

Use Visual Studio to Scale Agile in your Enterprise


Richard Hundhausen
Scrum / DevOps Trainer
Accentient, Inc.



Yesterday's Knowledge;
Tomorrow's Code!



About Me

- From Boise, Idaho, USA
- President of Accentient
- Microsoft MVP (Dev Technologies)
- Professional Scrum Developer
- Professional Scrum Trainer
- Co-creator of the Nexus (scaled Scrum Fx)
- richard@accentient.com
-  @rhundhausen

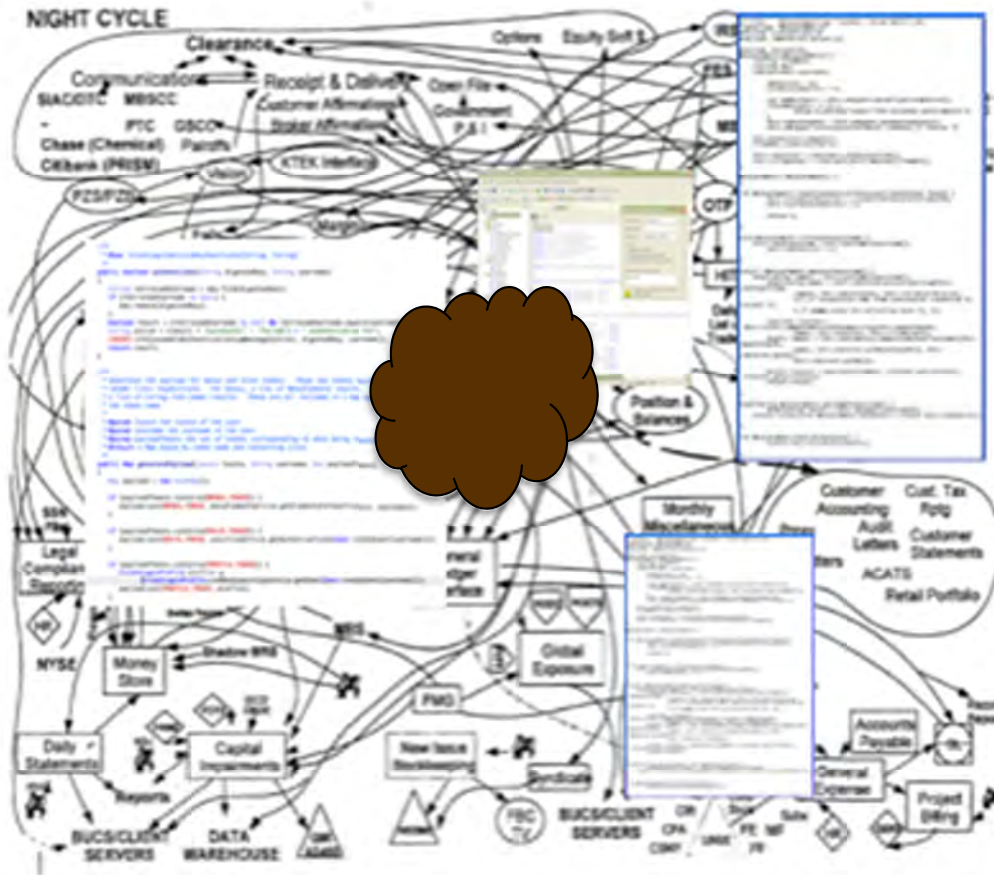


First, do you really need to scale?

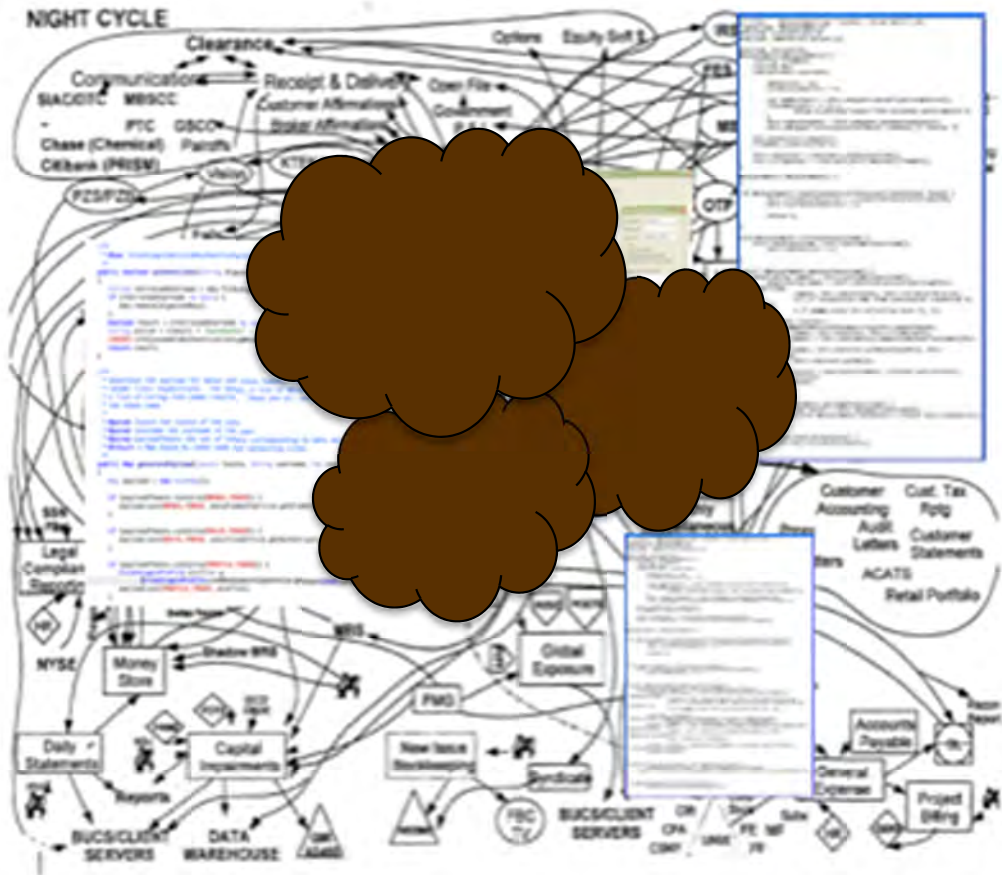
Organizations Want to Scale

- They see great results with small initiatives and want similar benefits from larger initiatives
- They are looking for a straightforward approach that builds on their current knowledge and skills

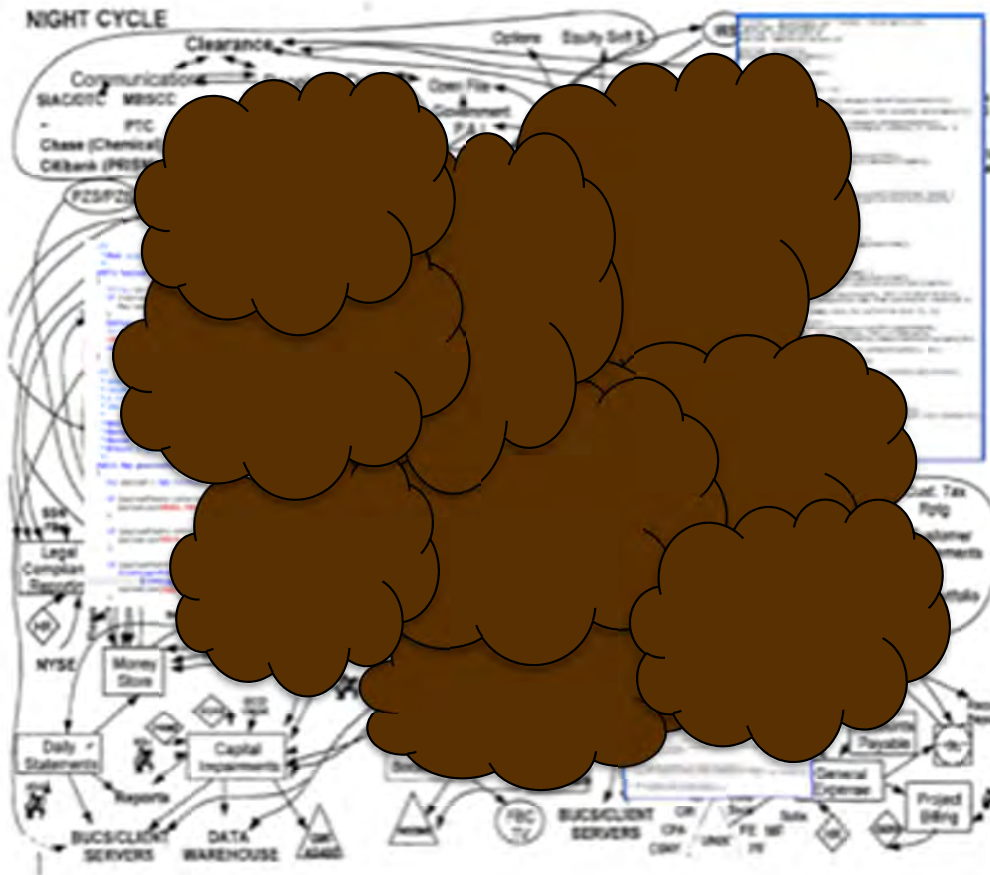
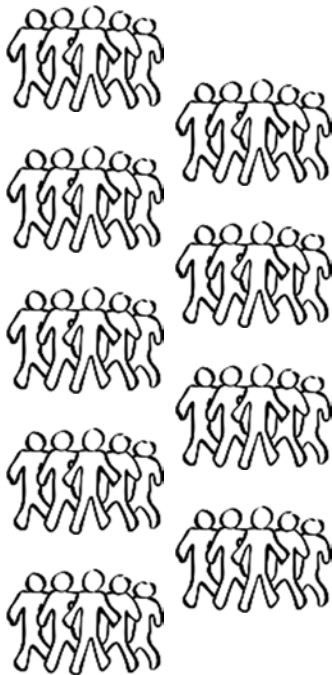
One Team Doing Work



Three Teams Doing Work



Nine Teams Doing Work



Scaled Agile Development is ...

- Multiple teams working on a single product
- Ideally with ...
 - A single Product Backlog
 - A single Product Owner (with helpers)
 - Cross-functional “feature” teams
 - Teams on the same cadence

Scaled Agile Development is not ...

- Organizational transformation
 - “My organization wants to be more agile”
- Spinning up more agile teams
 - “IT is using Scrum so we should try it in our group”
- Adding more people
 - “That team is always behind, add more resources”

What is Scaled Agile Development?

Number of Products	>1	Painful	Portfolio Management
	1	Scrum	Scaling!
		1	>1
		Number of Teams	

A better question to ask ...

“How can we simplify the unnecessarily big and complex organizational design, and be agile rather than do agile?”

- Craig Larman, co-creator of LeSS

Ironically ...

- You can do more with 
 - Removing teams
 - Removing people
 - Removing tools & process
 - Removing management
 - Removing structure
- You'll need good Scrum Masters and coaches!

But this is hard (impossible?)

See www.larmanslaws.org for why

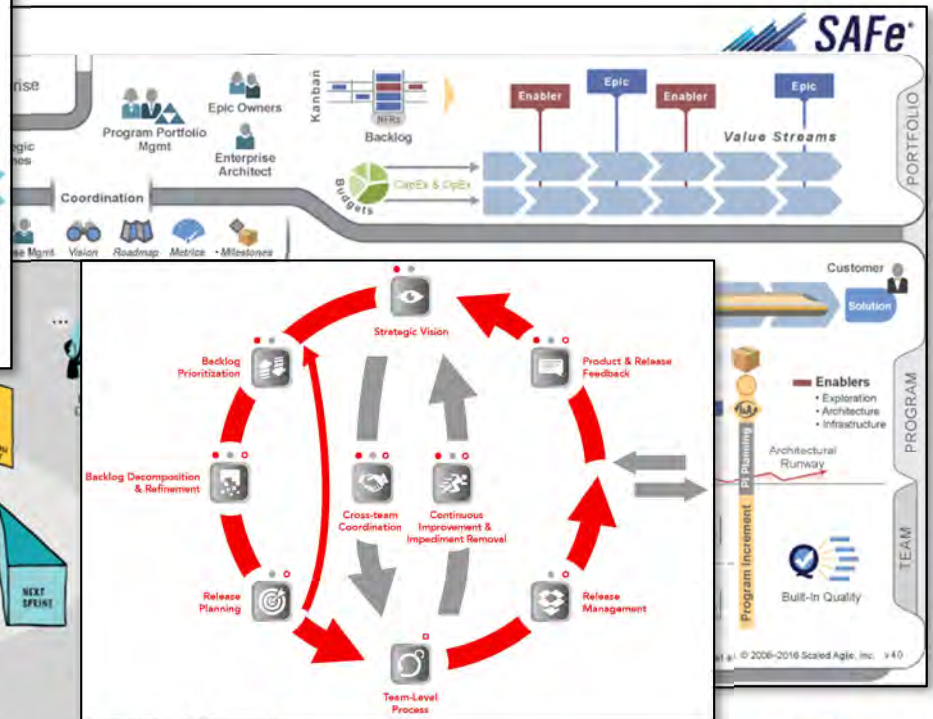
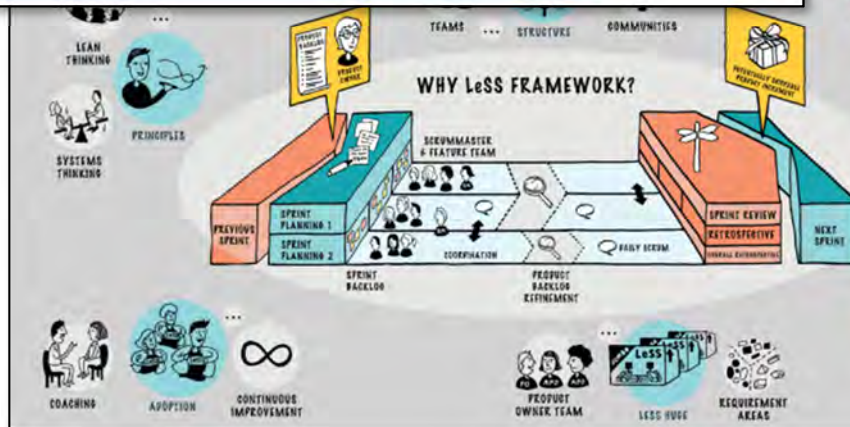
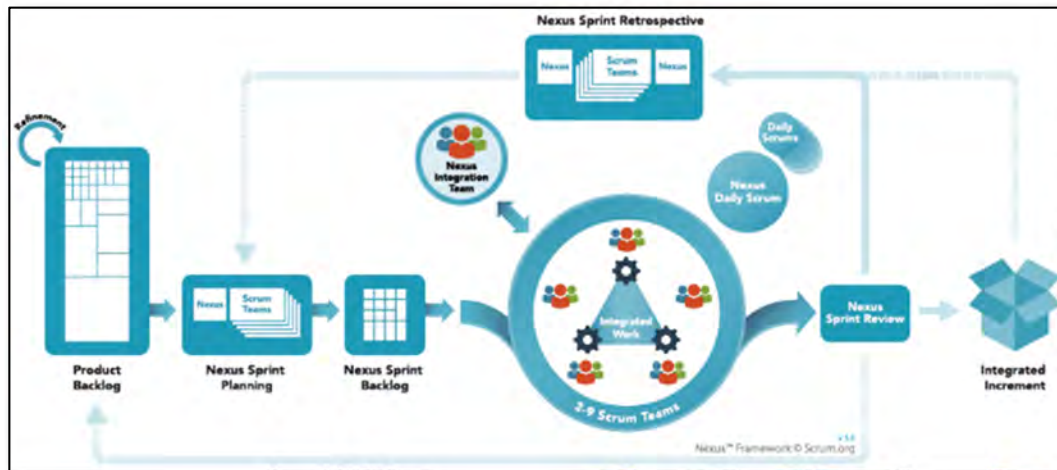
“It is difficult to get a man to understand something, when his salary depends upon his not understanding it!”

- Upton Sinclair

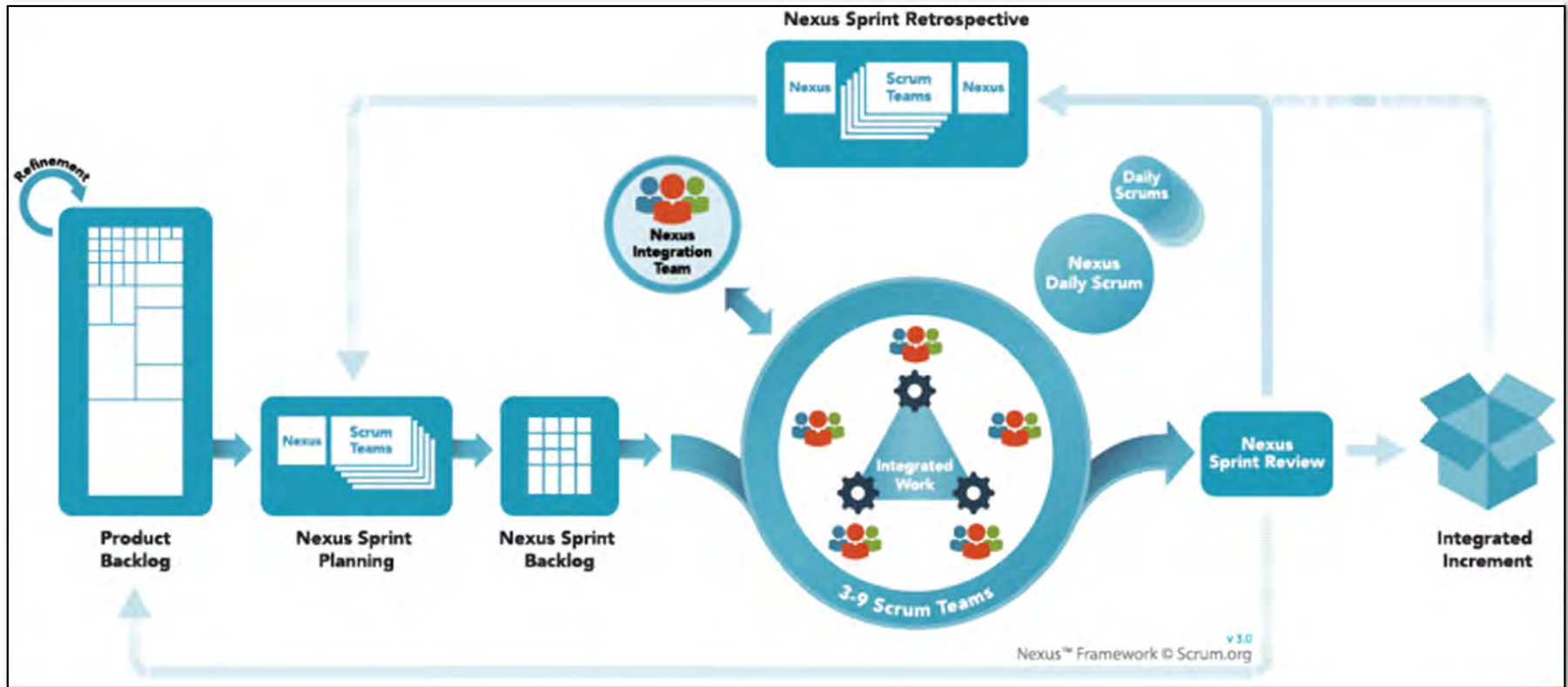
So, do you have a framework?

(or at least some proven experiments)

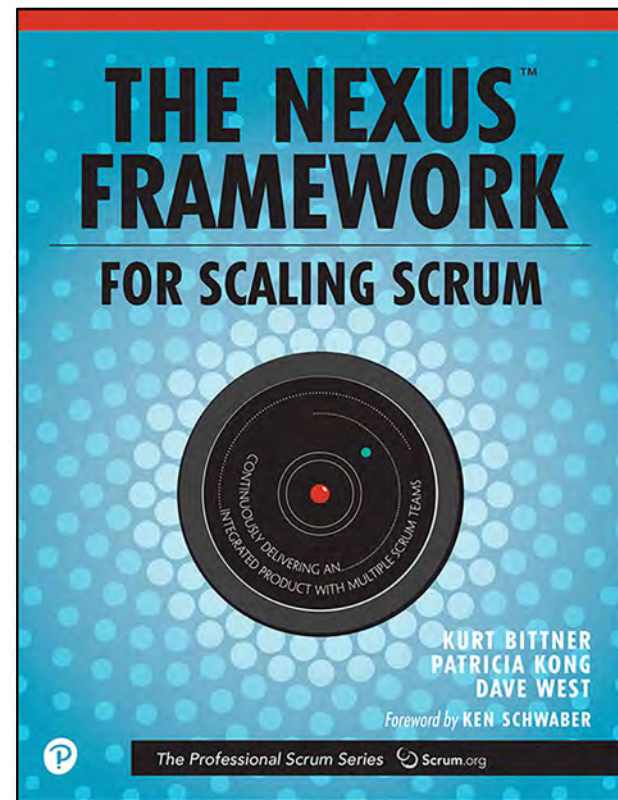
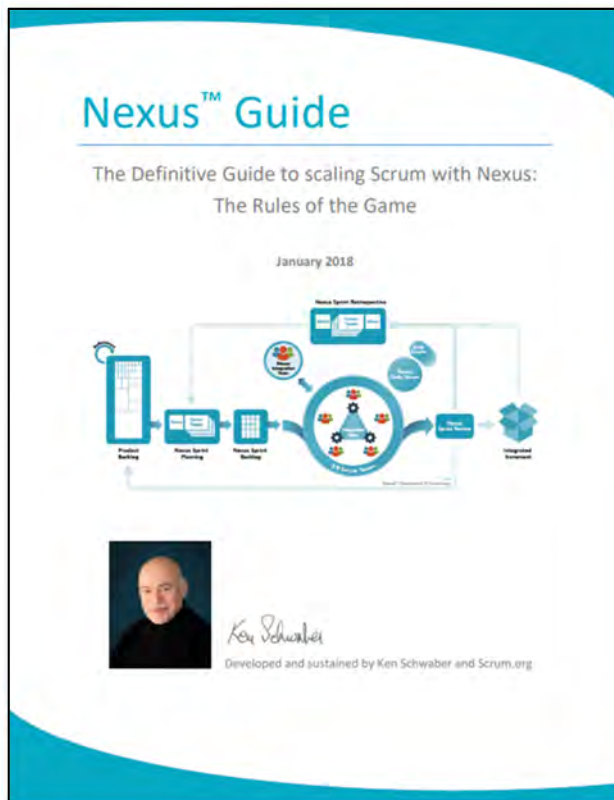
Nexus, LeSS, SAFe, S@S, SoS, RYO



Scaled Professional Scrum (The “Nexus”)



Learn more about the Nexus

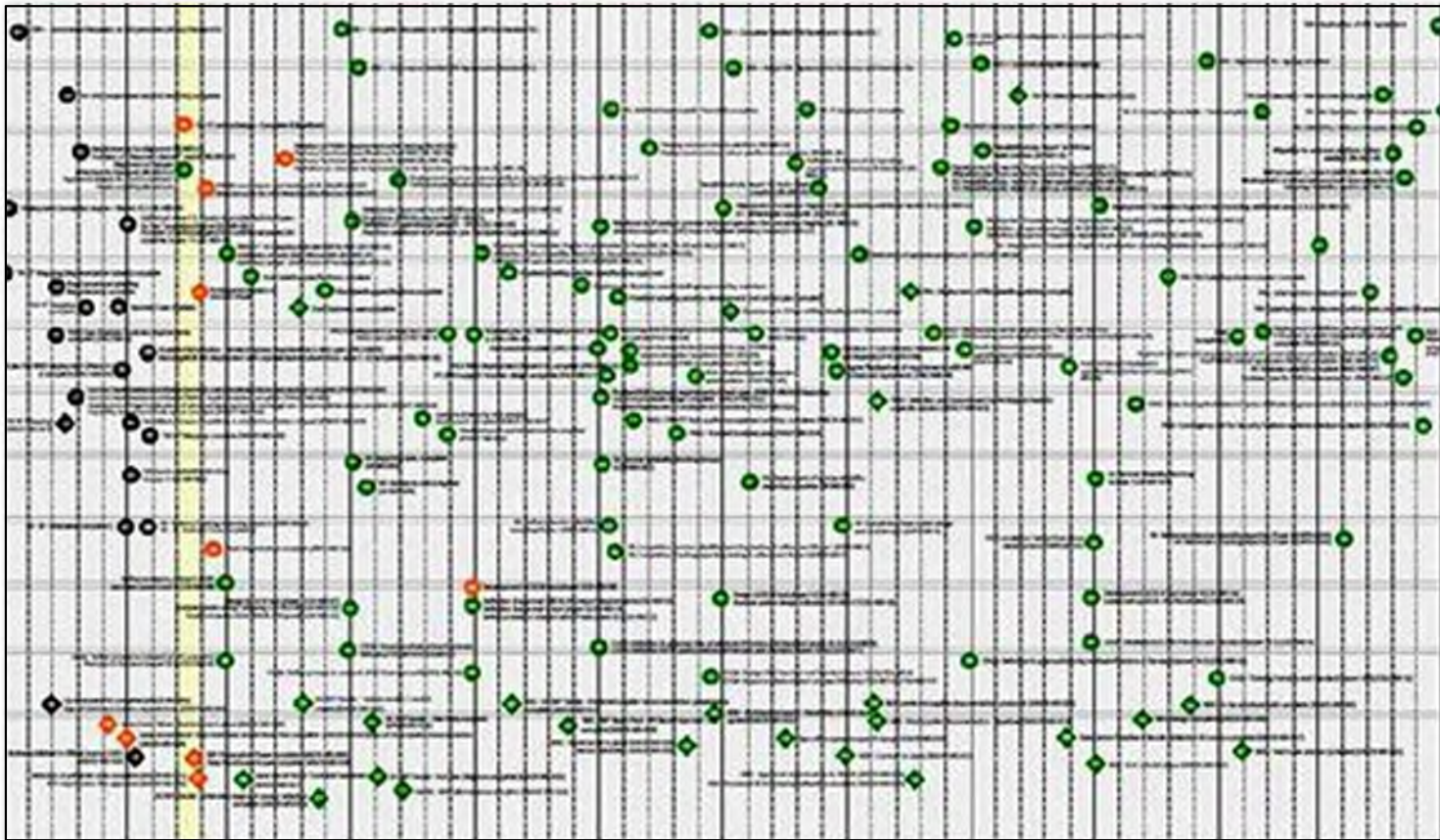


All VSTS Processes Support Scaling

- VSTS and TFS support scaled software development
 - Maybe not every detail of a specific framework
- All processes support scaling
 - Multiple and hierarchical product backlogs
 - Additional metadata fields on requirement and bug work item types

So, do you have a *product*?

Or just a bunch of projects ...



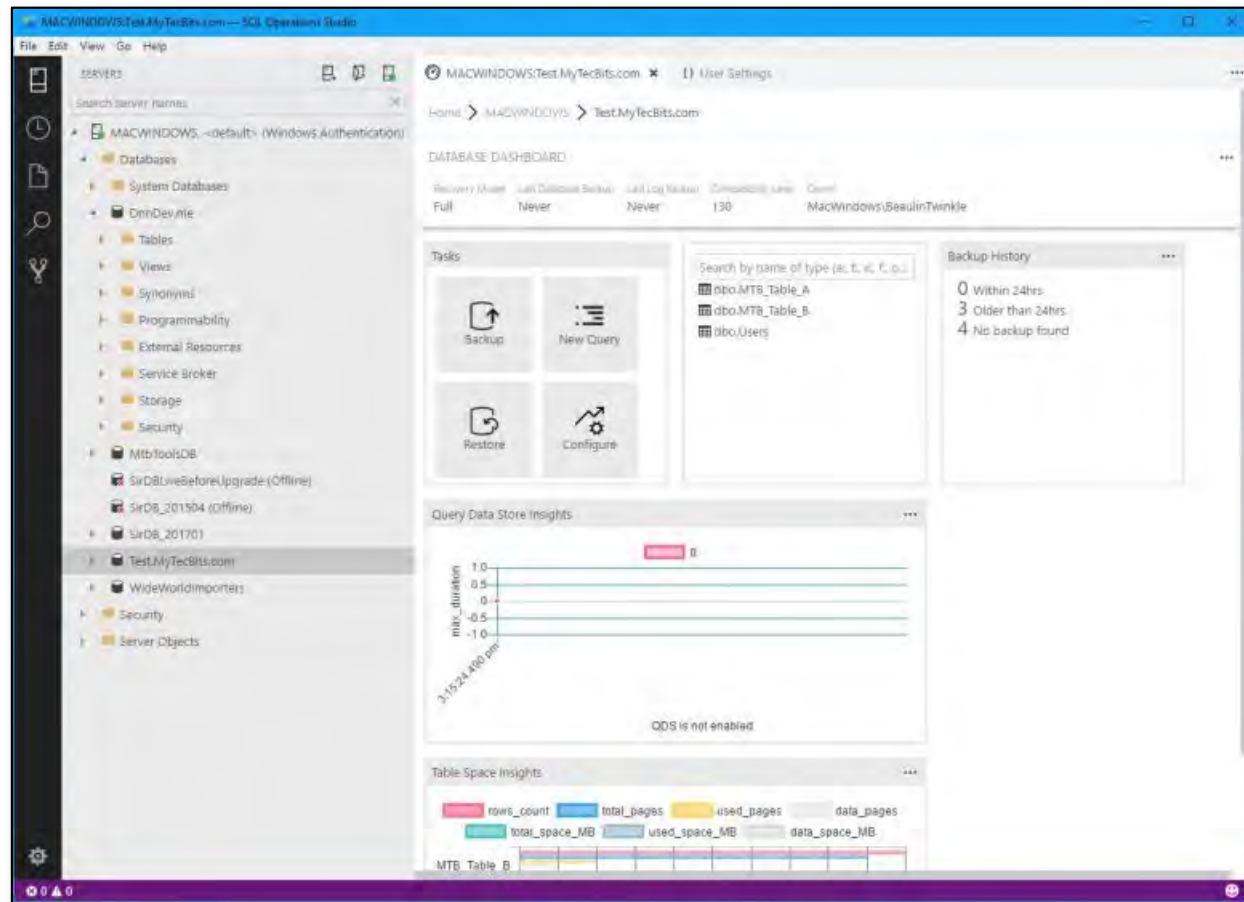
Product != Project

- It can be hard to identify the product behind all of your projects (and Visual Studio projects)
- Try ...
 - Asking your customers, users, stakeholders
 - Asking your codebase, UI
 - Start broad, and then ask narrowing questions

Start as Broadly as Possible

- Is your organization the product?
- Are multiple value streams the product?
- Is a single value stream the product?
- Is there an existing product manager/owner?
- Is there a cool acronym? (e.g. SCMODS)

Example: Where's the Product?



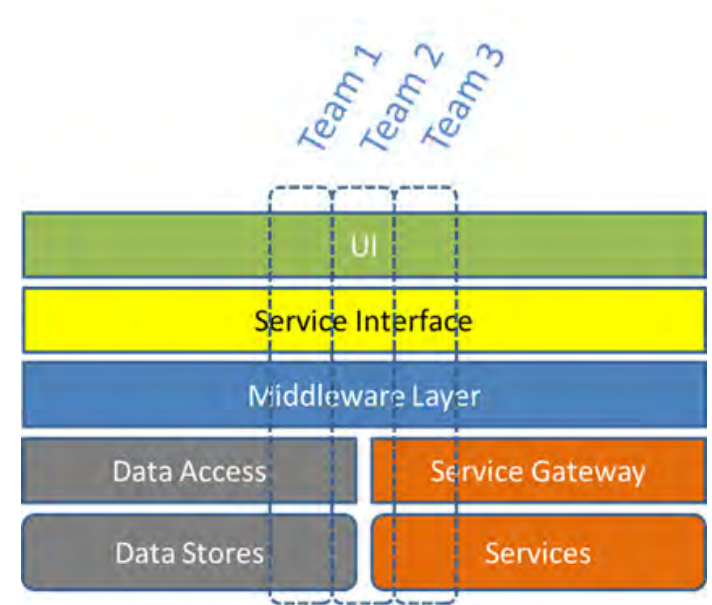
Product Areas

- In a large software development effort, the product may have many perspectives:
 - Platforms
 - Technologies
 - Geographical location
 - Specialties
 - Layers
 - Components
 - Features

Do you have the right *teams*?

Strive for Feature Teams

- Each team has all skills to turn Product Backlog items into increments of working software
- Vertical slicing; work is divided by end-user functionality
- Work is integrated continuously within each Sprint



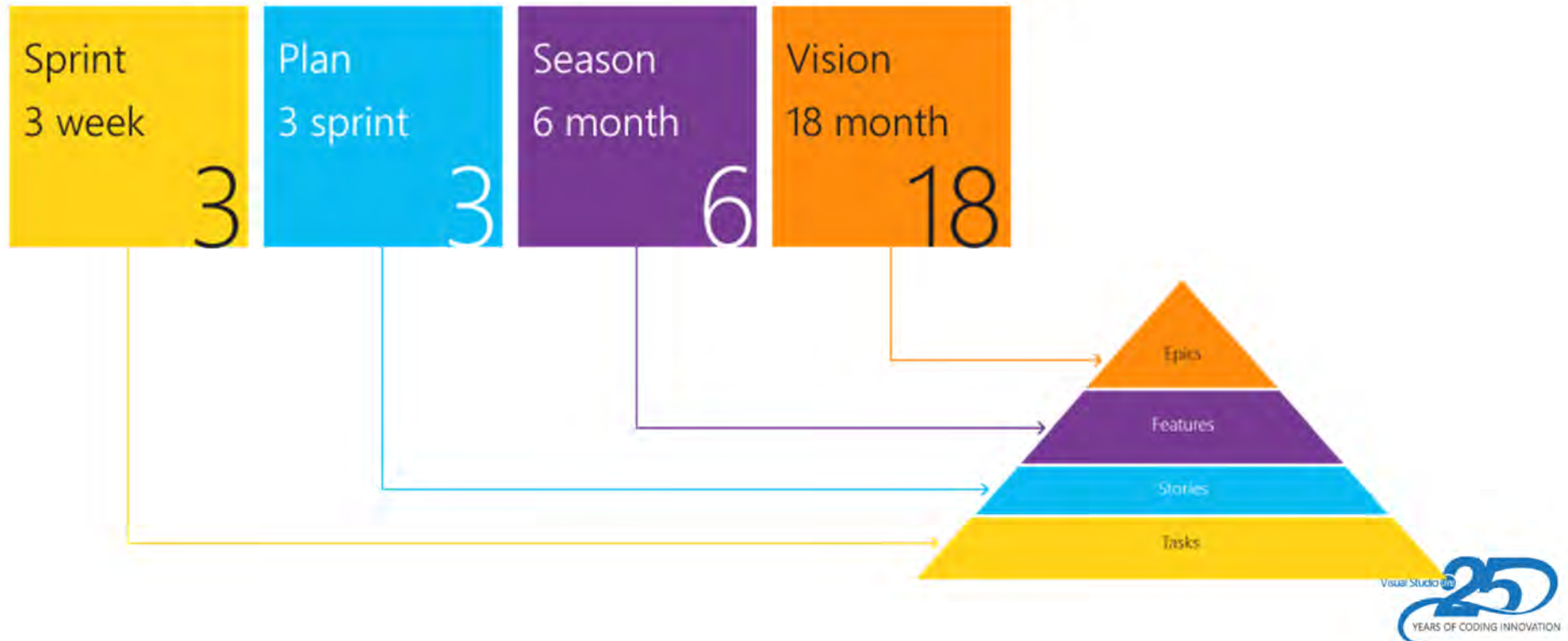
Multiple Teams in VSTS/TFS

- Create the teams
- Create a respective area path for each team
- Add team members

<http://bit.ly/1Q55dAh>

Scaled Agile Planning

- Example: “Rolling wave” planning ...



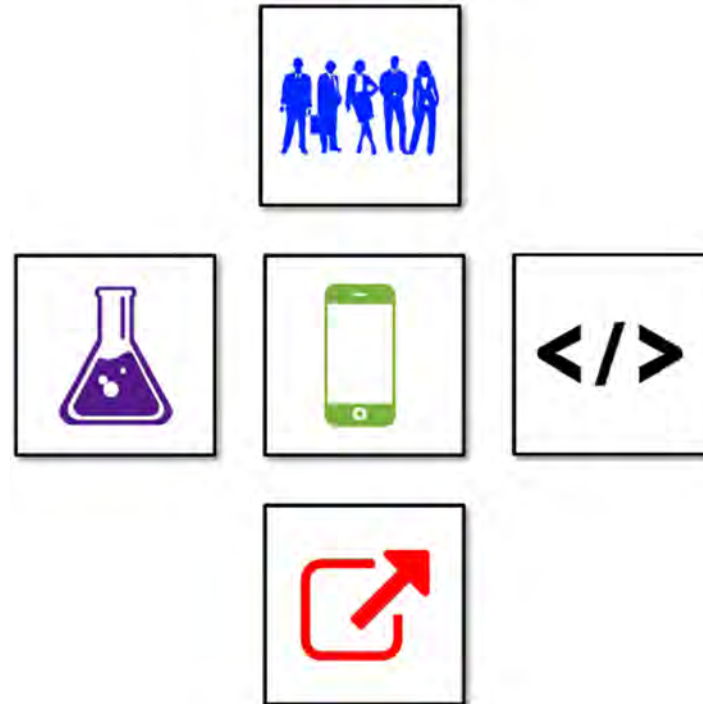
Are you *refining* the backlog?

Product Backlog Refinement

- Should be done regularly
- Should be done by the entire team
- Helps the Product Owner know the “cost”
- Helps identify dependencies early

Watch Out For Dependencies

- People
- Domain
- Technology
- Software
- External





**DONE IS
BETTER
THAN
PERFECT**

Done () ;

(thank you)

richard@accentient.com | [@rhundhausen](https://twitter.com/rhundhausen)



<https://github.com/rhundhausen>