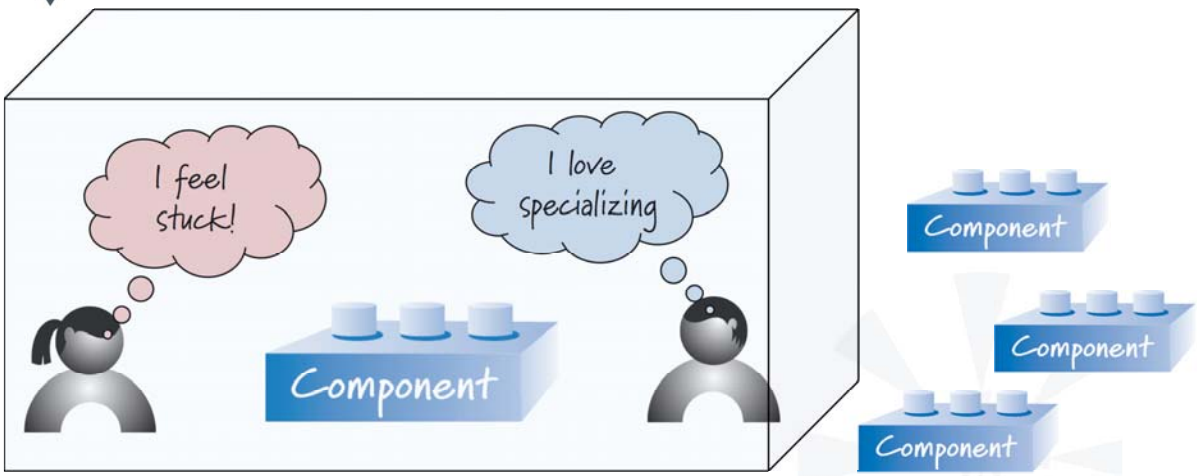




External feedback is slower



Issue – Limits Learning

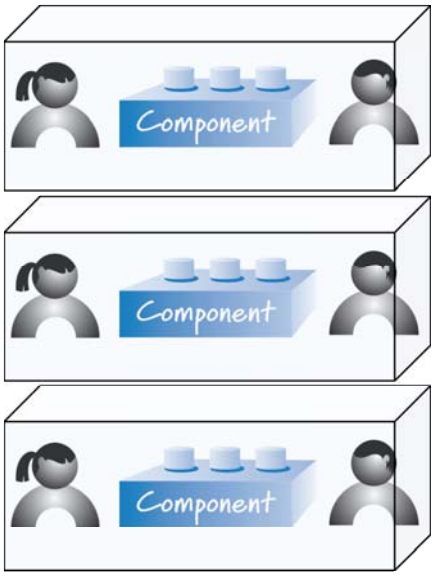


Risky: specialty knowledge in only a few heads

Issue – Harder to See the Whole

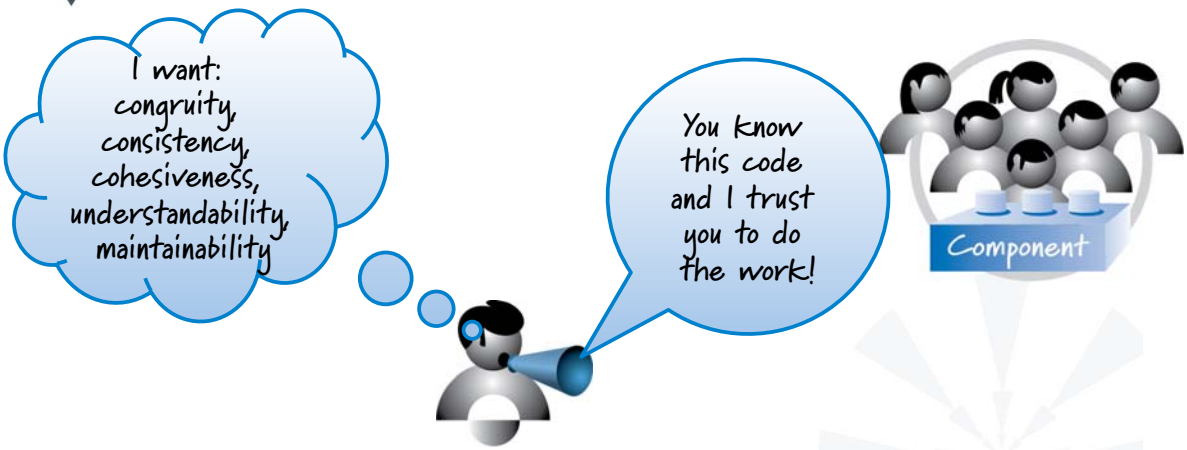
Best components ever!

But still a poor product



Alignment trumps local excellence

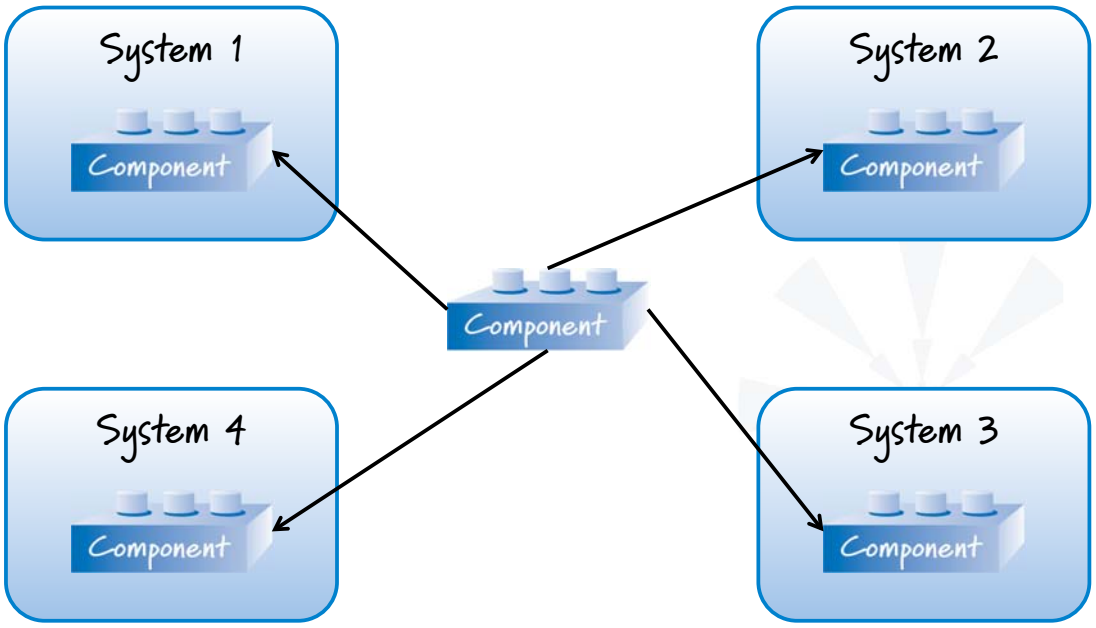
Desirable Property – Conceptual Integrity



Want conceptual integrity both at component and full system/product level

Conceptual integrity at the component level does not guarantee conceptual integrity at the product level

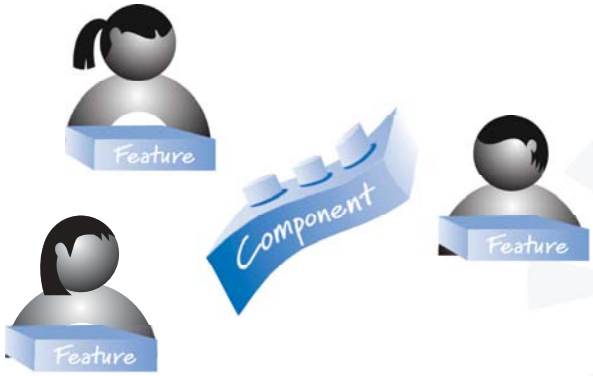
Desirable Property – Asset Reuse



Issue – Lack of Conceptual Integrity

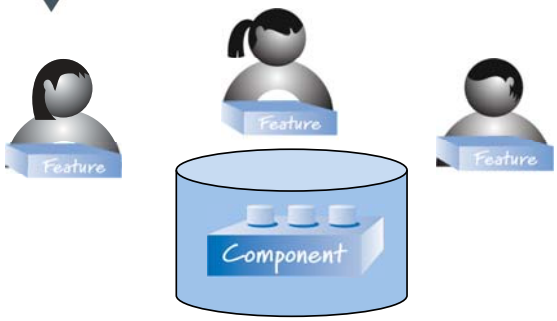
Incompatible changes

Shared design

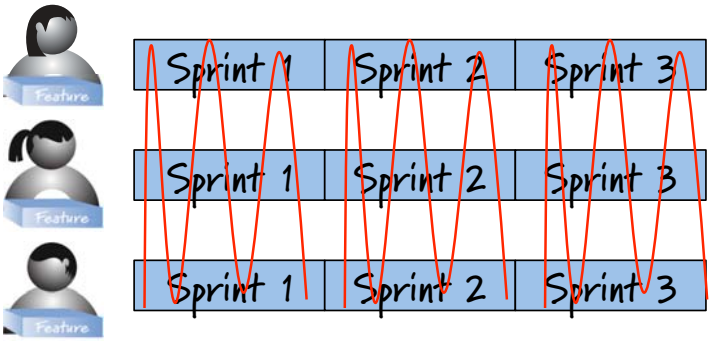


Who owns it?

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?

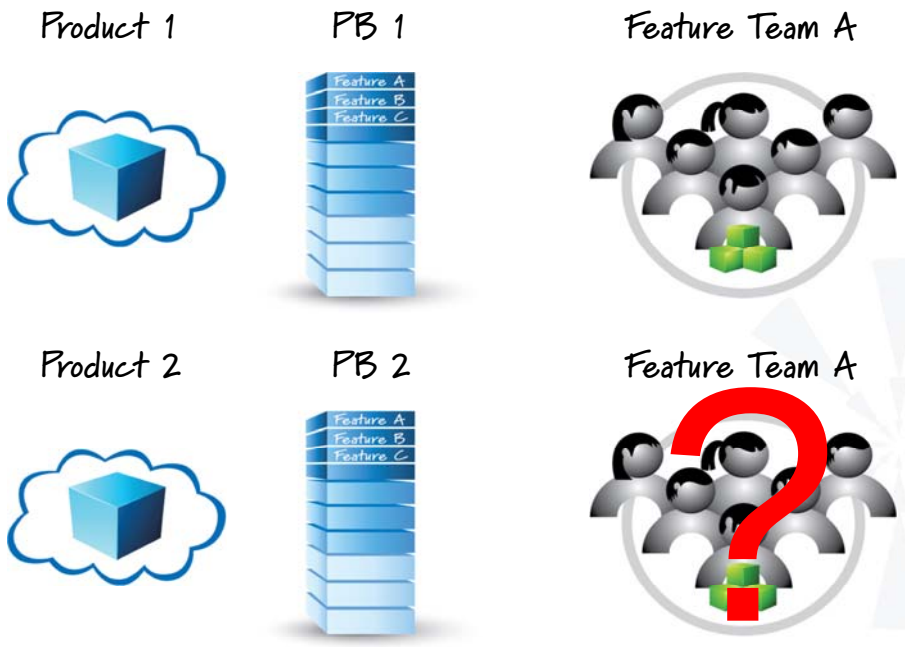
As a customer, I want to be one of 10,000 customers who can use the system during peak usage periods.

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

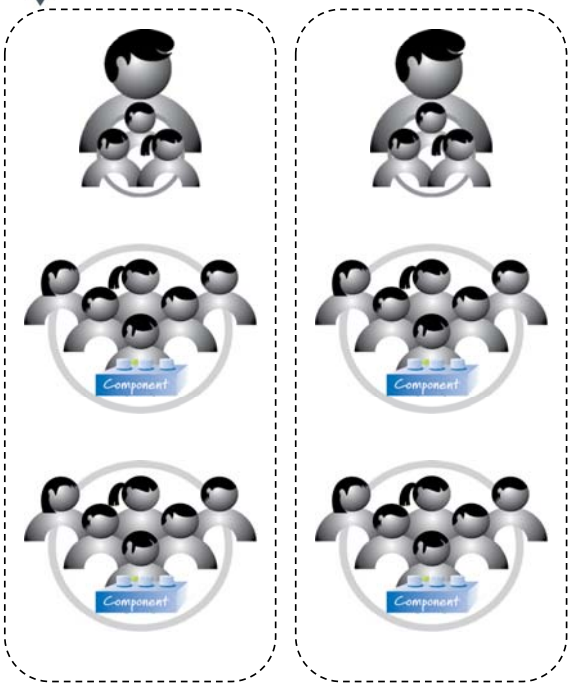
As the CTO, I want the new system to conform to our established security policies.

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

Issue – Team Longevity

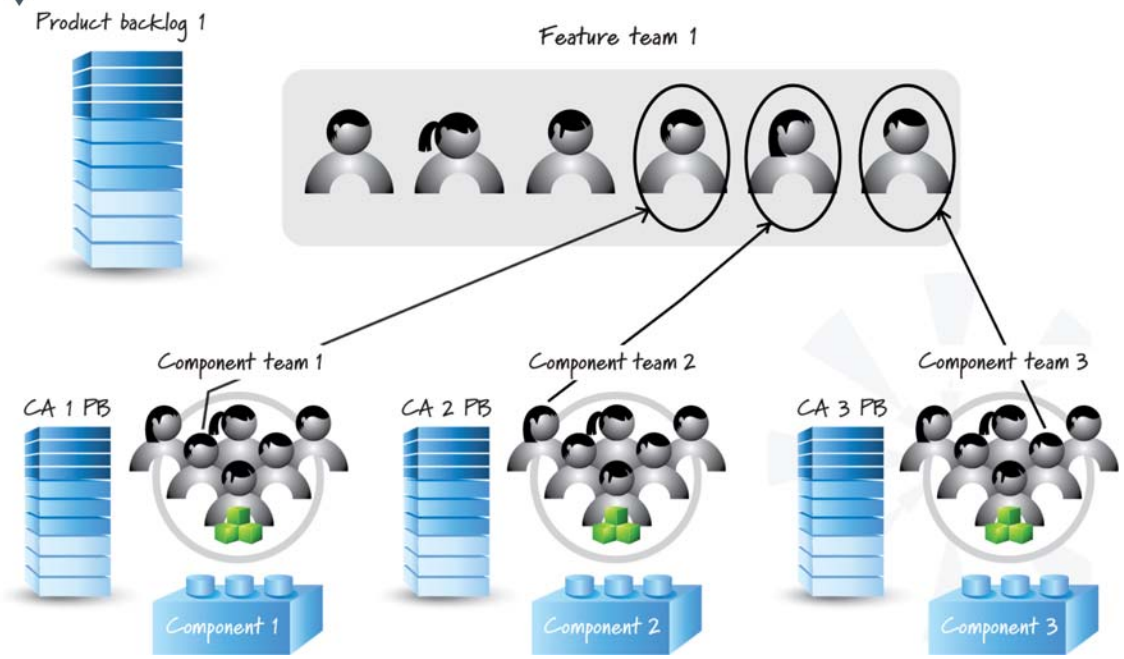


Issue – Organizational Resistance

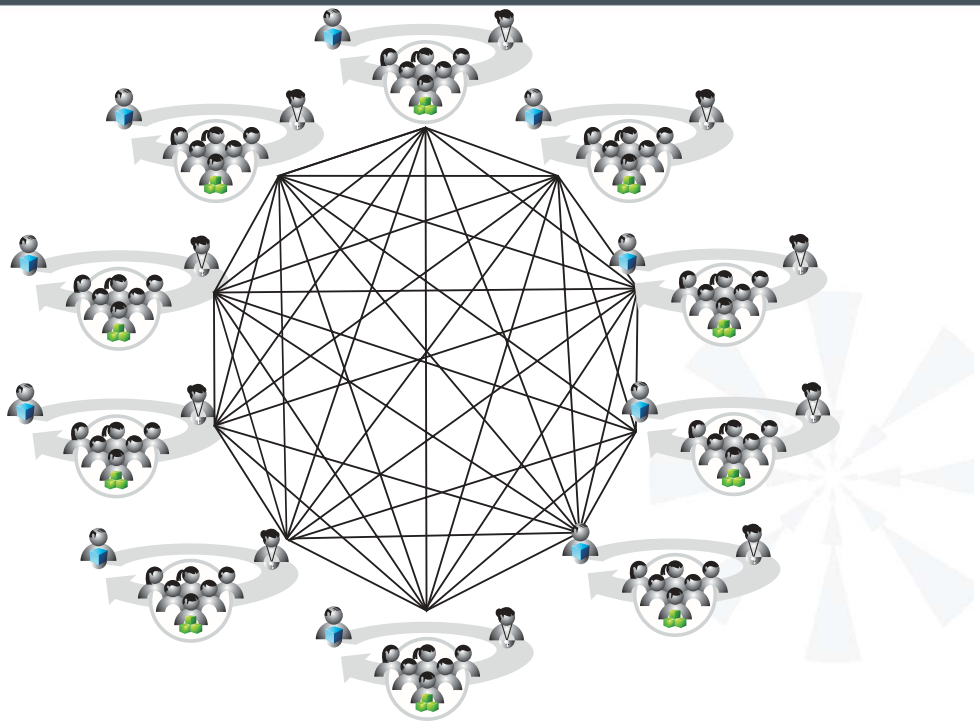


- Interferes with fiefdoms
- Too hard to reorganize into feature teams
- A general belief that feature teams will lead to significant technical debt

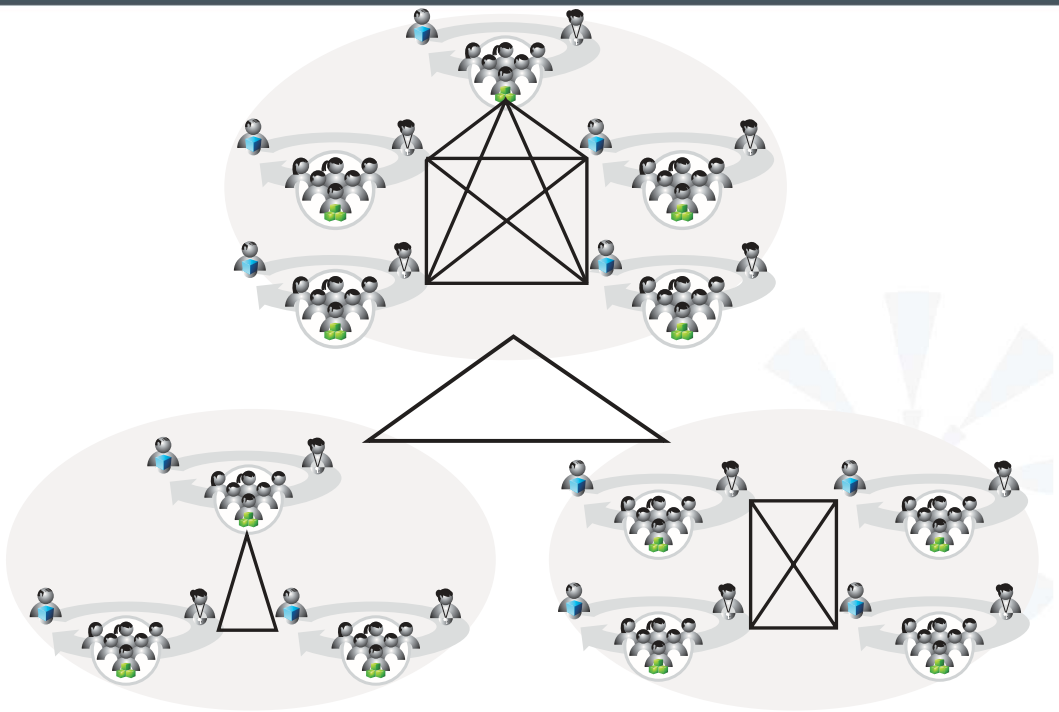
Combined Feature & Component Teams



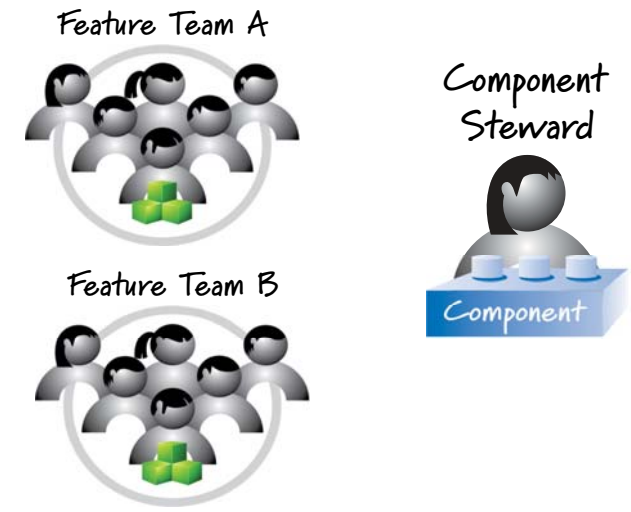
Teams with Fully Connected Communication Channels



Teams Form Collaboration Clusters



Component Stewards / Guardians



I teach other people about component

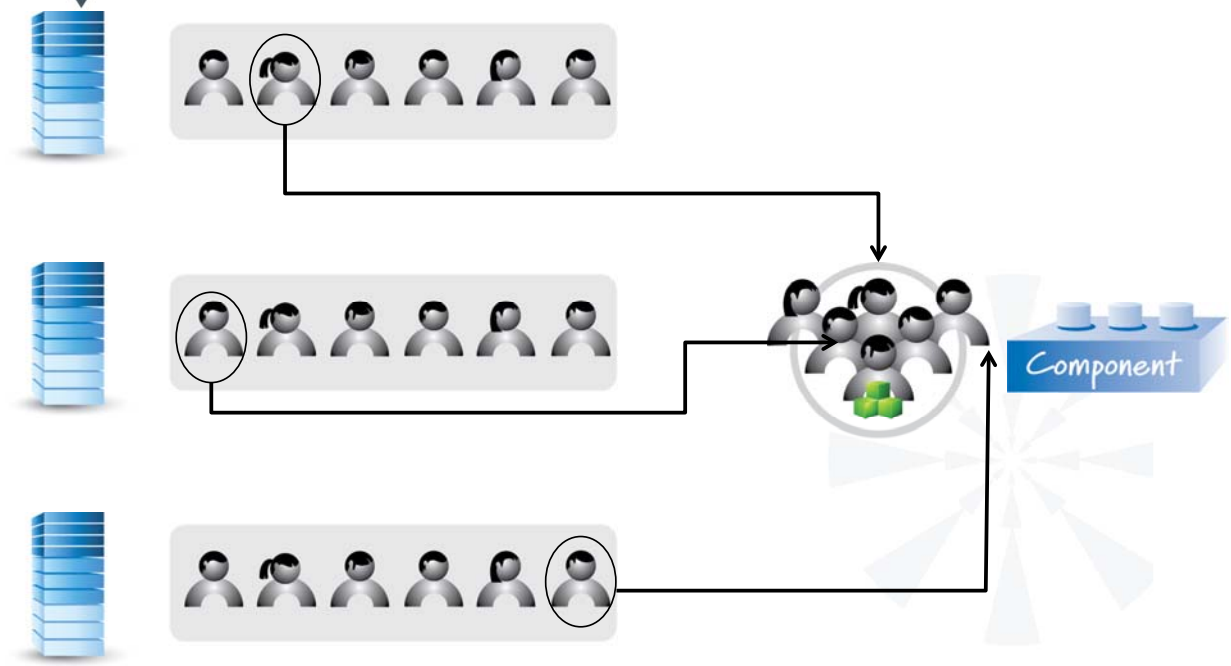
Ensure changes maintain or improve conceptual integrity

Take a leadership role in promoting reuse

I don't "own" the component

Feature teams make component changes

Create a Community of Practice from Feature Team Members



Top Down System Level Approach

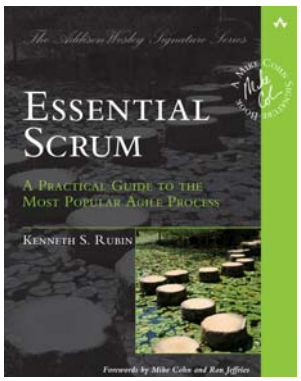
What are your products?

What are your product backlogs?

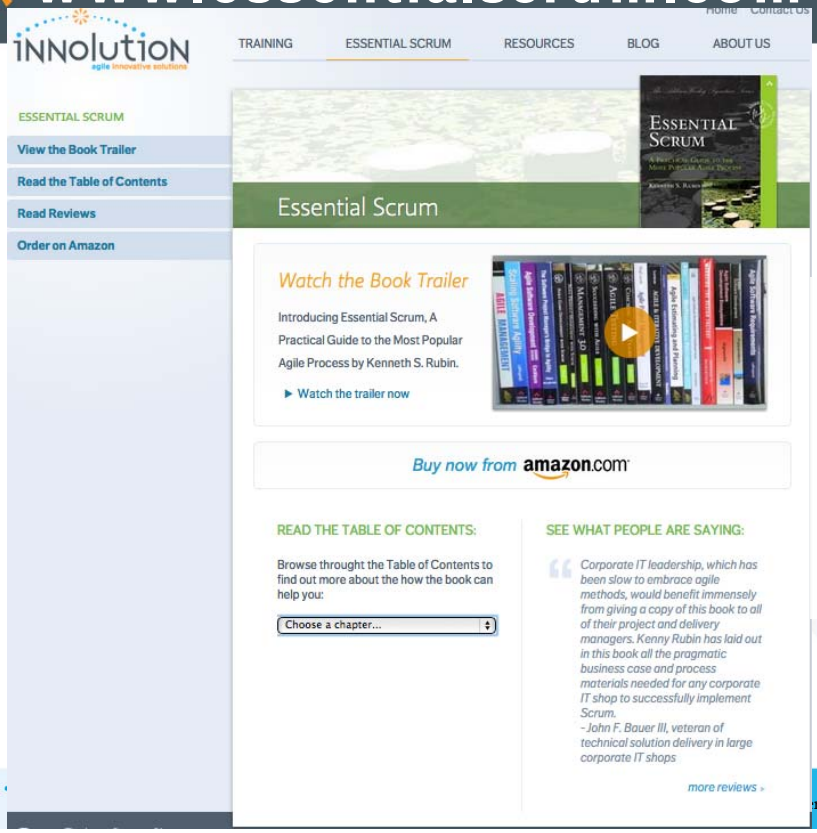
What teams do you need to deliver on your goals?

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



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