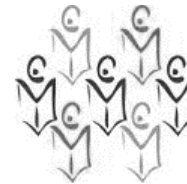


SCALING PRODUCT MANAGEMENT & INNOVATION

20170316 / Nadine Yap

My early background



Startup Stages



Agile & Scrum - Briefly

Mike Cottmeyer

Agile Program and Portfolio Management

<https://www.slideshare.net/mcottmeyer/agile-program-and-portfolio-management>

,

3/28/201

7

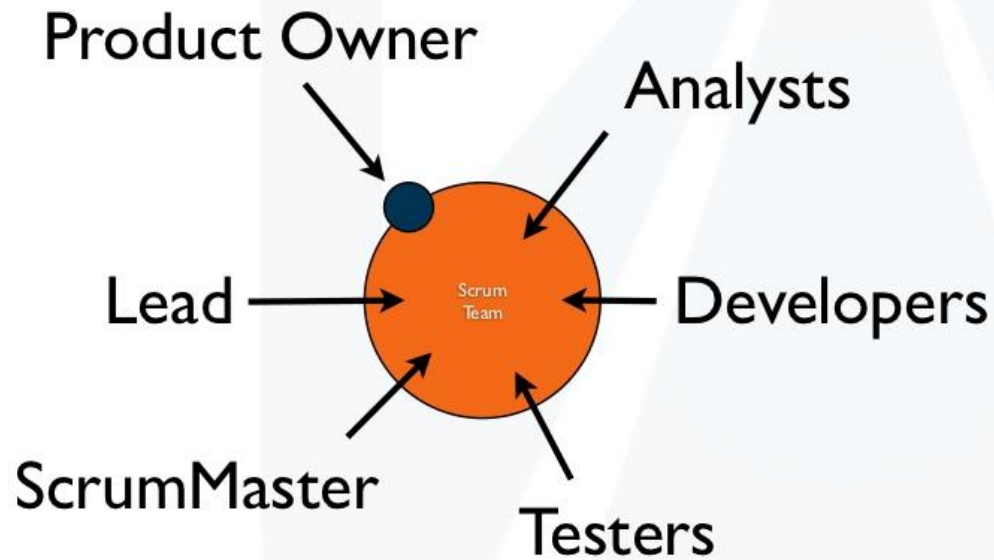
5

Agile and Lean Software Development

(Wikipedia)

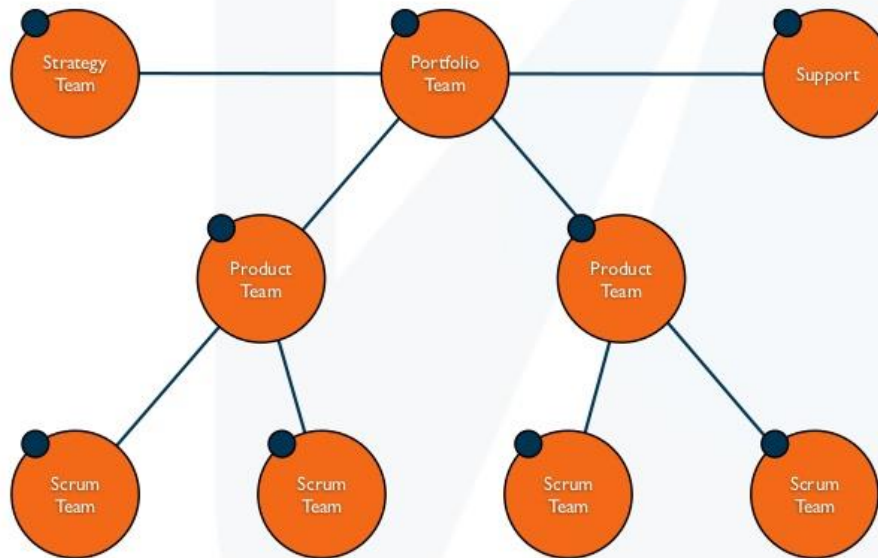
- ▶ **Agile software development** refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
- ▶ **Lean software development (LSD)** is a translation of [lean manufacturing](#) and [lean IT](#) principles and practices to the [software development](#) domain. Adapted from the [Toyota Production System](#),^[1] a pro-lean subculture is emerging from within the [Agile](#) community.
 - ▶ [Eliminate waste](#)
 - ▶ [Amplify learning](#)
 - ▶ [Decide as late as possible](#)
 - ▶ [Deliver as fast as possible](#)
 - ▶ [Empower the team](#)
 - ▶ [Build integrity in](#)
 - ▶ [See the whole](#)

Agile - Scrum Teams



Agile: Multiple Scrums

Enterprise Portfolio Management



Agile - Units of Definition

Epic

Epics collections of features, typically 1–3 months in duration. Epics span releases. Epics can span more than one team. These are the things senior leadership cares about.

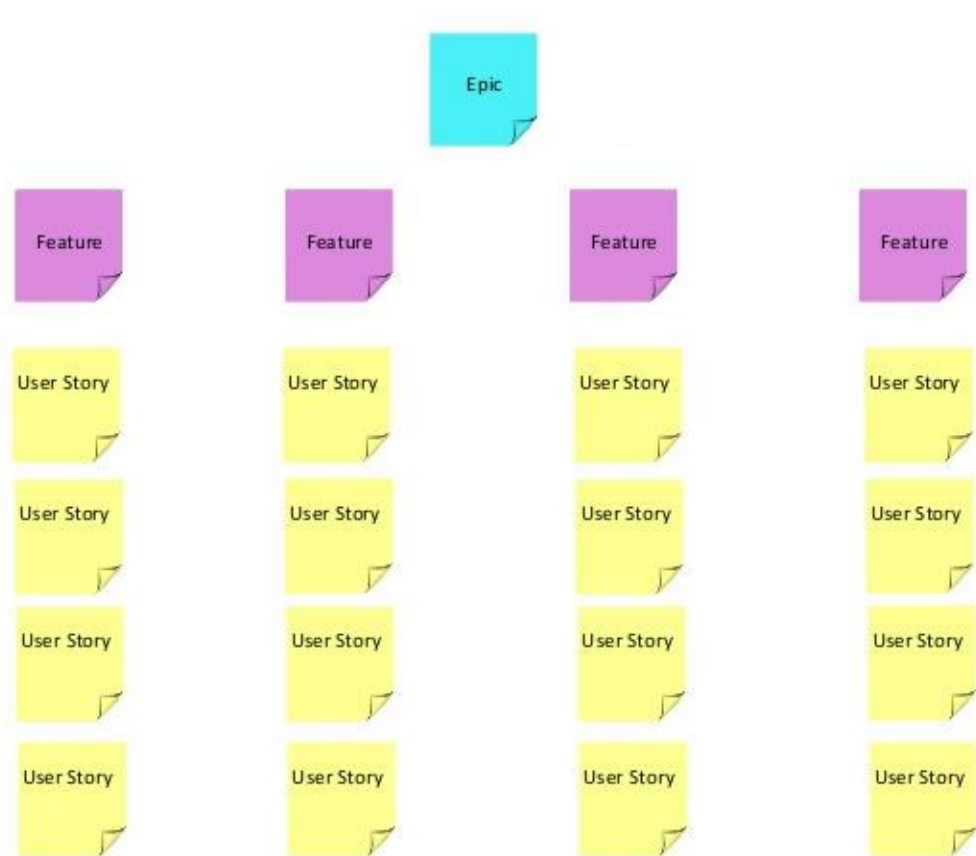
Feature

Features are smaller than epics, typically 2–4 weeks in duration. Features are contained within releases. Features are contained within a team. These are what the Product Owner Cares about.

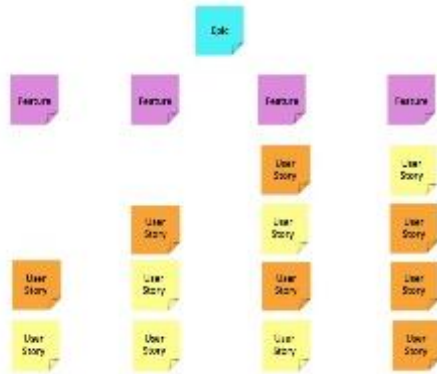
User Story

User Stories are the smallest increment of value, typically less than a week. User Stories are contained within sprint. These are the things Engineering Management Cares about.

Agile: Story Maps



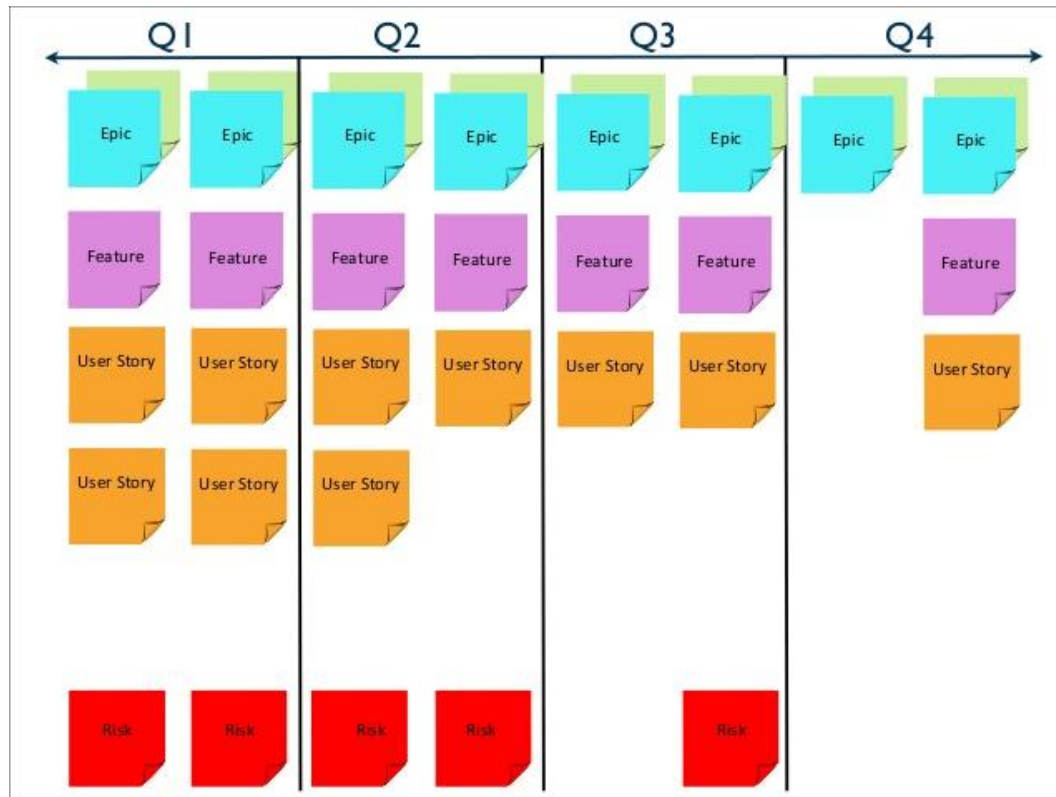
Agile: Sprinting



Planned Team Velocity = 6 points
Planned Estimated Hours = 98 hours

Story Backlog	Task Backlog	In Process	Task Done	Story Done
<div>User Story 1</div> <div>User Story 2</div>	<div>Task 16</div> <div>Task 4</div> <div>Task 4</div>	<div>Task 4</div> <div>Task 8</div> <div>Task 8</div>	<div>Task 8</div> <div>Task 16</div> <div>Task 2</div> <div>Task 16</div>	<div>User Story 3</div>

Agile: Roadmapping



Wednesday, May 8, 13

Product Management

Formula for a Winning Product

- Meets customer's needs
- Is better than other alternatives
- Is easy to use
- Has a good value / price

What does a PM do

- Expert on the market and the customer
 - Products and Services Vision
 - Strategy
 - Themes
- Identify, plan & prioritize product ideas to maximize ROI on engineering resources
 - Roadmaps
- Clearinghouse for all product ideas
- Translate business objectives and customer needs into product requirements
 - Stories / Requirements
- Work with the team to design and build great product
 - Marketing Deliverables
- Define and track key metrics
 - KPIs



Early Stage

3/28/201

7

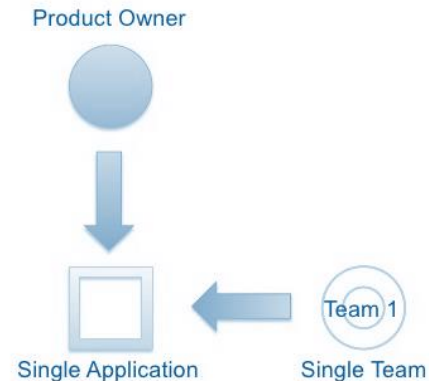
16

Developing early stage products

Way more uncertainty about

- Who are target customers really are
- Which customer needs you should address
- How best to meet those needs
- What design works best

A Simple Product



ThinQube Network, Inc.

Early Stage Product Management

- Truly understand your customer needs
- Ruthlessly prioritize and launch v1
- Talk with users 1 on 1 and get feedback
- Define equation of your business
- Identify and track key metrics
- Identify opportunities & prioritize by ROI
- Launch, **test**, learn, and **iterate**

Testing and Metrics

- ▶ Always have a hypothesis and a measure of success
- ▶ Formats
 - ▶ Prototypes
 - ▶ Focus Groups
 - ▶ A/B testing
- ▶ Tools
 - ▶ Instapage
 - ▶ Optimizely
- ▶ Business Intelligence Tools
 - ▶ Dashboards
 - ▶ Google Analytics



exchange.com

Timeline

- December - Brainstorm
- January - Code
- February - Test / Load
- March - Launch
- April - Get Acquired

Why?

- Knew the market
- KISS product
- JIT Technology
- Clear Exit Strategy
- Laser Like Focus
- Killer Timing



VideoGuide

- ▶ - Phase One - Build
 - ▶ Design Product
 - ▶ Targeted Public Beta
 - ▶ Built multiple products (Guide, news, sports)
- ▶ - Phase Two - Launch
 - ▶ Expensive Nationwide Marketing Launch
 - ▶ Into Mass Market Segment
 - ▶ Discovered technology problems at scale
 - ▶ Not enough runway to pivot product
- ▶ - Phase 3: Pivoted business strategy
 - ▶ Trimmed costs and staff
 - ▶ Ensured patents filed
 - ▶ Acquired by a competitor with a worse UX, and a less experienced engineering team
 - ▶ But a better business development strategy which also allowed it to make the better technology choice



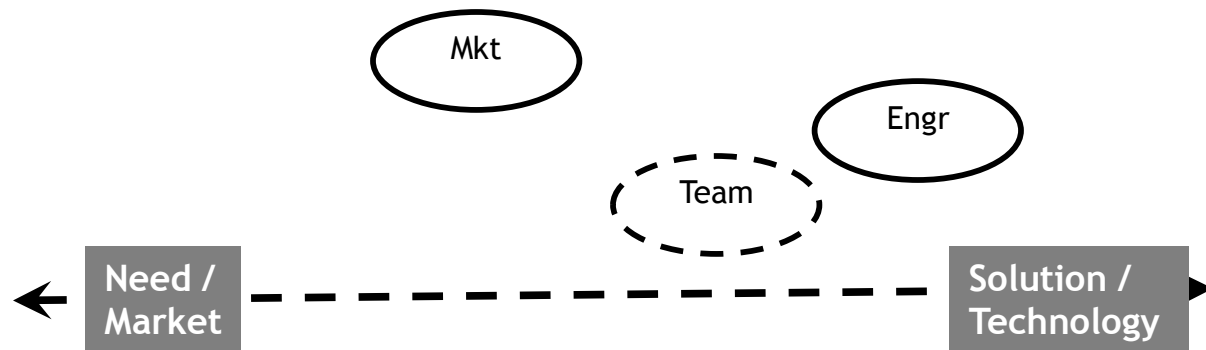
Scaling

3/28/201

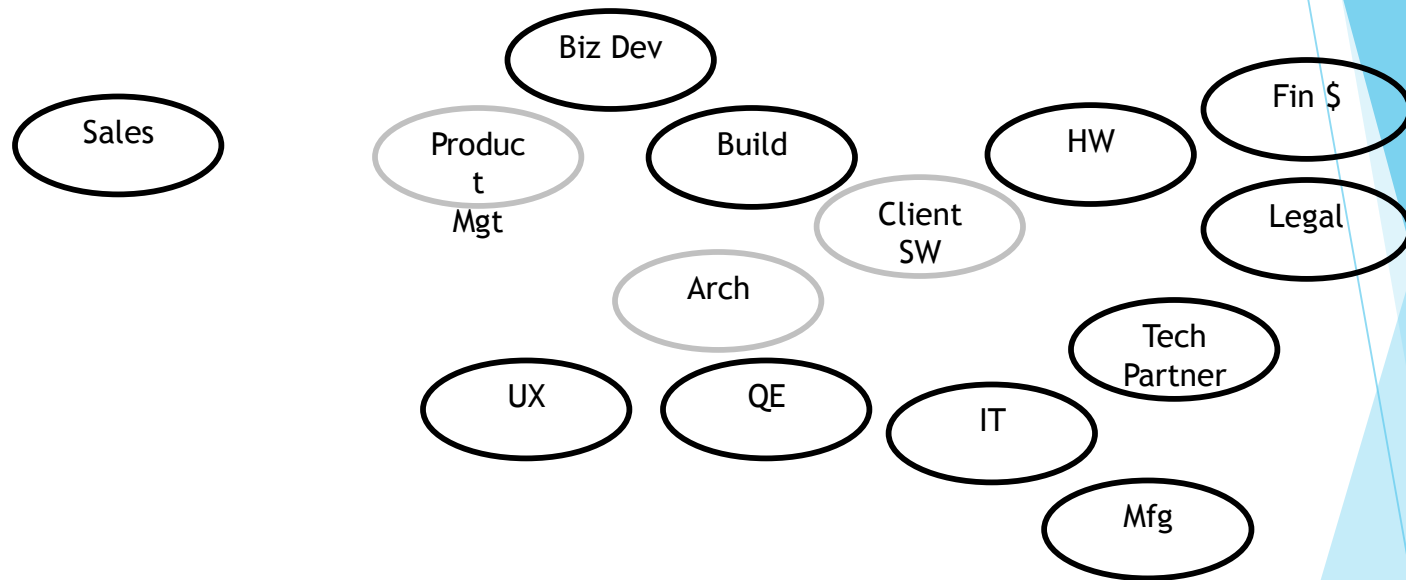
7

22

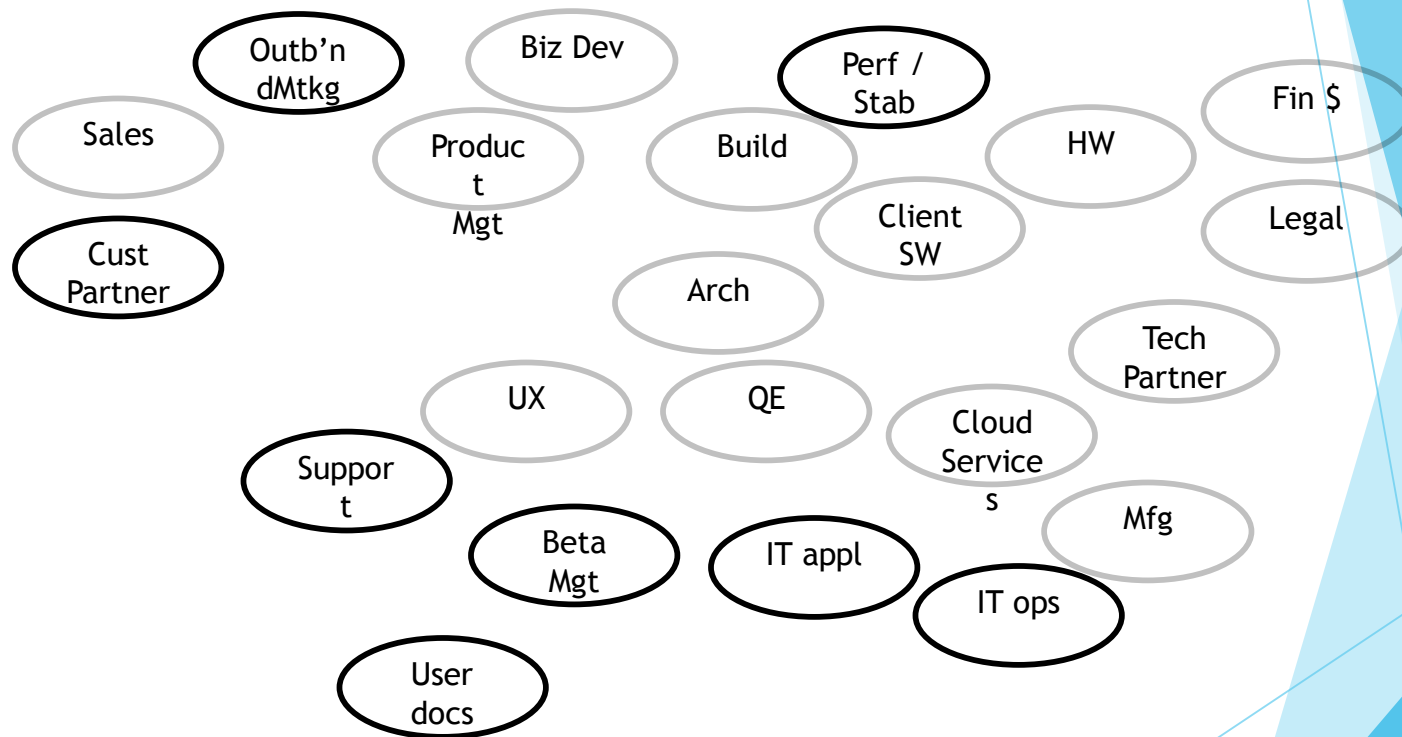
Tech Business: Early / Small



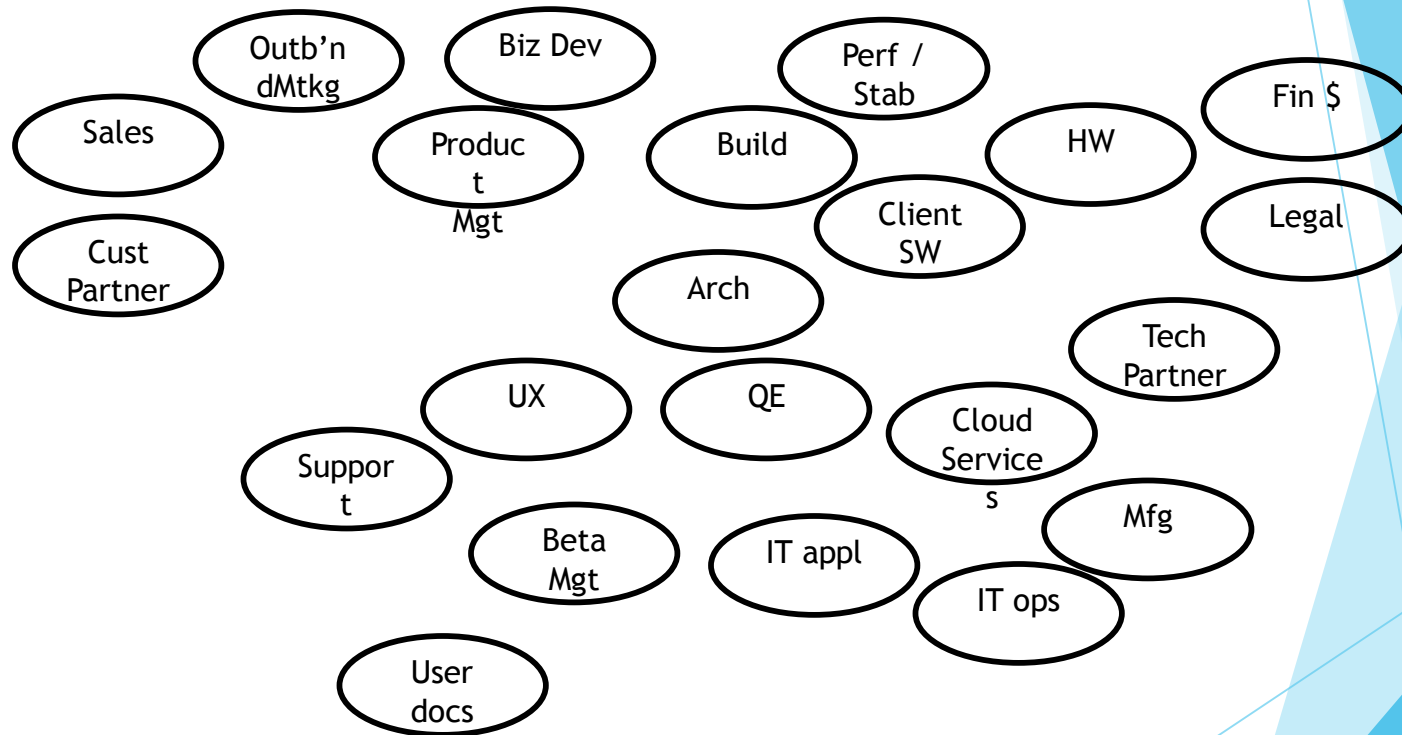
Medium



Larger / Later

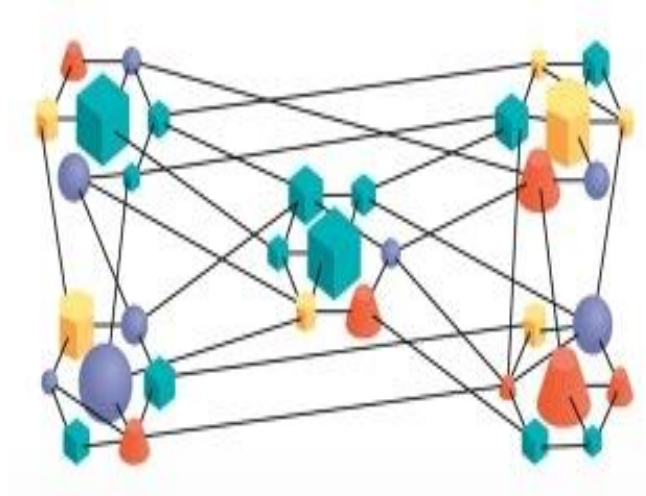


Coordinating This = Complex



So, what happens when you scale

- ▶ More users (and types of users)
- ▶ More products, platforms
- ▶ More employees / teams
- ▶ Competing agendas
- ▶ Dependencies





- ▶ Phase One - Video Reseller
- ▶ Phase Two - Pivoted to WebRTC Technology
 - ▶ Attempted SaaS
 - ▶ Resolved to productize PaaS instead
 - ▶ Launched Platform in Beta
- ▶ Phase Three - Monetize
 - ▶ Scale Product (additional Platforms)
 - ▶ Scale Team
 - ▶ Scale Customer Support
 - ▶ Scale Analytics
 - ▶ Improve Product Features

JS Team - Several products
C++ Team - 3 products
Mobile Team - 4 products
Support Team - The support Portal



\$200K Seed Money



Phase one

- Meet User need
- Organic, Simple

\$4MM Series A



Phase two

- Meet Content Creator's Needs
- Rewrite Platform in Rails
- Augment with Pivotal
- Adopt Agile
- Monetize through Ads

- Platform Team
- Web Team
- Mobile Team
- Analytics Team

\$20MM Series



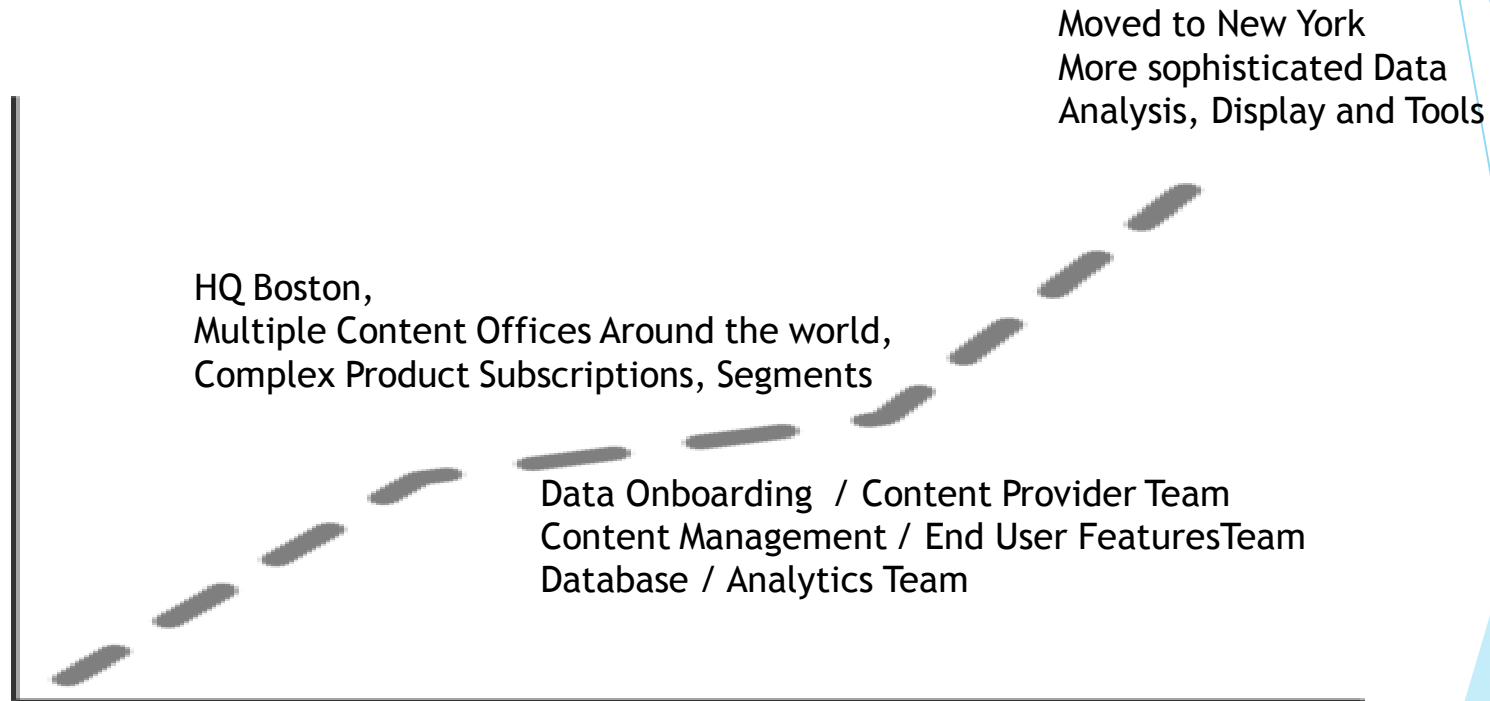
Phase three

- Meet Investor's needs
- Grow In-House Team
- multiple products and segments
- Scale Customer support, Analytics, etc



ISI EMERGING MARKETS

A EUROMONEY INSTITUTIONAL INVESTOR COMPANY



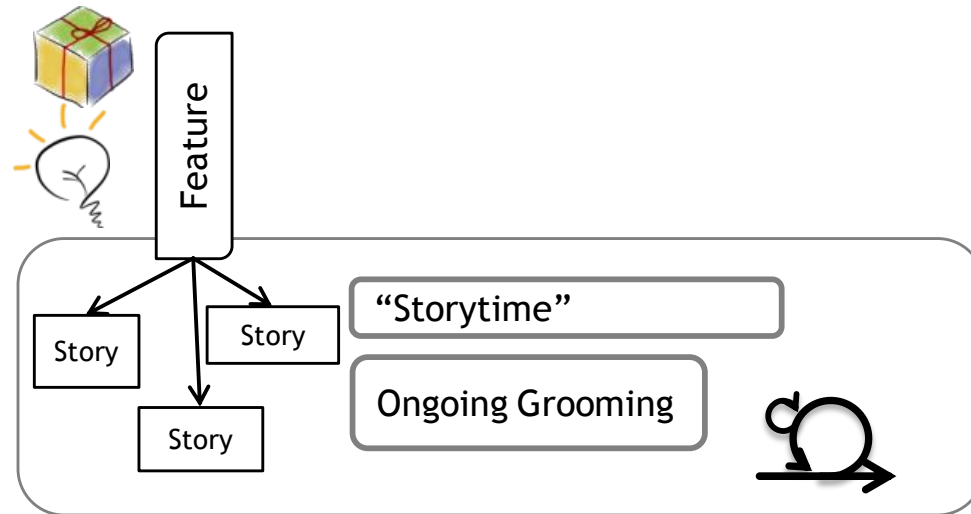
“One obvious yet under-appreciated law of business physics is:

The larger the company becomes, the more opportunities emerge to screw it up”

Ben Horowitz, A16Z

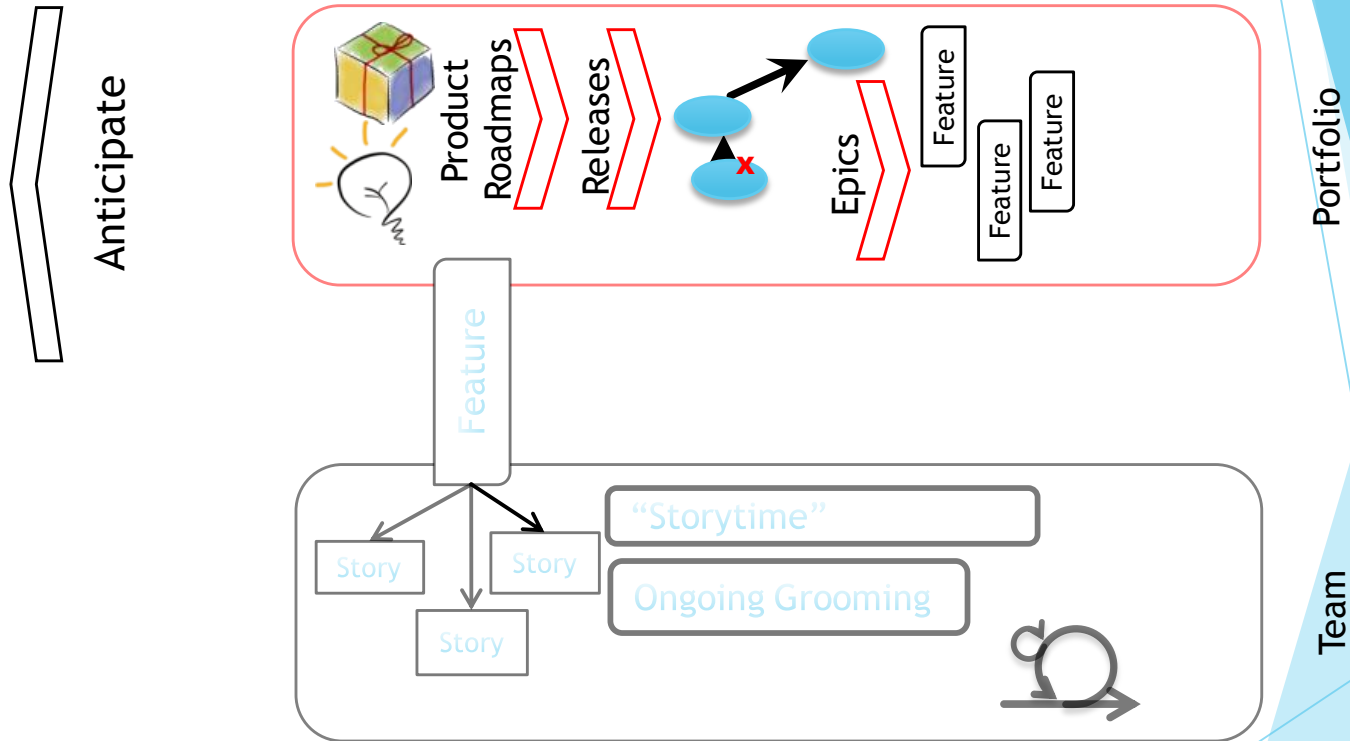
Initially : WORK **IN** THE BUSINESS

Operate

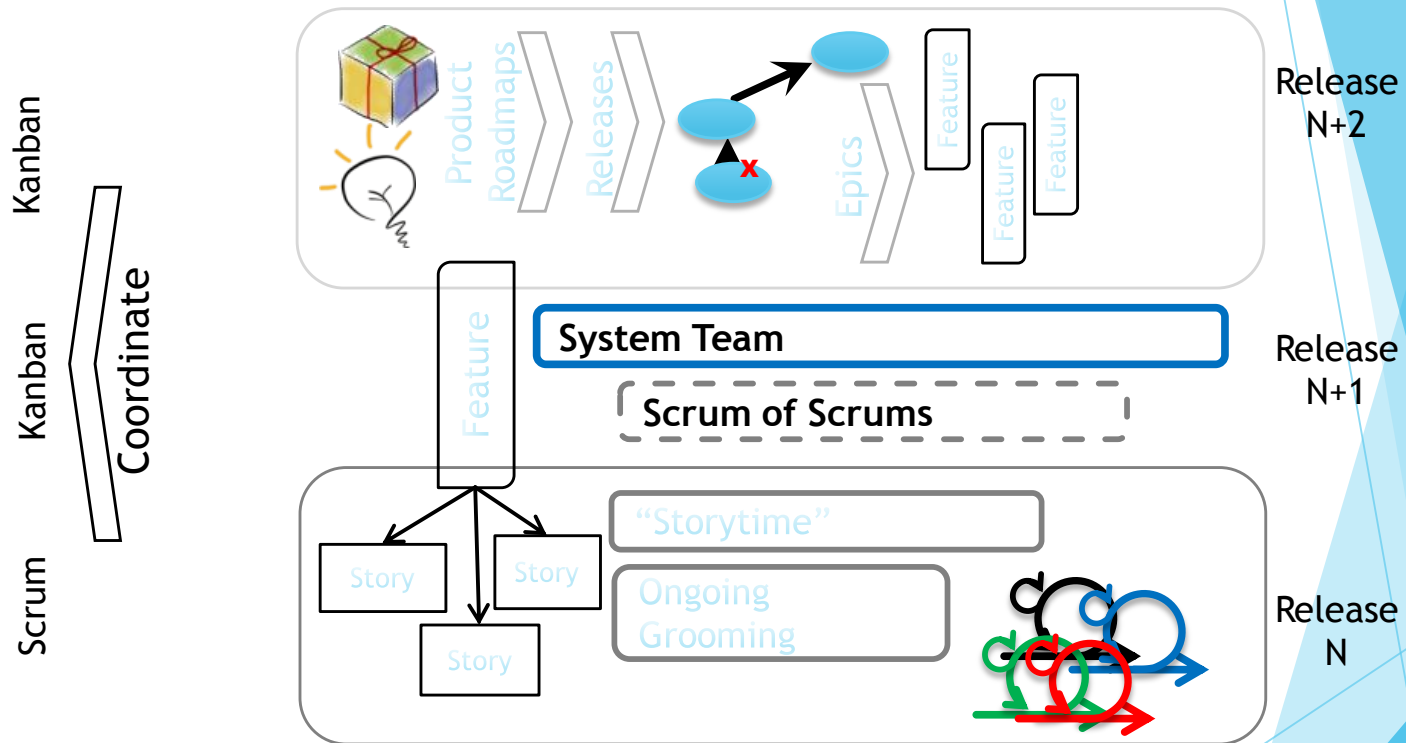


Team

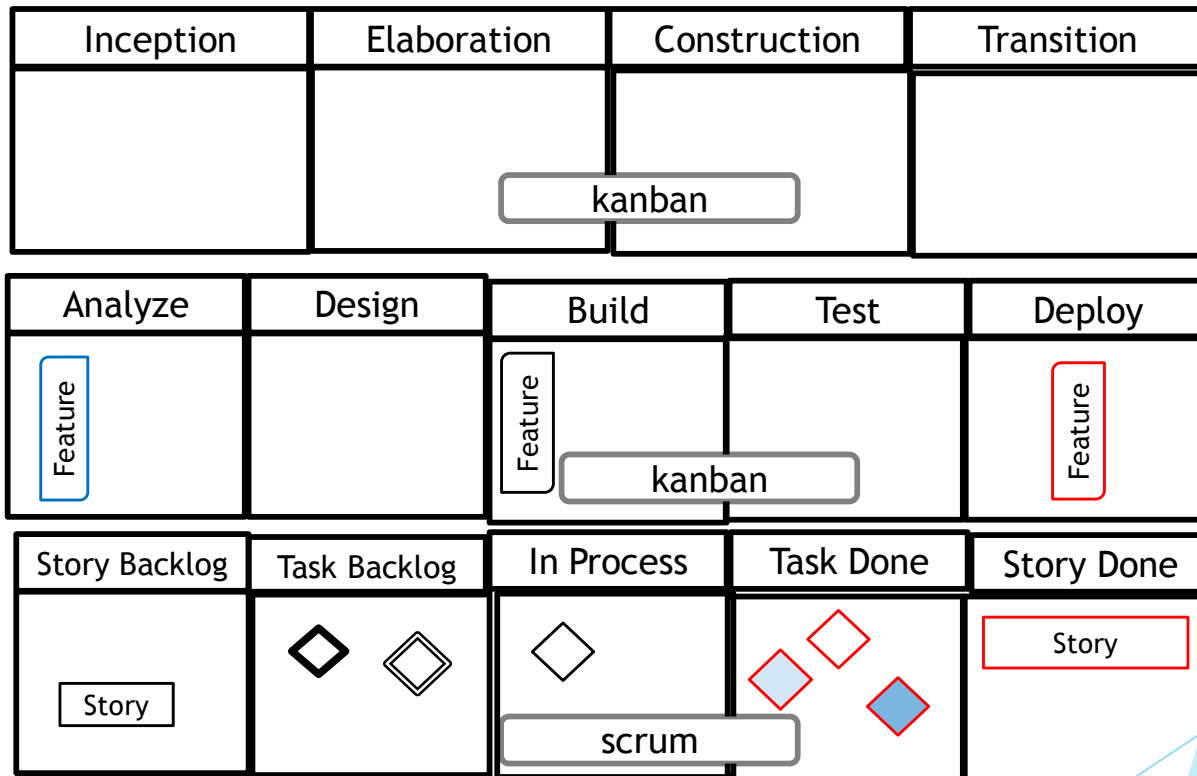
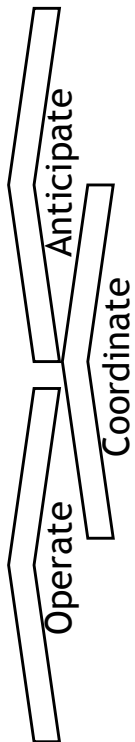
Then: **WORK ON THE BUSINESS**



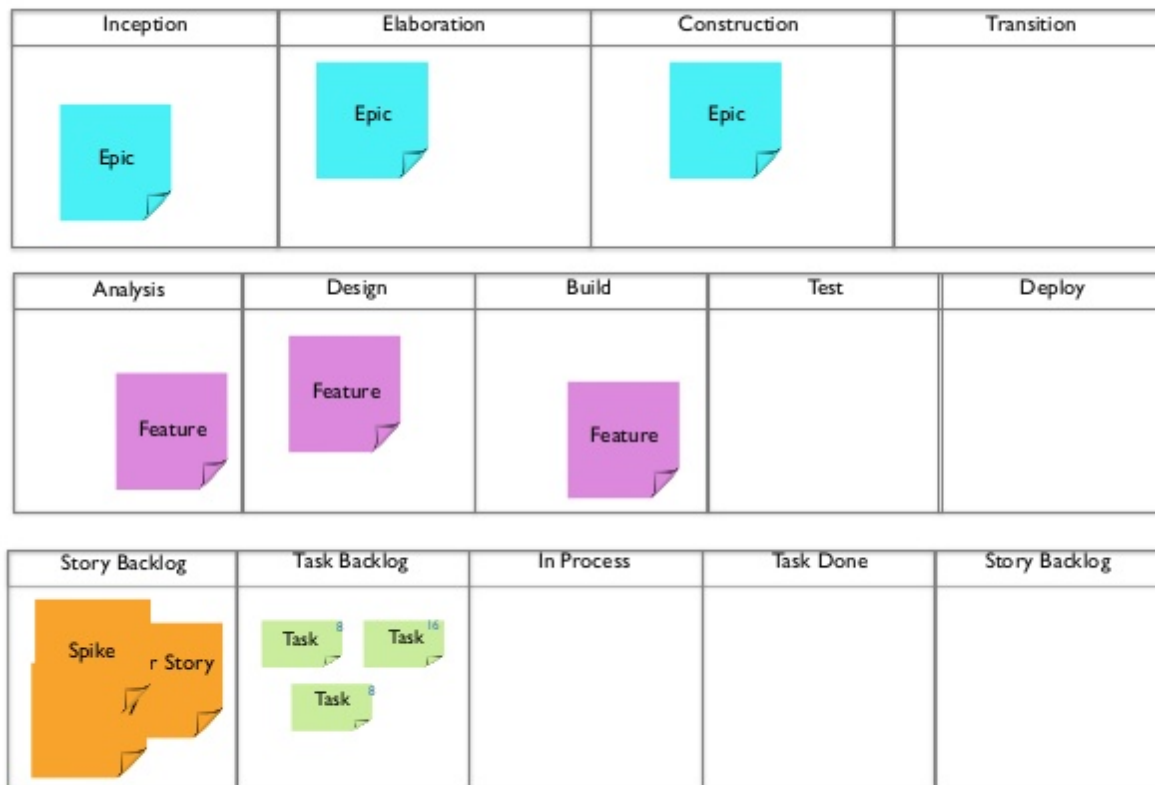
Eventually: **Steer** the system



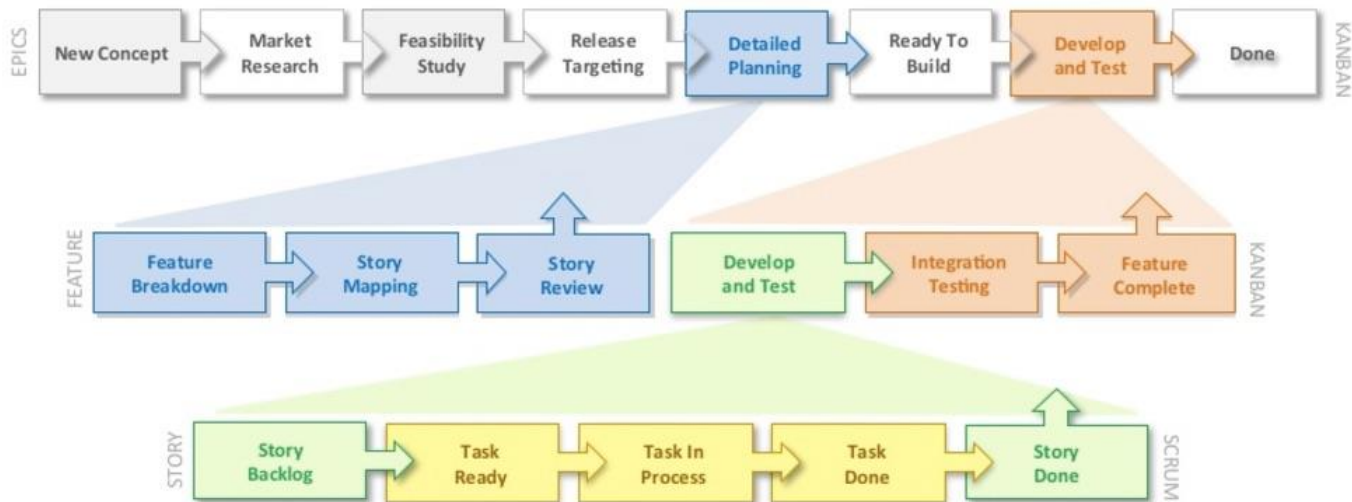
At enterprise scale



In terms of “Agile”



Development Flow



What needs to scale?

- Decision Making / Empowerment
- Process (but not too much)
- Communication
- Roles and Team Structures

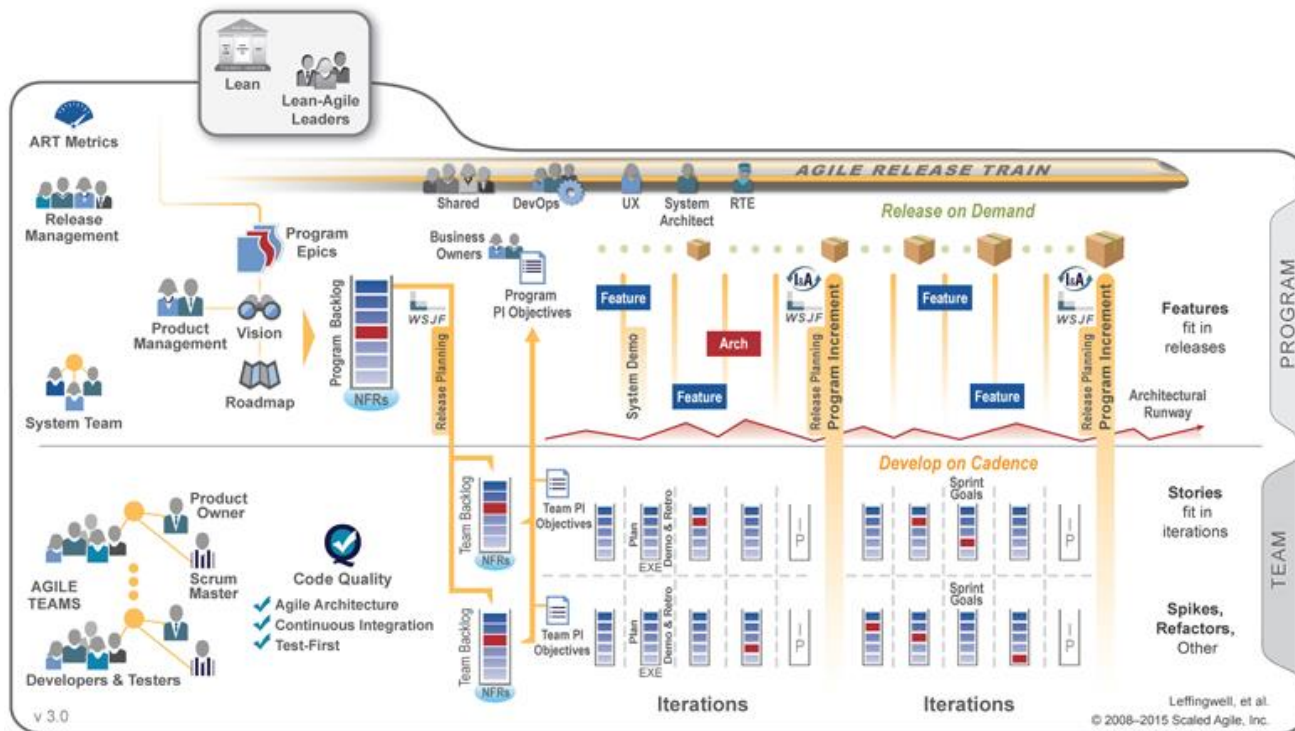
Process: Decision making / Empowerment

- Create Stable, long lasting, cross functional decision making teams with authority in their domains
- Form cross-functional core teams to drive effective decision making down the org
- Regular Operating rhythm and accountability
- Aligned on Vision - Purpose
- Clear on Business Goals - Mission critical priorities
- Articulate Definition of “Done”; Marketing Deliverables



Process: Explicit Project Flow

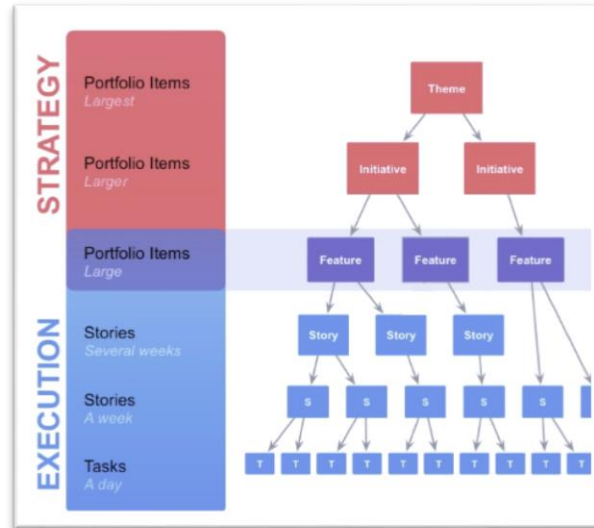
- ▶ Articulate a clear progression of steps and the transitions between them
- ▶ Ensure clarity about everyone's place in the flow
- ▶ Regular updates on what stage each project is at



Process: Planning

Move from a simple early stage 2-level Feature-Task/Story/Bug hierarchy -> to 3 levels.

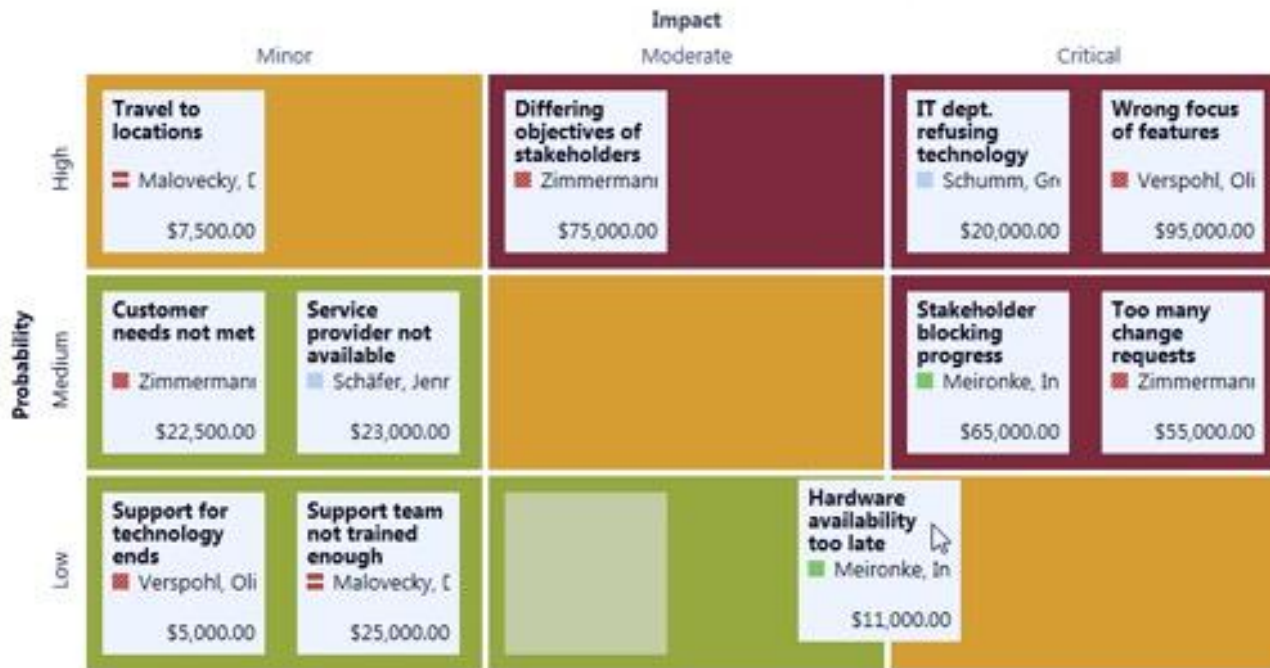
Track larger scale projects throughout the value stream (not just through the build portion).



		Release quarter			
Product name	Goal name		2015 Q4	2016 Q1	2016 Q2
Fredwin Cycling Software	#1 in social fitness cycling software		Enhance "Tour de California" coverage	Launch "Giro d'Italia"	
				Build Progress Tracking Dashboard	
	Triple revenue YoY			Launch "Giro d'Italia"	
	Top rated social fitness cycling apps				Mobile app upgrades
	Largest partner ecosystem		Partner portal		
	International Expansion			Launch "Giro d'Italia"	

Process: Risk Management

- ▶ Scope is no longer the biggest risk
- ▶ Dependency management and other risks Now
- ▶ Manage risks explicitly



Description

Communication: Collaboration

Meetings don't scale well With 2-3 teams - you can just get your 4-6 PMs and Eng Mgrs in a room and discuss. This doesn't work when you have 50 of them.

- Eliminate useless meetings
- Use supporting tools to organize meetings (agendas), record and disseminate information
- Distinguish types of meetings & manage them differently
 - Status Update Meetings
 - Information Sharing Meetings
 - Decision Making Meetings
 - Problem Solving Meetings
 - Innovation Meetings
 - Team Building Meetings



Communication: Knowledge Mgmt

When decisions are made, how are they communicated and documented so that we don't have the same conversations over and over?

How do we prevent information from being bottlenecked?

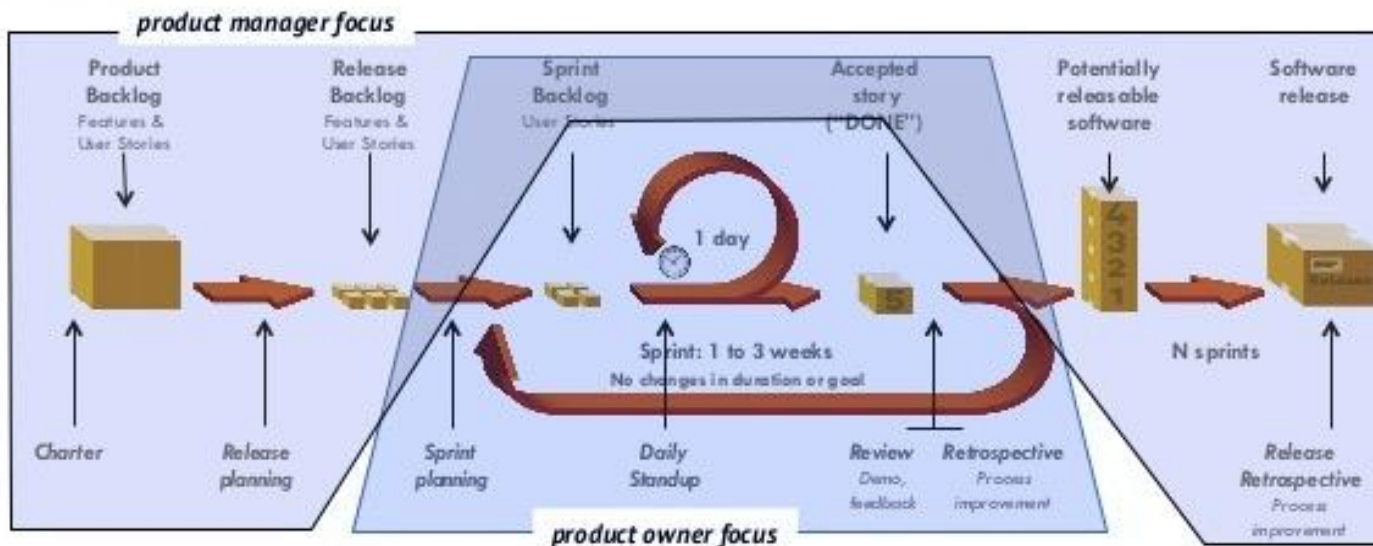
- Lots of tools - pick and tweak the ones that fit
- Implement content and communication workflows
- Differentiate between "ungroomed" and "groomed" parts
- Adopt a wikipedia-like "editor model" for the latter
- Don't duplicate, integrate
- Be consistent and disciplined



Roles / Team Structures

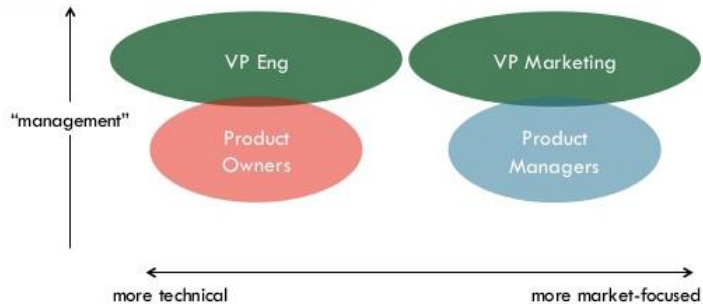
Product Mgr vs Product Owner

- Product Owner represents the customer in backlog prioritization & requirements questions.
- Provides intense sprint- level focus: stories, backlog, prioritization, acceptance
- One product owner per team, not per product
- Full time

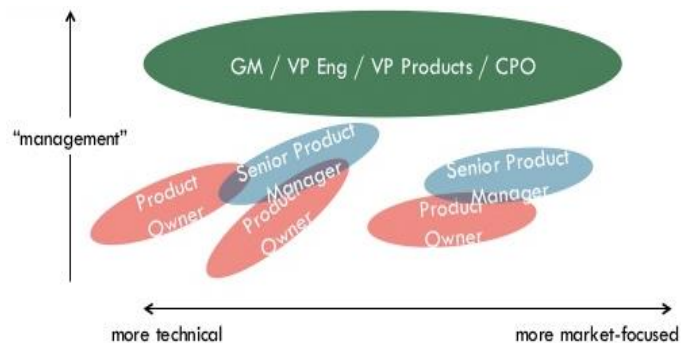
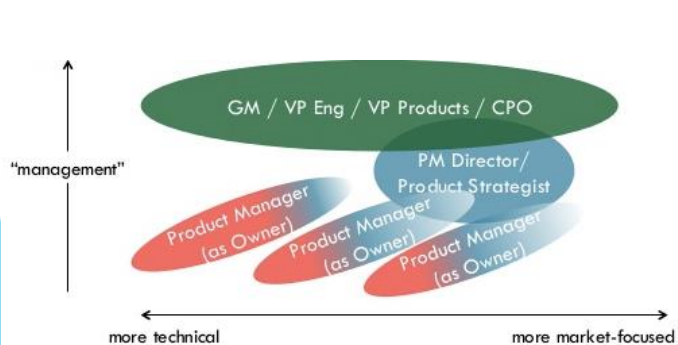


Roles and Team Structures

Dysfunctional / Silos

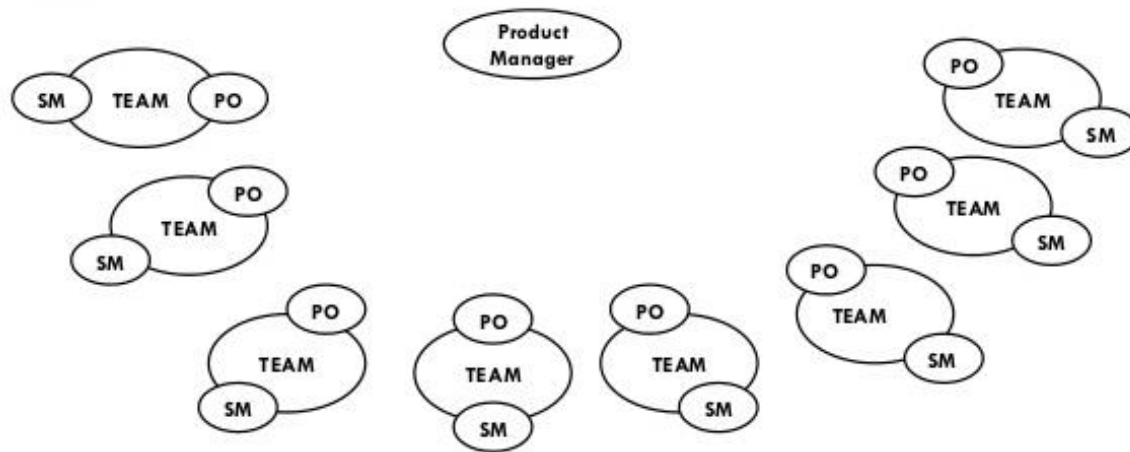


Other, Better Options - depending on your Products

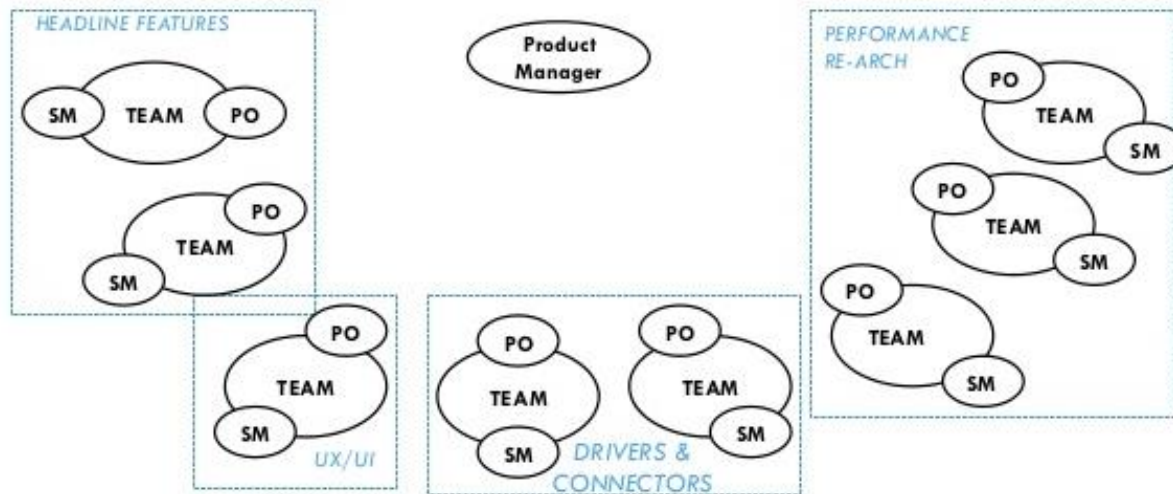


90 Person Project

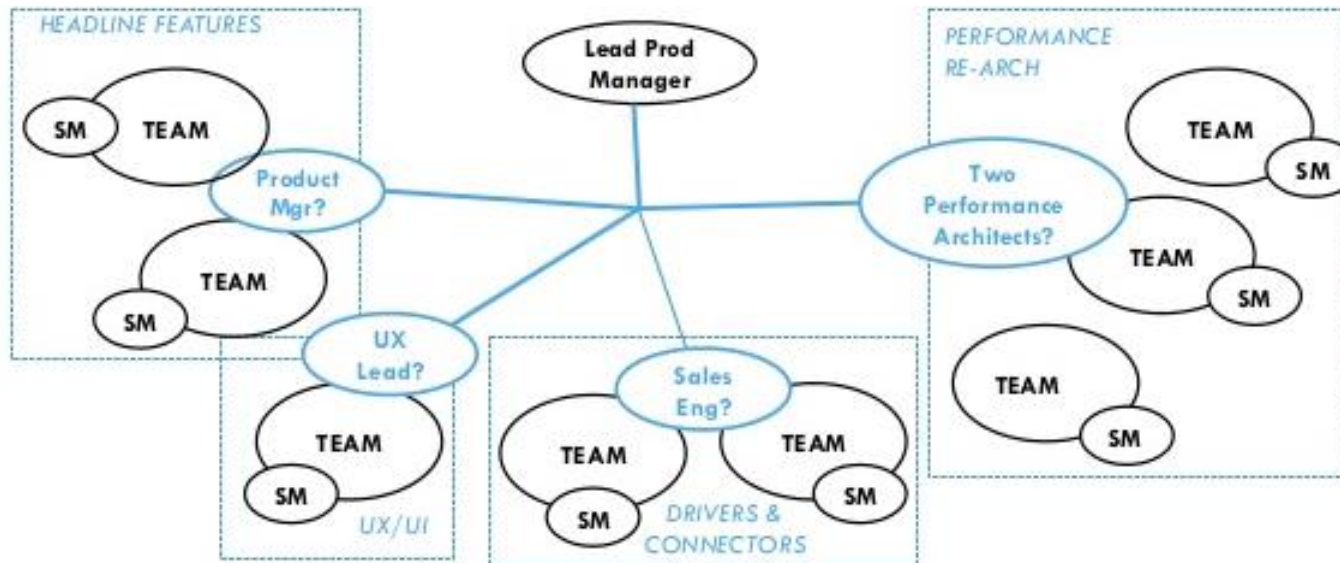
- 1 Product, 8 Teams



What does each team do?



Right Product Owners?





in short

Early Stage

- Truly understand your customer needs
- Define equation of your business
- Identify opportunities & prioritize by ROI
- Identify and track key metrics
- Be Agile - MVP - Use simple tools
- Launch, test, learn, and iterate

Scaling to Manage Complexity

- Decision Making / Empowerment
 - Clearly defined goals
 - Cross functional decision making teams
- Process (but not too much)
 - Explicit project flow
 - Multi Stage Planning
 - Manage through Value chain
 - Explicit Risk Management
- Communication
 - Manage meetings effectively
 - Use Knowledge Management tools consistently
- Roles and Team Structures
 - PM vs PO
 - Avoid dysfunctional silos
 - Assign the right person for the product

THANK YOU

Reference

Rich Mironov

Scaling up product manager/product owner organizations

<http://www.slideshare.net/RichMironov/agile-pm-popdx>

Itamar Goldminz

Lean Scaling: From Lean Startup to Lean Enterprise

<https://confengine.com/agiledc/proposal/417/-lean-scaling-from-lean-startup-to-lean-enterprise>

Nitin Karandikar

Software Startups: Scaling the Product Development team

<https://www.linkedin.com/pulse/software-startups-scaling-product-development-team-nitin-karandikar>