



SSW-555: Agile Methods for Software Development

Scaled Agile Development (SAFe)

Instructor: Prof. Zhongyuan Yu
School of Systems and Enterprises
Stevens Institute of Technology

Acknowledgements

Some of the material in these slides is from:

- "The Big Picture of Enterprise Agility" by Dean Leffingwell

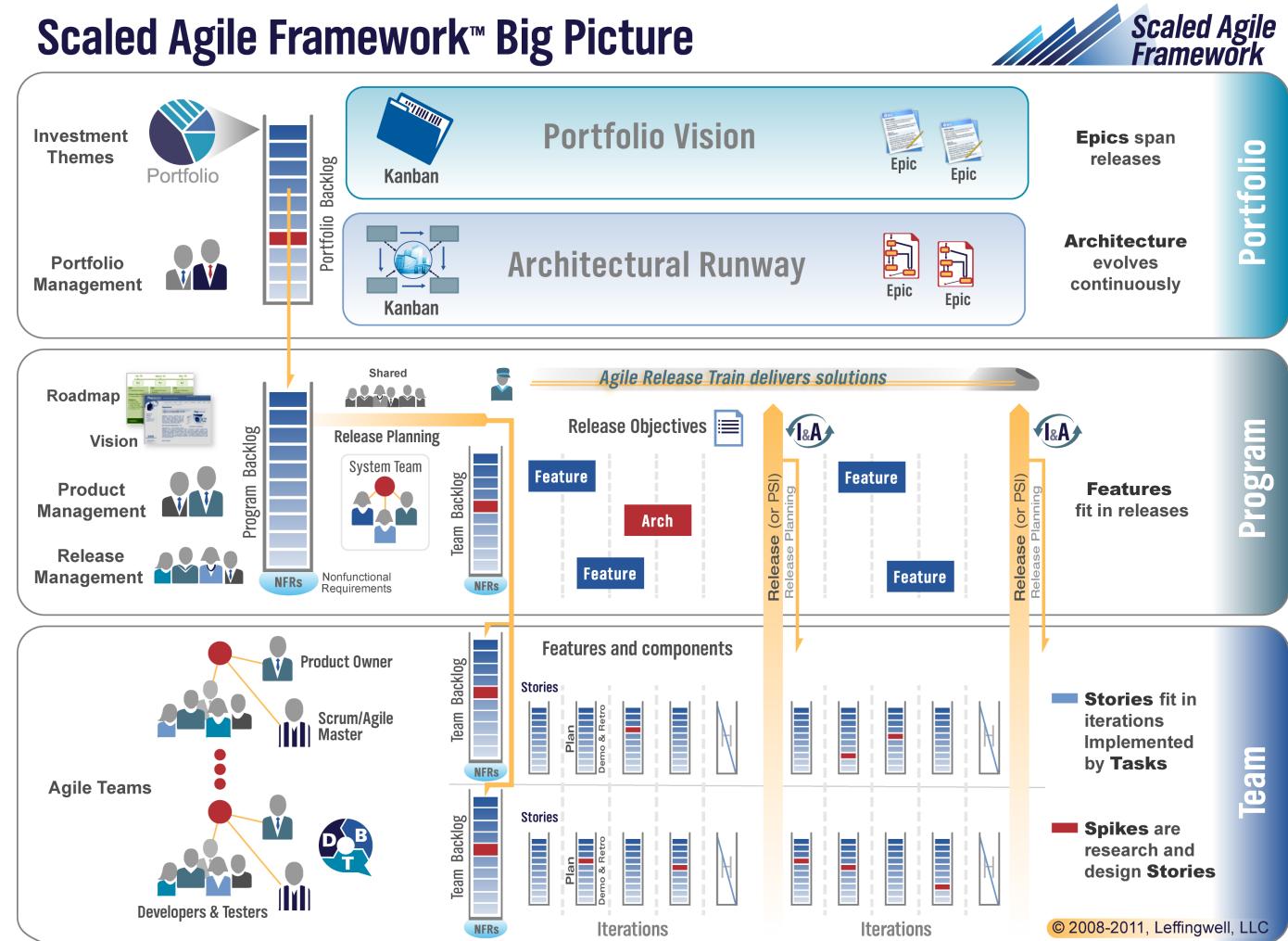
<https://scalingsoftwareagility.files.wordpress.com/2007/03/the-big-picture-of-enterprise-agilitywhitepaper.pdf>

- Scaling Software Agility Blog

Today's topics

Scaling Agile and Scrum to larger projects

- Overview of SAFe
- Project Level of SAFe
- Program Level of SAFe
- Portfolio Level of SAFe



See also www.scalingsoftwareagility.wordpress.com and Leffingwell, D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley (Pub. 2011)

v 0.8

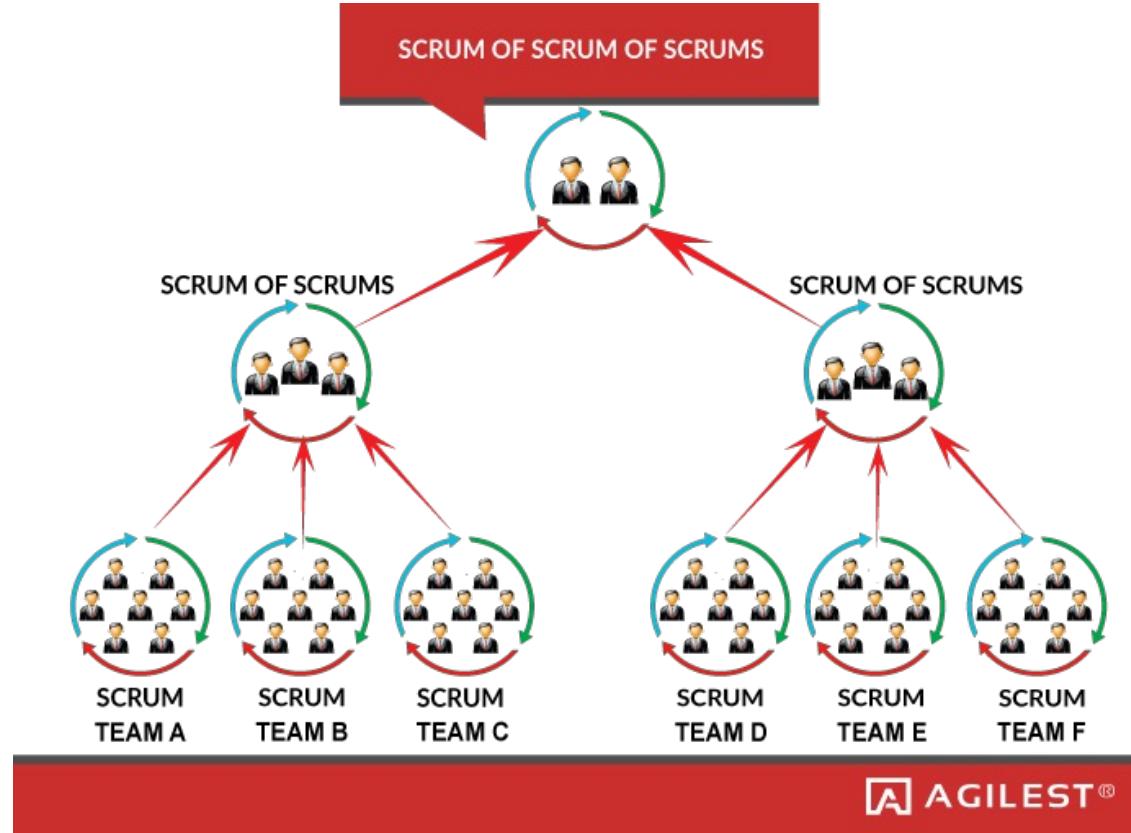
Limitations of Scrum for large projects

- Small teams
 - Self-directed teams - don't wait for instructions from the boss
 - Rely on face-to-face communication
- Only short-term planning
 - No Big Design Up Front (BDUF)
 - FDD recommends JEDI
 - Isolated projects
- Single Product Owner (Customer)
 - Answers developers' questions
 - Sets priorities and release dates



Scrum of Scrums: One solution

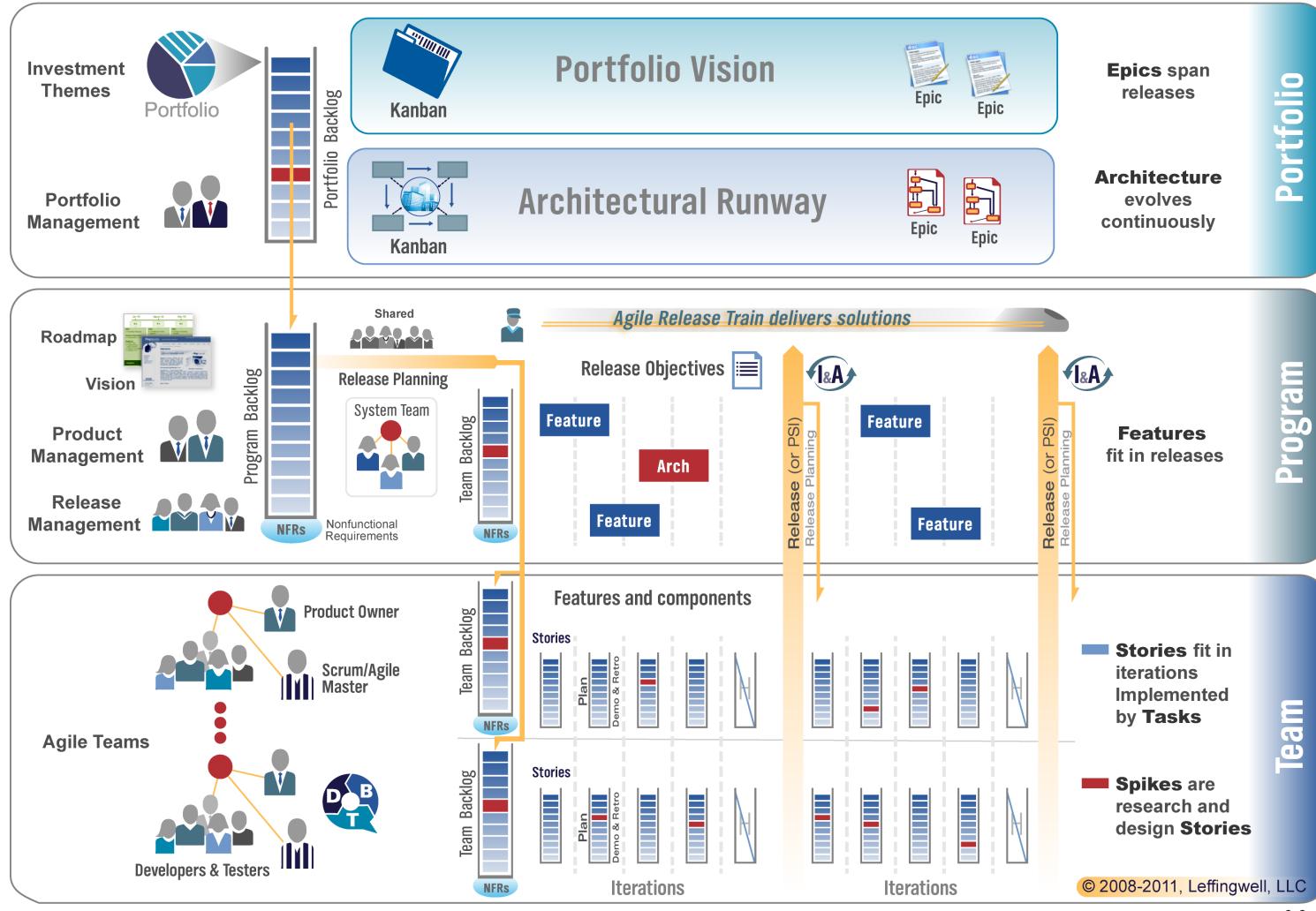
Scrum of scrums provides one technique to scale Scrum to larger projects



<https://www.agilest.org/scaled-agile/scrum-of-scrums/>

Overview of SAFe

Scaled Agile Framework™ Big Picture



See also www.scalingsoftwareagility.wordpress.com and Leffingwell, D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley (Pub. 2011)

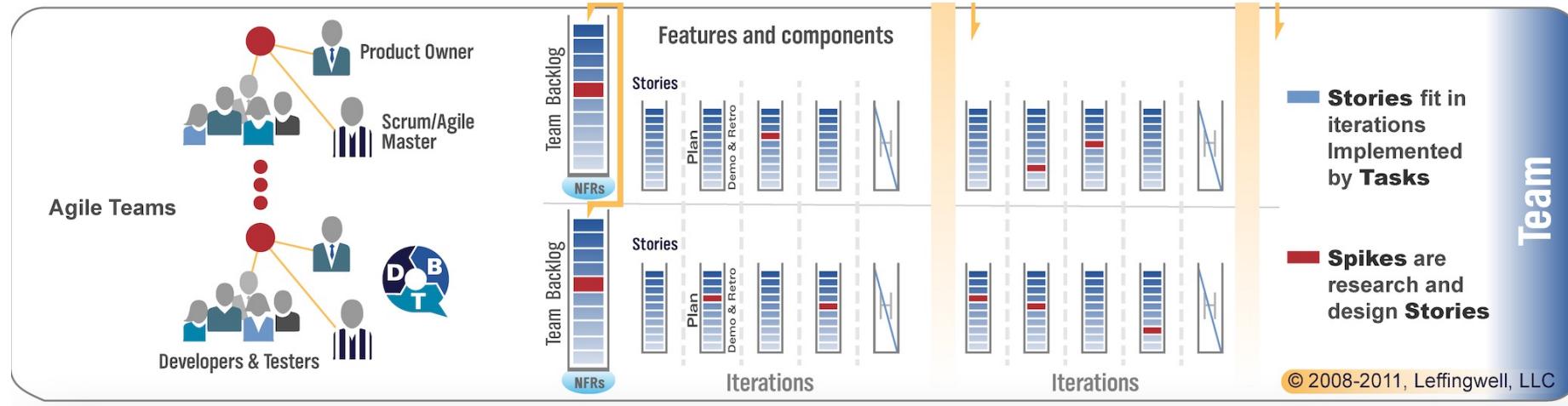
<https://scalingsoftwareagility.wordpress.com/>

v 0.8

Three levels of organization

Level	Role	Title	Requirements
Portfolio	Long-term planning	Portfolio Manager	Epics
Program	Large system management	Product Manager	Features
Project	Traditional Scrum teams	Product Owner	User Stories

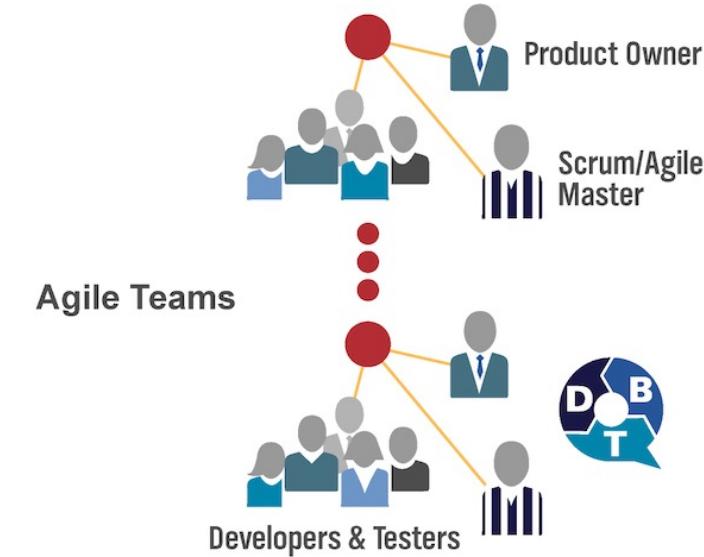
Project Level of SAFe



- Agile teams define, implement, and test user stories.
 - Use Lean, Scrum and/or Kanban
 - Deliver components or features
- Small enterprises may have a few teams.
- Larger enterprises may have “**pods**” of teams.

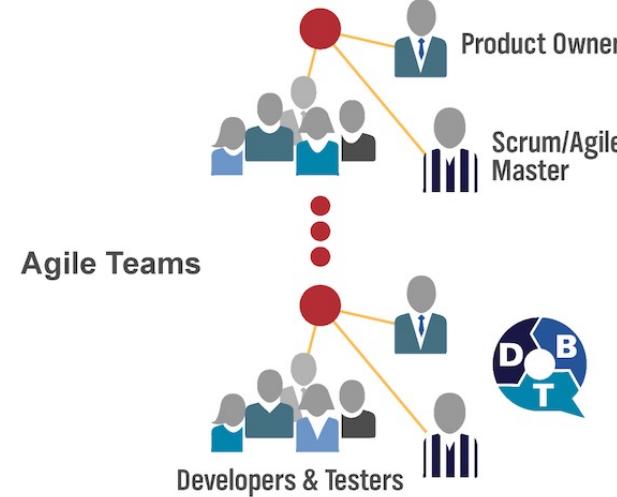
Project Level: Agile Teams

- 5-9 people
- Organized around a single component or feature
 - Define user stories
 - Build
 - Test
 - May do acceptance testing
- May be supported by specialists:
 - Architects
 - Technical writers
 - Internal IT



Project Level: Pods of Agile Teams

- Larger systems may require groups of teams
 - Deliver features, systems/subsystems
- 3-10 teams cooperating with one another
- 50-100 people collaborating face-to-face
- Each team primarily works alone but the other teams are readily available for collaboration



Project Level: Roles on Agile Teams

- **Product Owner**

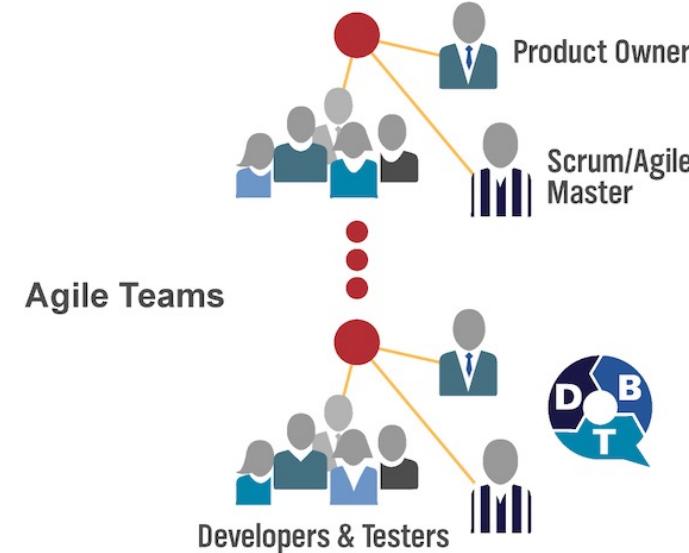
- Determine and prioritize requirements
- Maintain Product Backlog
- Collocated with the team for daily interaction

- **Scrum/Agile Master**

- Train and coach team in use of Scrum
- Act as "management/leadership proxy"
- Product management role

- **Developer**

- Define, build, test and deliver software
- May include "full or part-time architects, tech leads, user experience and documentation experts"



Project Level: Iterations (Sprints)

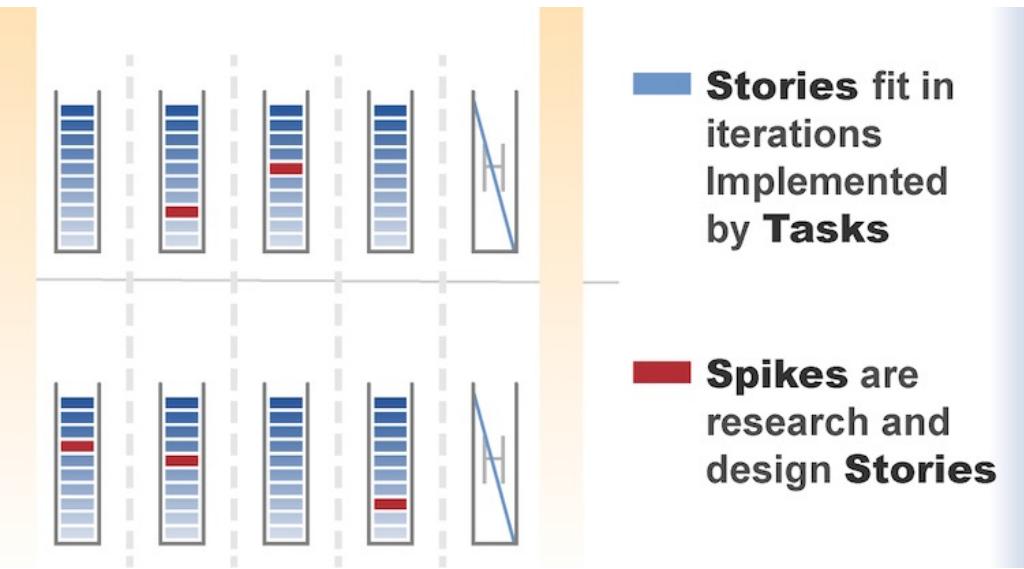
- Standard time-boxed intervals
- Iteration length is same for all teams
 - Typically, 2-4 weeks
 - Need to integrate modules across teams
 - Features may slip to meet time-boxed intervals
 - Must meet quality goals
- Each release contains:
 - 4-5 development iterations/sprints
 - 1 "hardening" iteration - resolve integration problems, etc.
 - Fully shippable increment every 90 days or so



Project Level: User stories and tasks in the backlog

- Each team has its own Iteration Backlog:

- User stories
 - Defects
 - Refactoring
 - Spikes



- Stories may be decomposed into Tasks for estimation and tracking

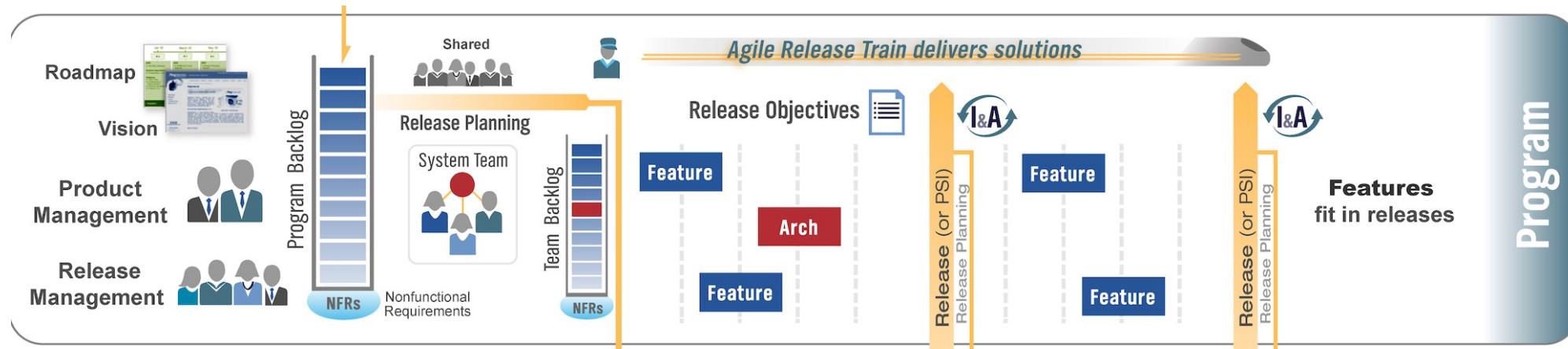
Project Level: .Agile Release Train

Multiple Agile teams delivering functionality

- Time boxed iterations with quality requirements.
- Features in each release may change to meet the schedule.
- Deliveries are made every time the train “enters the station”.

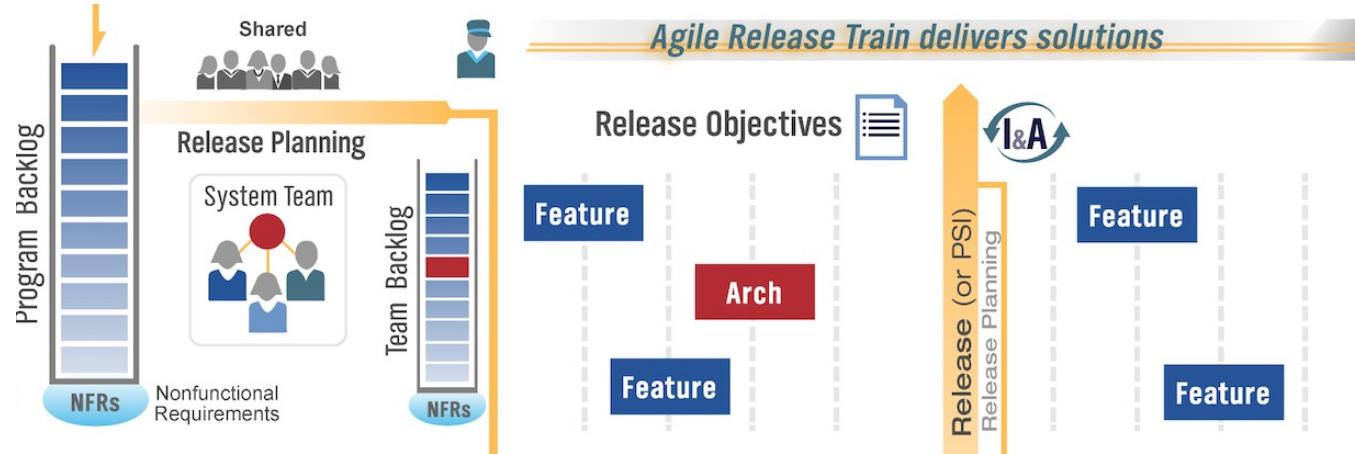


Program Level of SAFe



- System team and Release management team
 - System level testing and release evaluation
- Coordinate work of Agile Teams in the Project Level
- Understand how the work should evolve over time

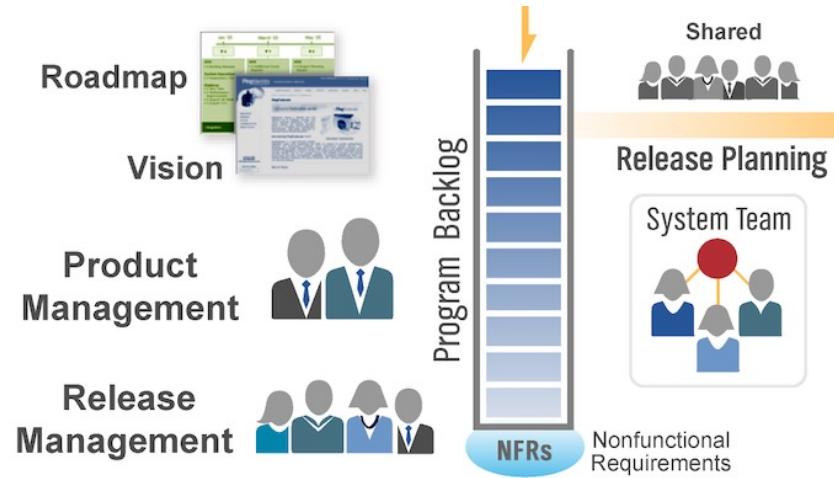
Program Level: Releases



- Each iteration produces a “potentially shippable increment”
- Typically, 2–4-month intervals between releases
 - Recall that the Agile teams are using 2-week sprints
- May not want to release every increment to customers:
 - Defects, refactoring to remove technical debt
 - May interfere with customer's licensing and service agreements
 - Potential disruption to customer

Program Level: *Vision and features*

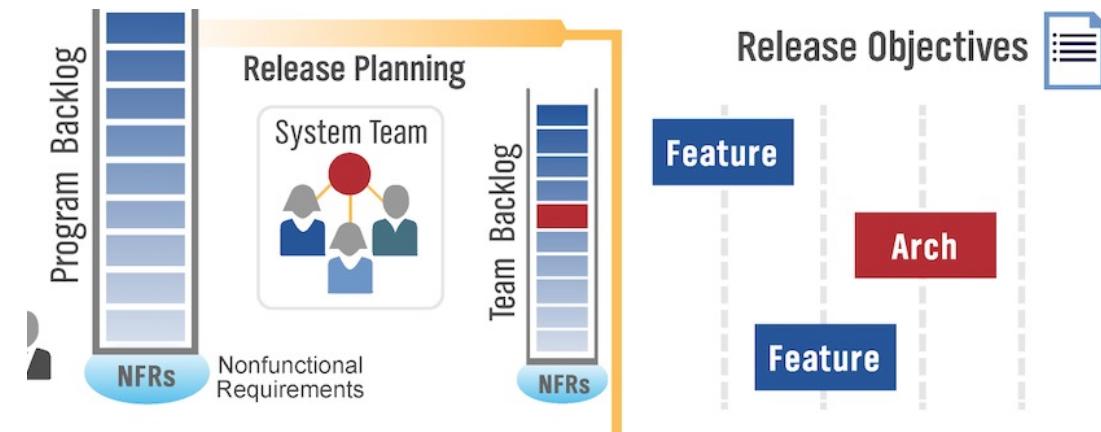
- Product Managers are responsible for overall Vision
- The Vision answers:
 - What problem does this solve?
 - What features and benefits does it provide?
 - For whom does it provide it?
 - What performance, reliability, etc. does it deliver?
 - What platforms, standards, etc. will it support?
 - Technical and marketing issues
- Release Backlog consists of prioritized features
 - Features later refined into stories
 - Release Backlog contains all desired features that haven't been delivered yet
- Specify non-functional requirements: “*ilities”
 - Scalability, reliability, compatibility, security, ...



Program Level: Release planning

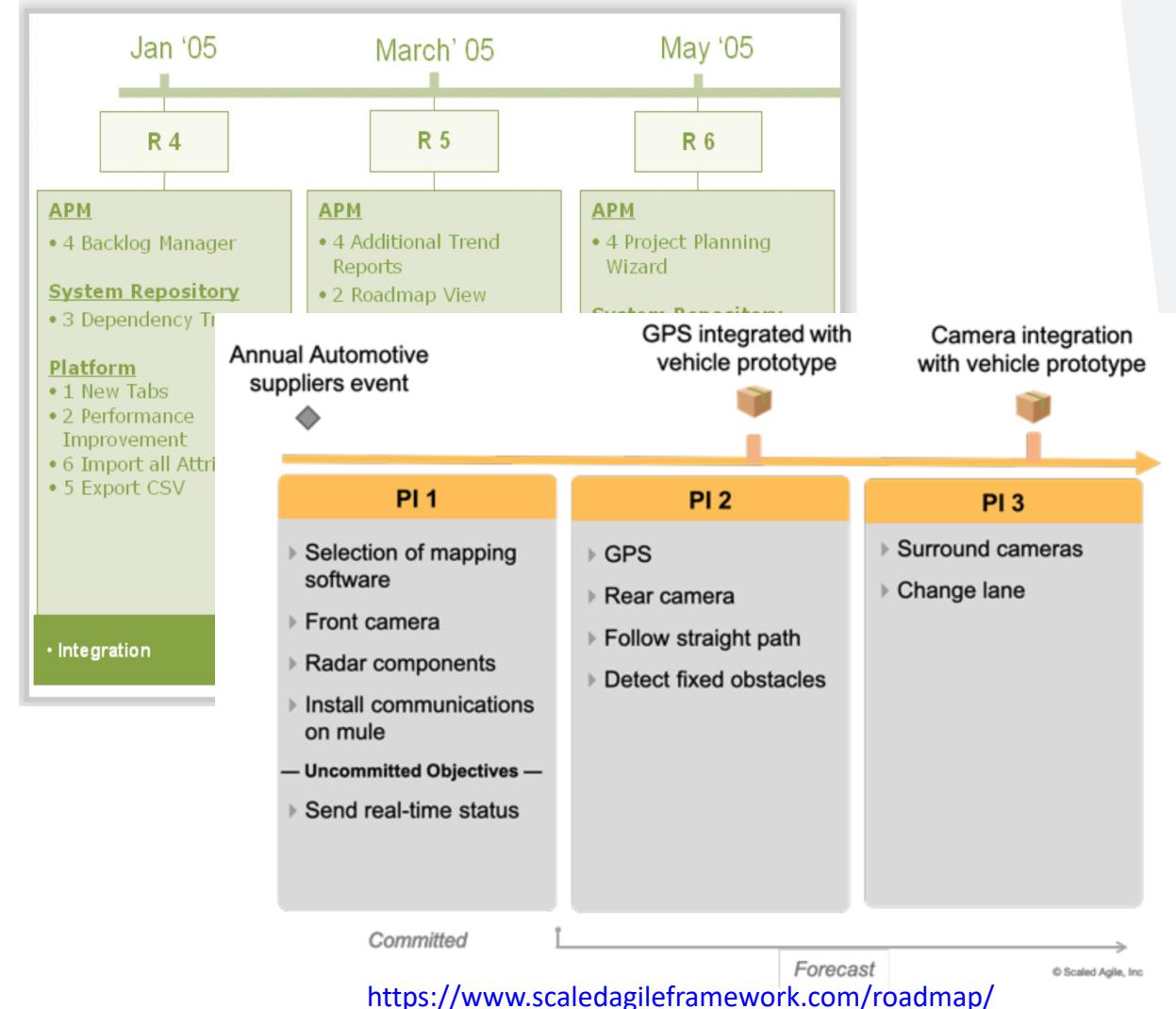
- Product Managers are responsible for Release Planning:

- Analogous to Sprint Planning
- Identify time box with typically multiple sprints
- Hold Release Planning meeting
- Use vision and business objectives to identify prioritized list of features to include in next release
- Create a **Roadmap** to describe the “plan of intent”



Program Level: The Roadmap

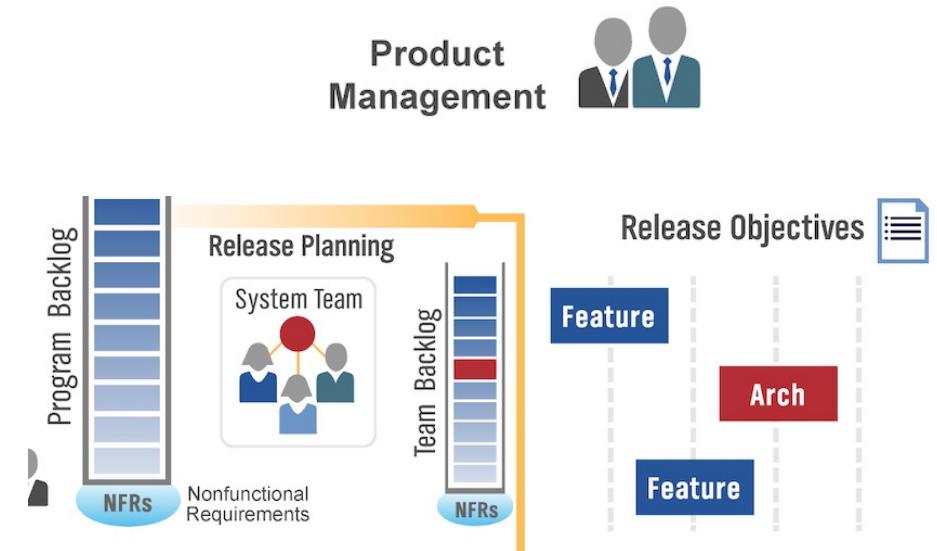
- Release planning breaks features into stories
- Stories assigned to teams
- Interdependencies are identified and addressed
- Result is a Roadmap:
 - Planned release dates
 - Each release includes:
 - Theme
 - Objectives
 - Set of features



Program Level: Product Management

- Scrum Product Owners:
 - Represent the interests of everyone with a stake in the resulting project
 - Achieve initial and ongoing funding
 - Create initial requirements, return on and release plans

- SAFe Product Managers responsibilities are broader:
 - Define the product
 - Position the solution in the market



Portfolio Level of SAFe



- Portfolio Managers responsibilities:
 - Manage enterprise investments
 - Manage enterprise resources
- Investment themes
- Epics

Portfolio Level: Investment Themes

- Large enterprises make decisions once or twice per year on investment themes:
 - Existing offerings (enhancements, support and maintenance)
 - New offerings
 - Futures - require investment now, but will not contribute toward revenue until later
 - Sunset - end support for existing products and services

Themes: how should the enterprise allocate financial and other resources?



Fintech

[View fund](#)



Future of Transport

[View fund](#)



Sustainable Energy

[View fund](#)



Circular Economy

[View fund](#)



Next Generation Technology

[View fund](#)



Nutrition

[View fund](#)

<https://www.blackrock.com/lv/individual/themes/therapeutic-investing/why-invest-thematically>

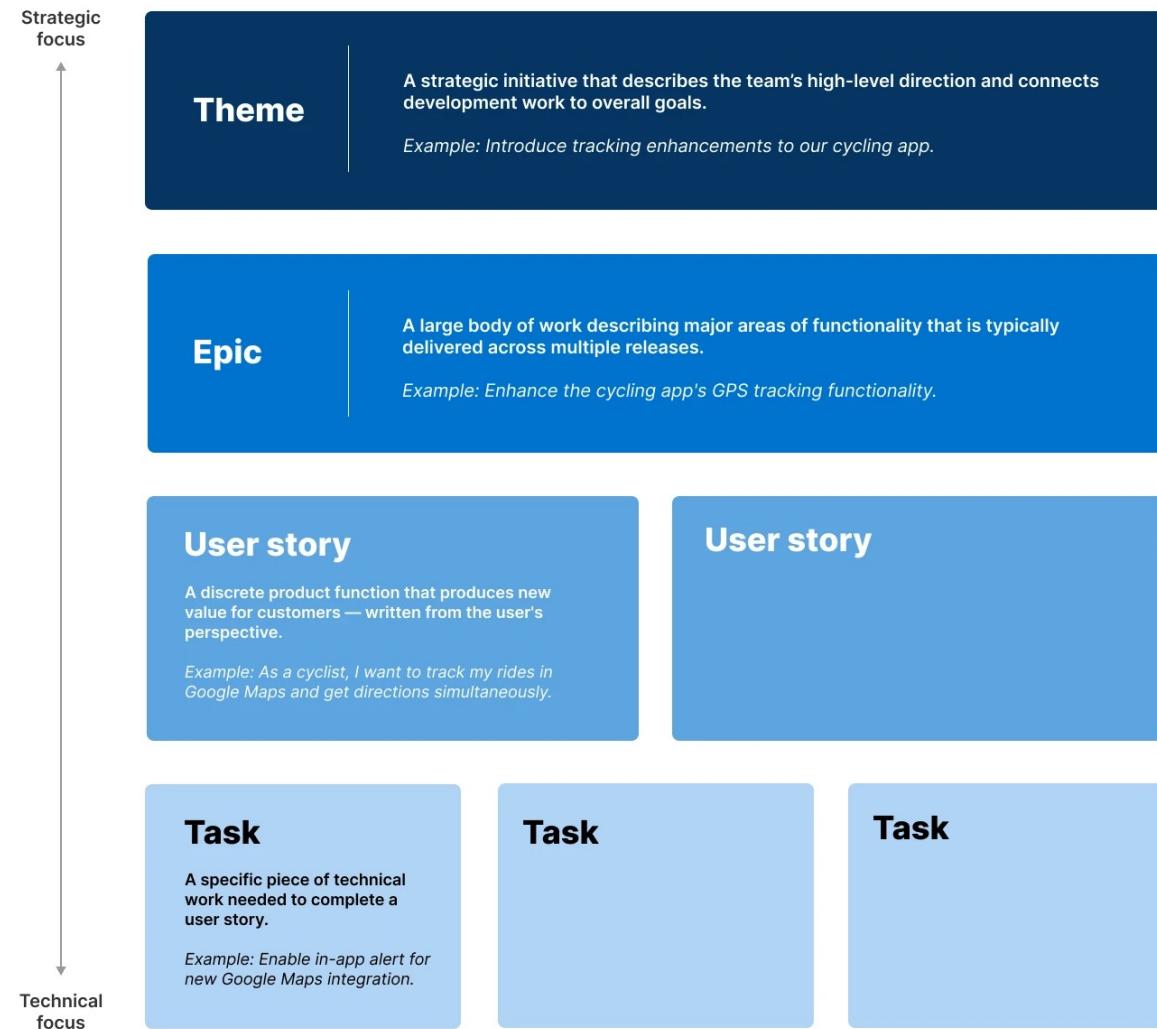
Portfolio Level: Investment Themes

- Themes identify key value propositions
 - Differentiate the company's products
 - Provide competitive advantage
- Themes may span multiple years



Portfolio Level: Epics, Features, Stories, Tasks

- Themes are later defined as Epics in the Portfolio Backlog
 - Epics describe strategic intent
- Epics are further refined into Features
- Features are broken into Stories
- Stories are broken into Tasks



<https://www.aha.io/roadmapping/guide/agile/themes-vs-epics-vs-stories-vs-tasks> © 2022 Aha! Labs Inc.

Three levels of backlogs

Level	Backlog	Contents
Portfolio	Portfolio backlog	Backlog of prioritized epics describing strategic intent
Program	Release/product backlog	Backlog of prioritized features
Project	Iteration backlog	Backlog of user stories aligned to business needs

Portfolio Level: Architectural Runway



"A system with architectural runway has existing or planned infrastructure sufficient to allow incorporation for current and near-term requirements without excessive refactoring." - Leffingwell

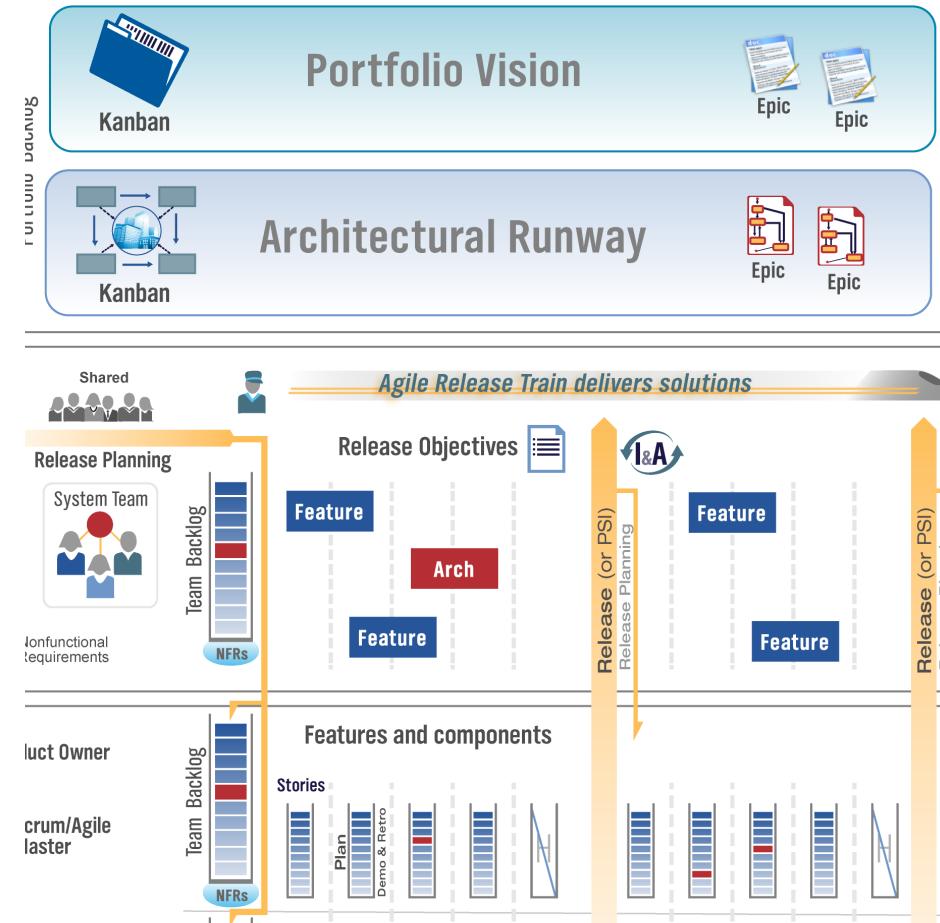
Failure to maintain architectural runway may lead to:

- Missed release dates due to large-scale refactoring
- Slowed velocity
- Inability to create new features

Portfolio Level: Architectural Runway

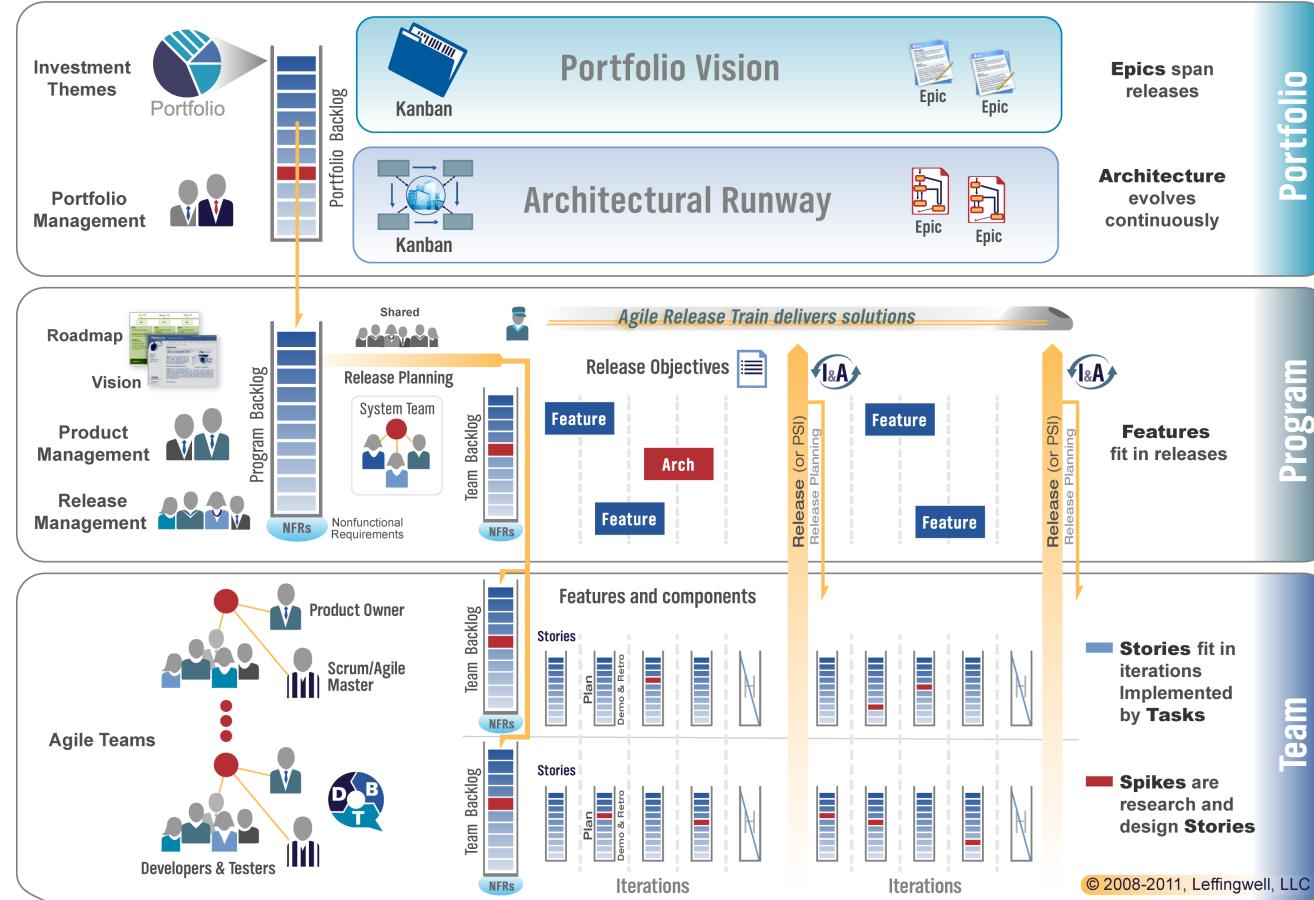
Architectural Runway spans all three levels

- Portfolio
 - Infrastructure initiatives
 - Architectural epics
- Program
 - Architectural features for releases
- Project
 - User stories



SAFe Summary

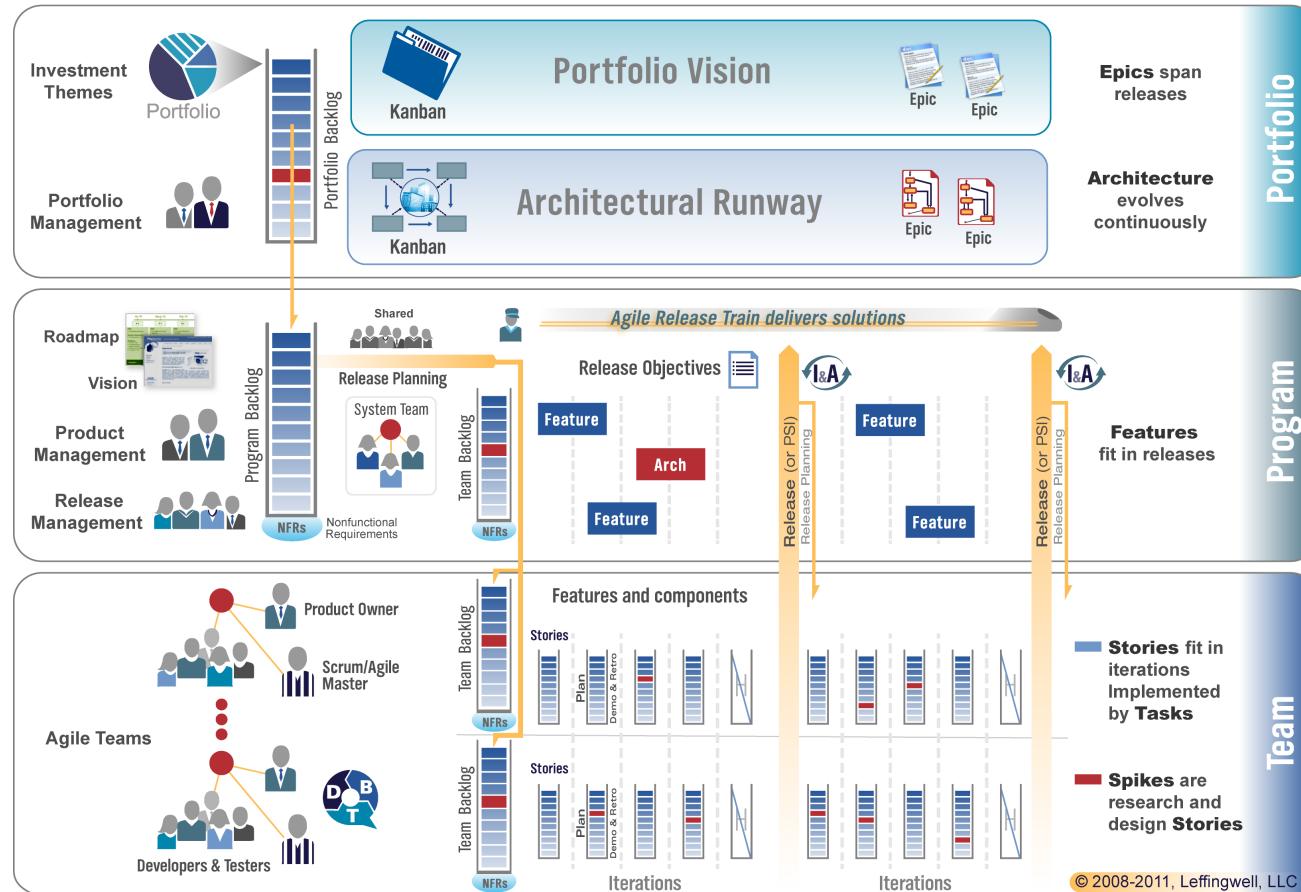
Scaled Agile Framework™ Big Picture



- Resource investment vision
- Overall architecture
- Manage large organizations
- Focus on releases to meet customer needs
- Traditional Scrum/Kanban teams
- Optional pods

SAFe Summary

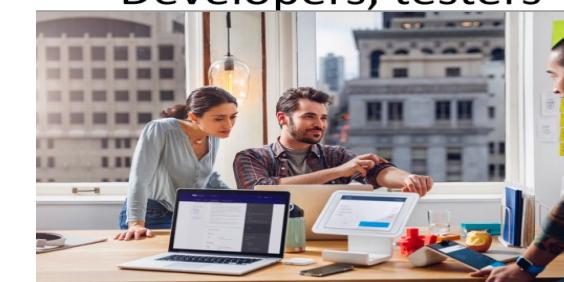
Scaled Agile Framework™ Big Picture



Executives



Middle managers



Developers, testers

Questions?





THANK YOU

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030

