Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS
Anaheim

★ CAMPAIGN FOR CODE 2016 ★

Use Visual Studio to Scale Agile in Your Enterprise

Richard Hundhausen

Consultant/Trainer

Accentient

About Me

- From Boise, Idaho, USA
- President of Accentient
- Microsoft Regional Director
- Microsoft MVP (Visual Studio ALM)
- Professional Scrum Developer
- Professional Scrum Trainer
- Co-creator of the Nexus
- richard@accentient.com

@rhundhausen

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

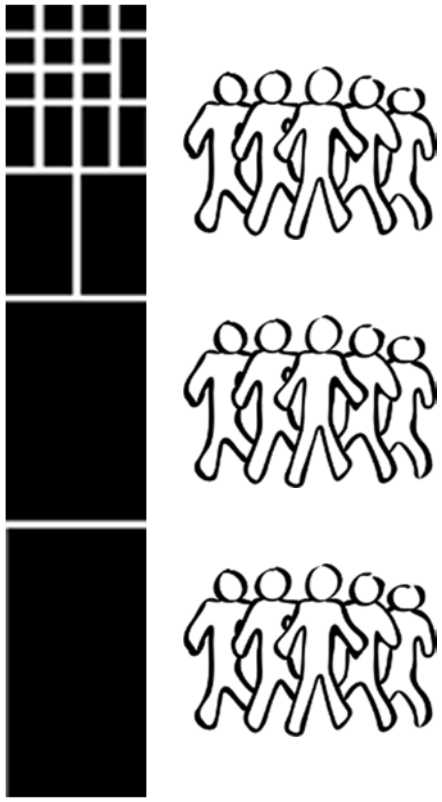
First, do you *need* to scale?

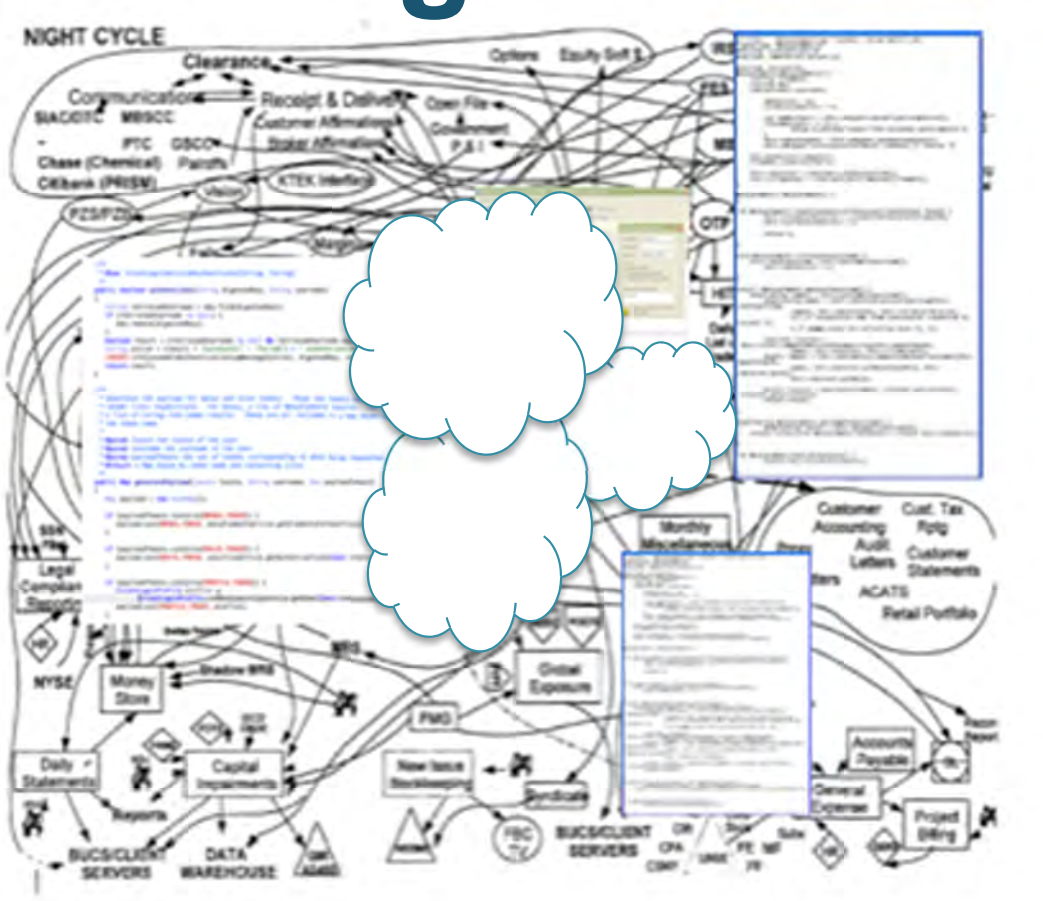
Organizations Want to Scale

- 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

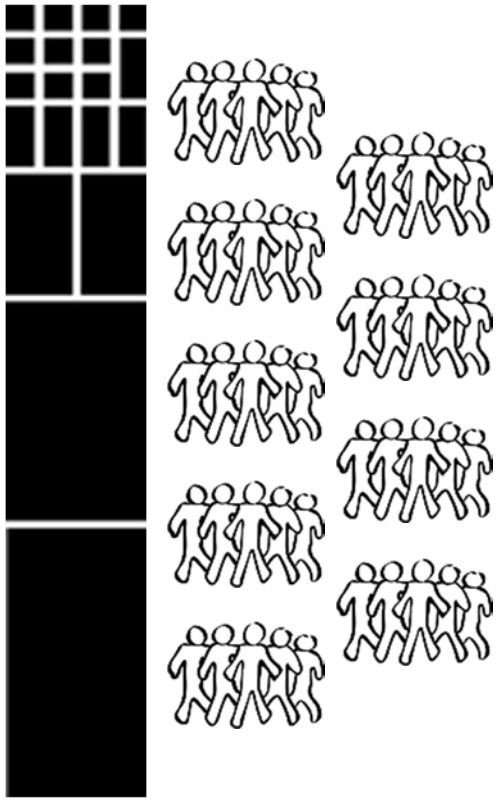
One Team Doing Work

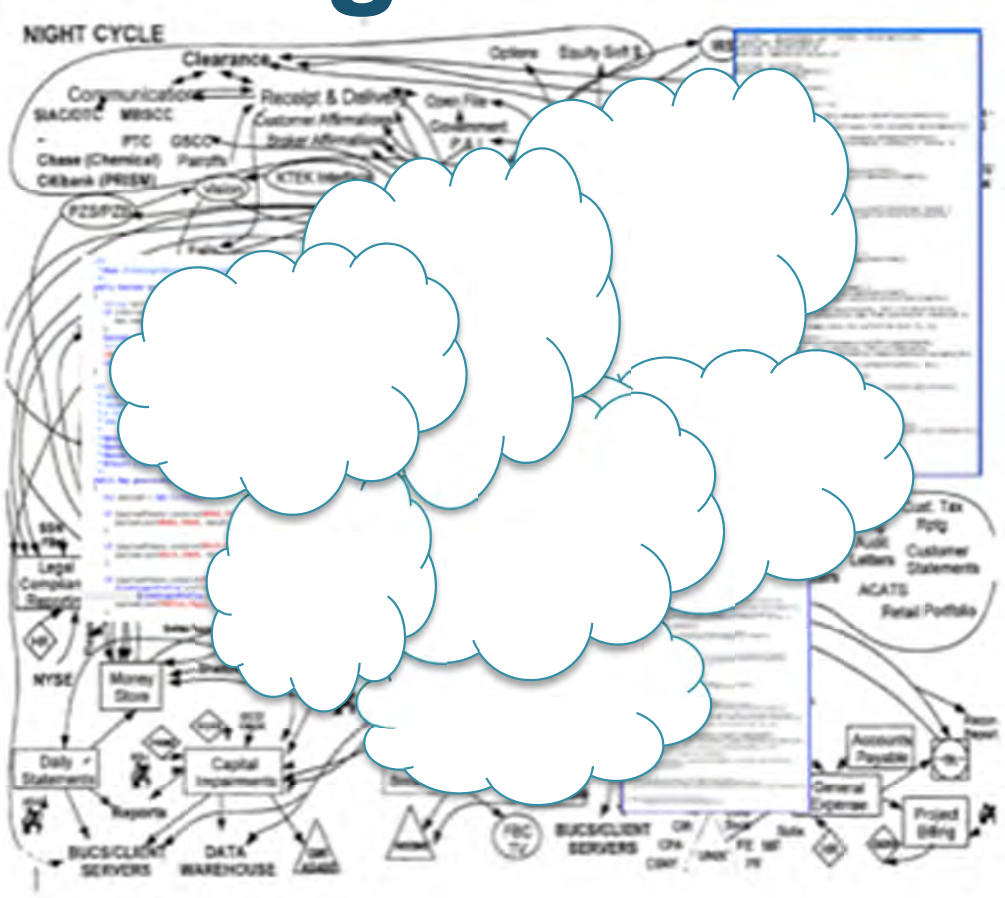
Three Teams Doing Work





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
  - Cross-functional “feature” teams
  - Teams on the same cadence

Scaled Agile Development is not …

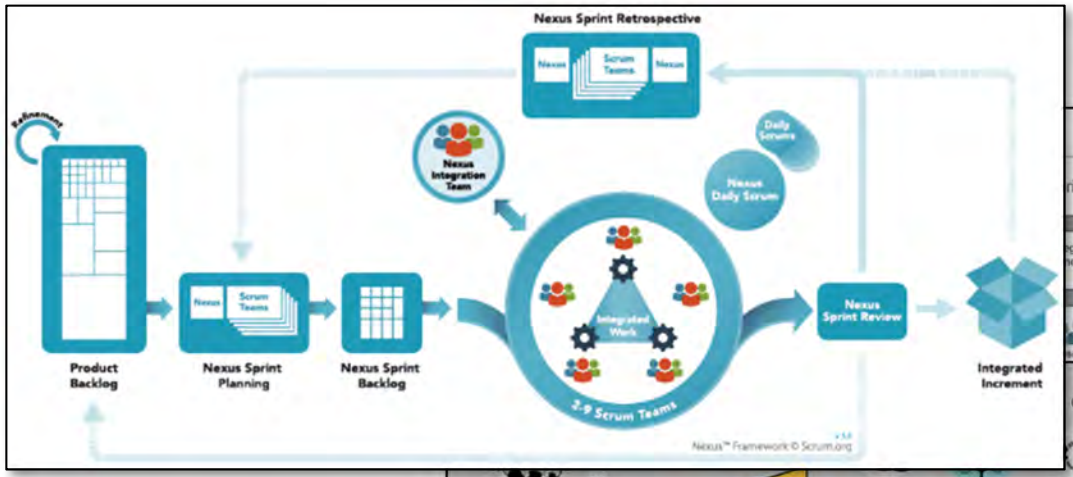
- Organizational transformation
  - "My organization wants to be more agile"
- Spinning up more agile teams
  - "IT is using Scrum so we should try it in our group"
- Adding more people
  - "That team is always behind, add more resources"

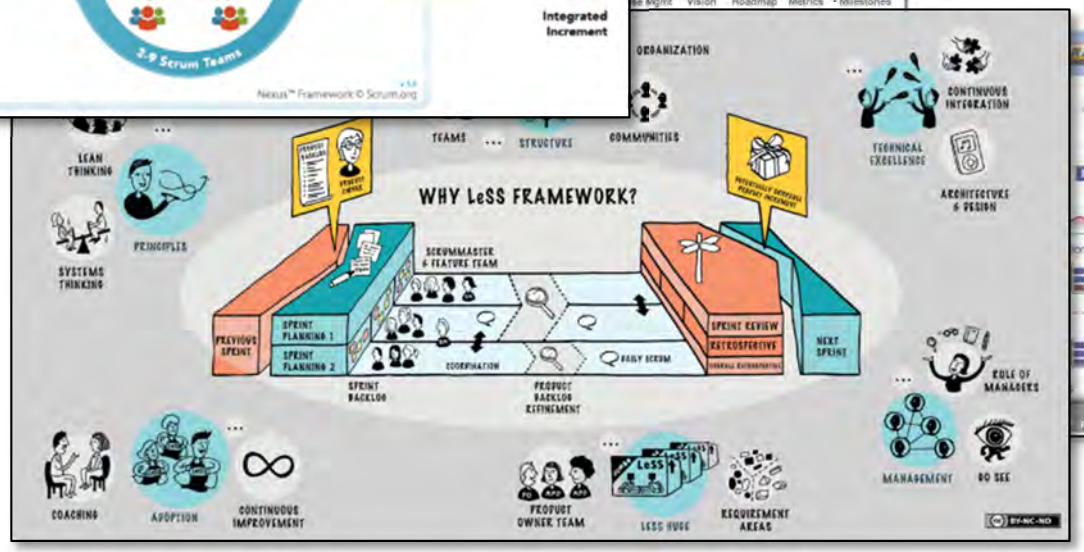
Ironically …

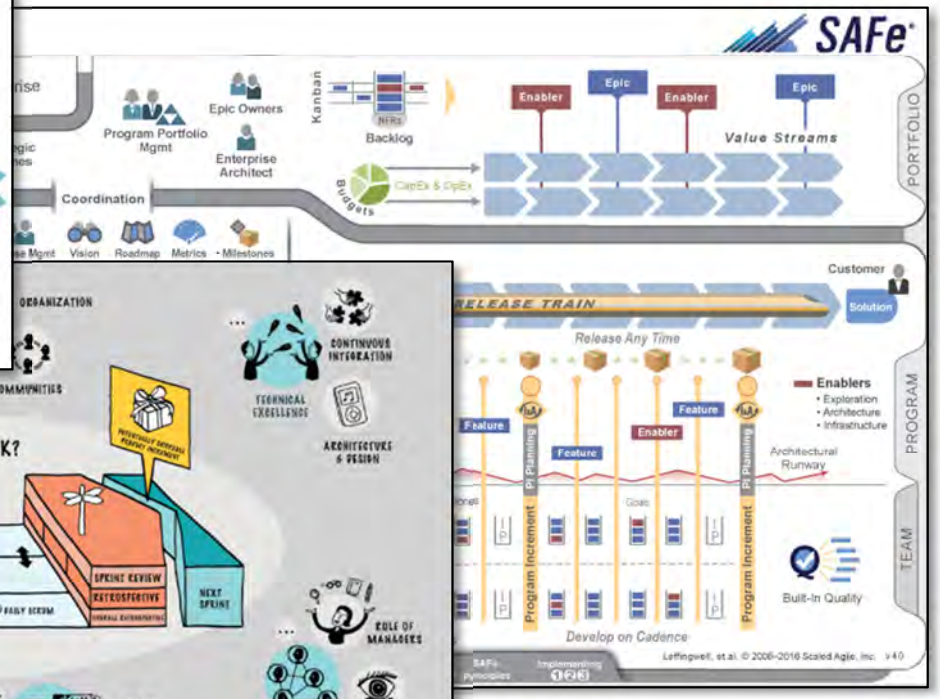
- It might be that you can do more with less
  - Removing teams
  - Removing people
  - Removing management
- Get a good Scrum Master or coach!

Next, do you have a *framework*? (or at least some proven experiments)

Nexus, LeSS, SAFe, SoS, RYO

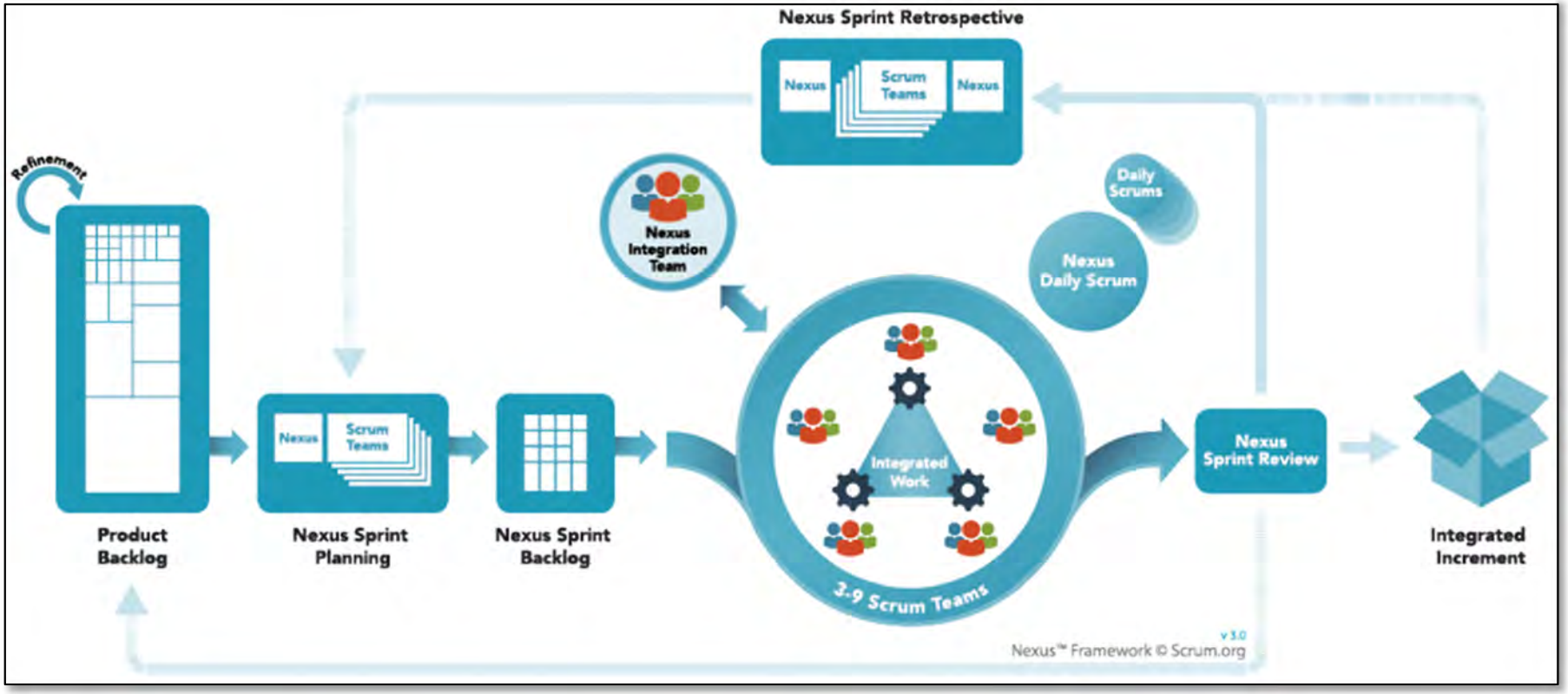






Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Scaled Professional Scrum (The “Nexus”)

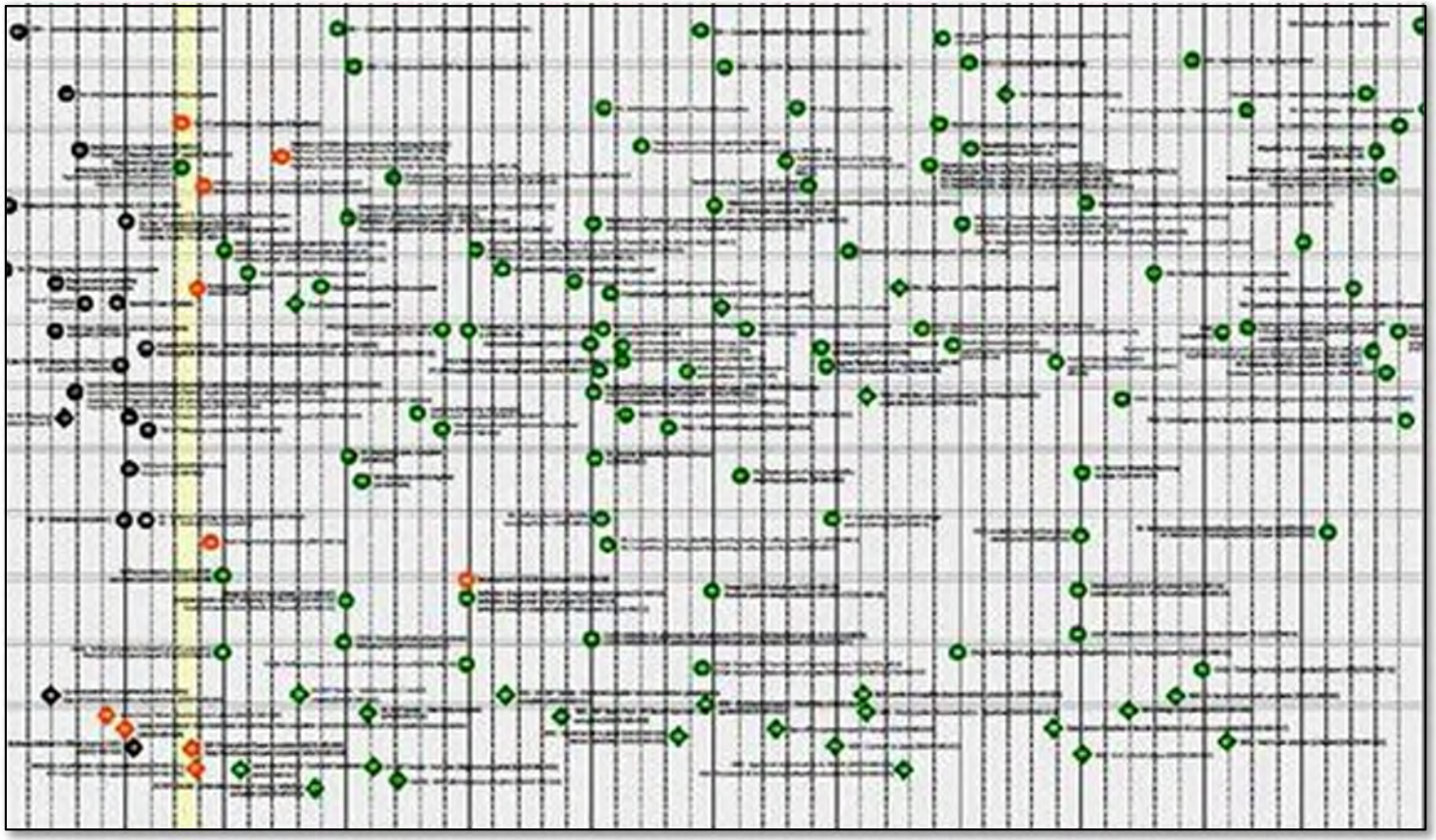


All Process Templates Support Scaling

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

Do you have a *product*?

Or just a bunch of projects …



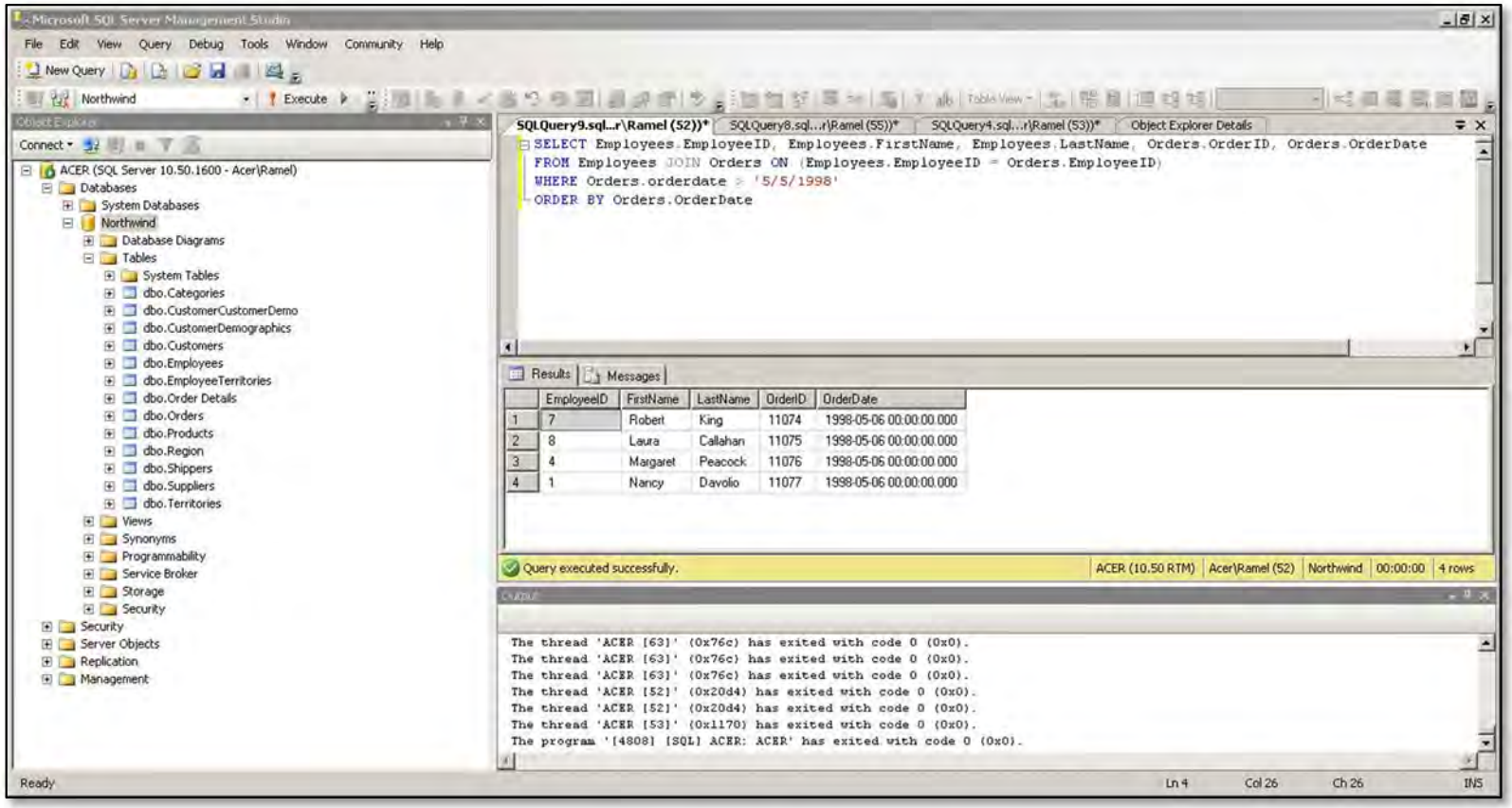
Product != Project

- It can be hard to identify the product behind all of your projects (and Visual Studio projects)
- Try asking …
  - Your customers, users, stakeholders
  - Your codebase
  - Your UI

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. WOPR)

Example: Where's the Product?



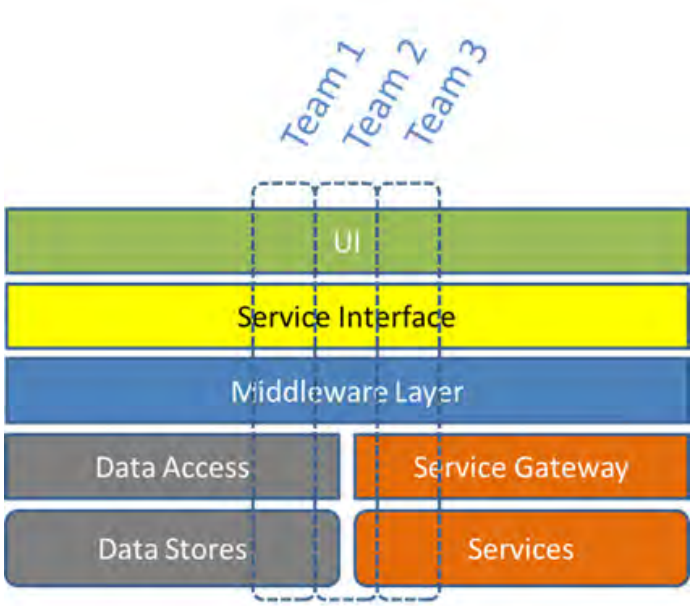
Product Areas

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

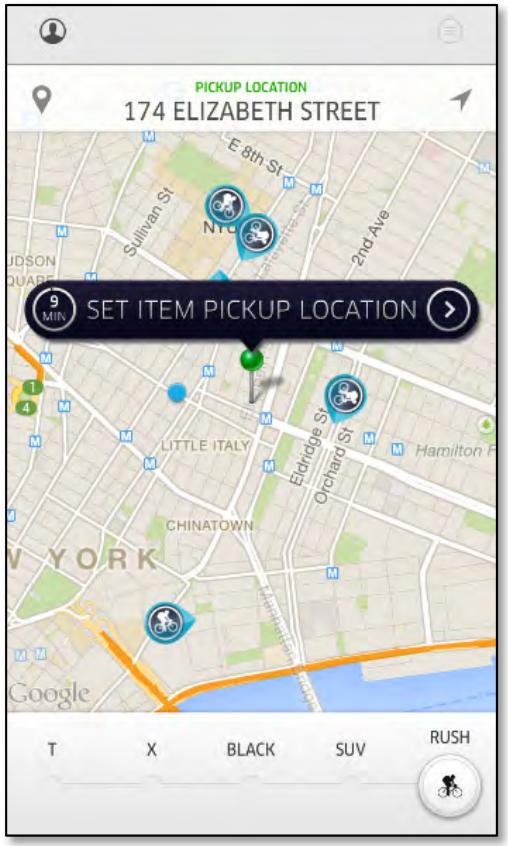
Do you have *teams*?

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



Consider Persona Teams



Multiple Teams in VSTS/TFS

- Create the teams
- Create a respective area path for each team
  - On-prem TFS can use a custom Team field and a global list instead http://bit.ly/20XzOCJ
- Add team members

http://bit.ly/1Q55dAh

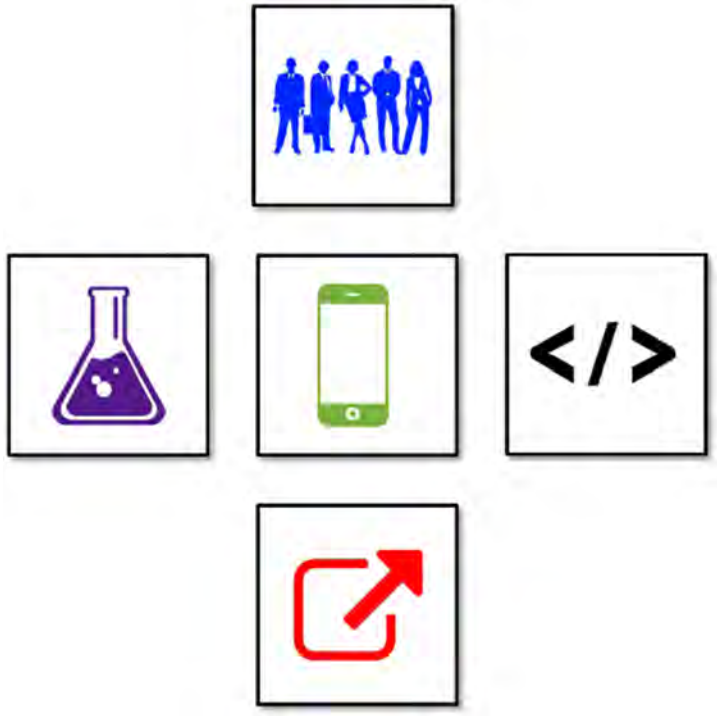
Are you *refining* the backlog?

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();

(thank you)

richard@accentient.com | @rhundhausen