CLOUD ENGINEERING

Data Science Project Management

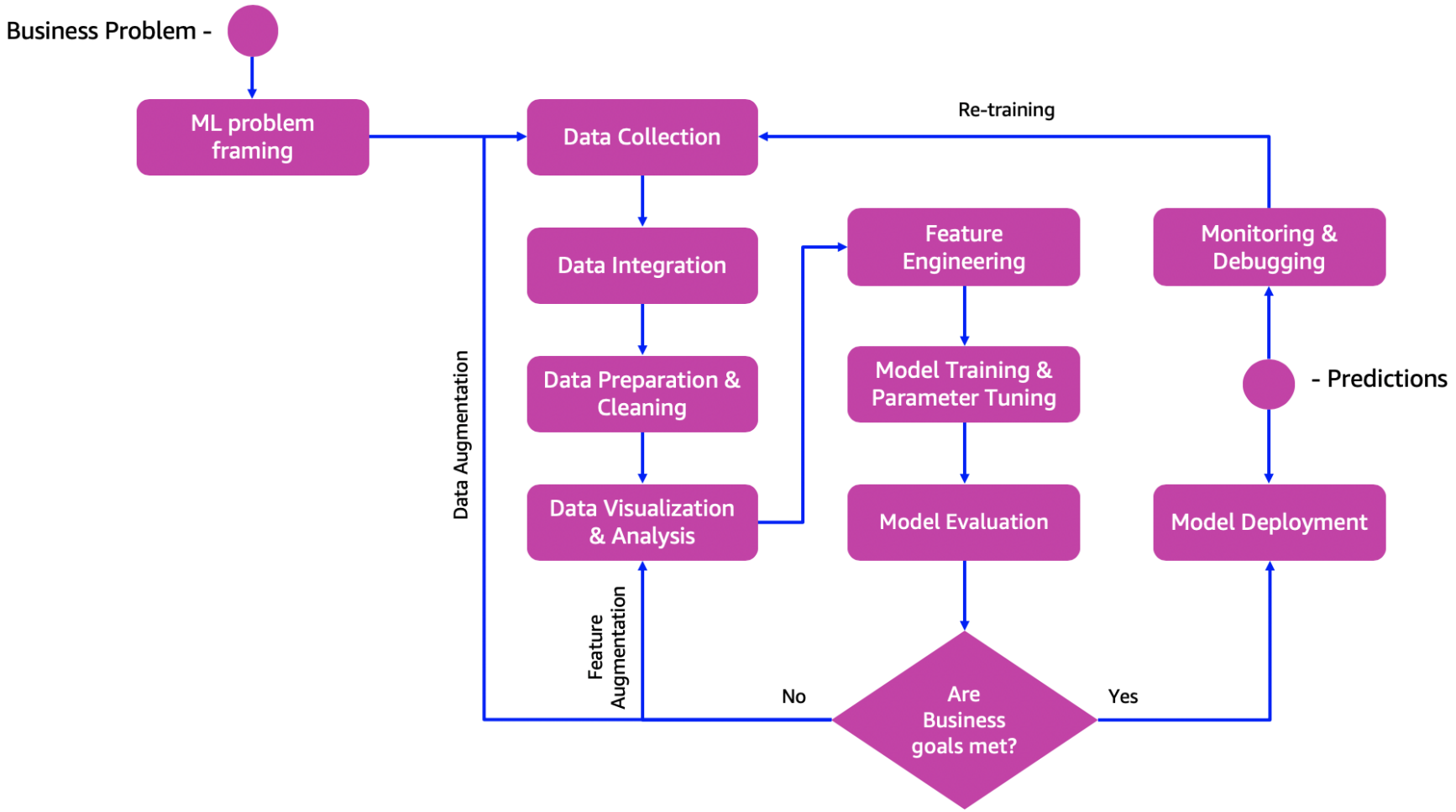
Ashish Pujari

Lecture Outline

- Data Science Project Phases
- Agile Process

Data Science Project Phases

Data Science Flow



1.Planning and Project setup

- Define the business problem
- Define project scope and high-level requirements
- Define model consumption patterns
- Define model SLAs – target accuracy, latency, drift, etc.
- Determine project feasibility
- Discuss general model tradeoffs (accuracy vs speed)
- Set up project code repository
- Determine team skills and size

2. Data Collection

- Identify data sources
- Build data ingestion pipeline
- Validate quality and volume of data
- Data labeling and ground truth
- Collect more data as needed

3. Data and Model Exploration

- Perform EDA (exploratory data analysis)
- Understand training and test data distributions
- Research available models and SoTA for the problem domain
- Establish baselines for model performance
- Start with simple model(s) using basic data pipeline
- Experiment with other models and ideas during early stages
- Develop baseline model

4. Model Development

- Feature Engineering
- Develop model training, testing and evaluation pipelines
- Perform model hyperparameter tuning
- Revisit model scope and evaluation metrics as needed
- Follow data science and coding standards and best practices:
  - Unit testing
  - Logging
  - Monitoring
  - Error Handling
- Create initial version of production-ready model(s)

5. ML Ops

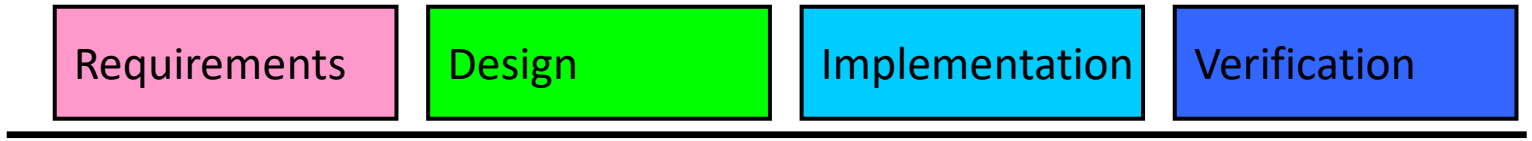
- Model deployment
  - Automated model deployment pipelines
  - Model endpoints/Offline models
  - Maintain the ability to roll back model to previous versions
  - Monitor live data and model prediction distributions
- Ongoing model maintenance
  - Perform A/B Testing
  - Monitor model drift, accuracy and SLAs
  - Periodically retrain model to prevent model staleness
  - Model improvements

AGILE METHODOLOGIES

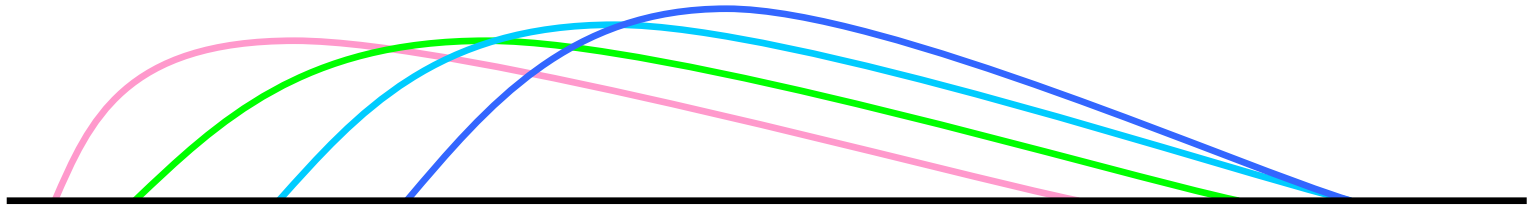
Software Development Lifecycle

- Agile methods have replaced the traditional sequential waterfall approach

Sequential Waterfall Approach



Agile Overlapping Approach



Source: "The New New Product Development Game", Hirotaka Takeuchi and Ikujiro Nonaka, *Harvard Business Review*, January 1986.

Agile

- Agile is an iterative approach which uses cross-disciplinary teams to rapidly build software products and solutions
- Benefits
  - Faster time to market
  - Increased customer satisfaction
  - Values employees
  - Minimizes rework

Agile Methodology: Characteristics

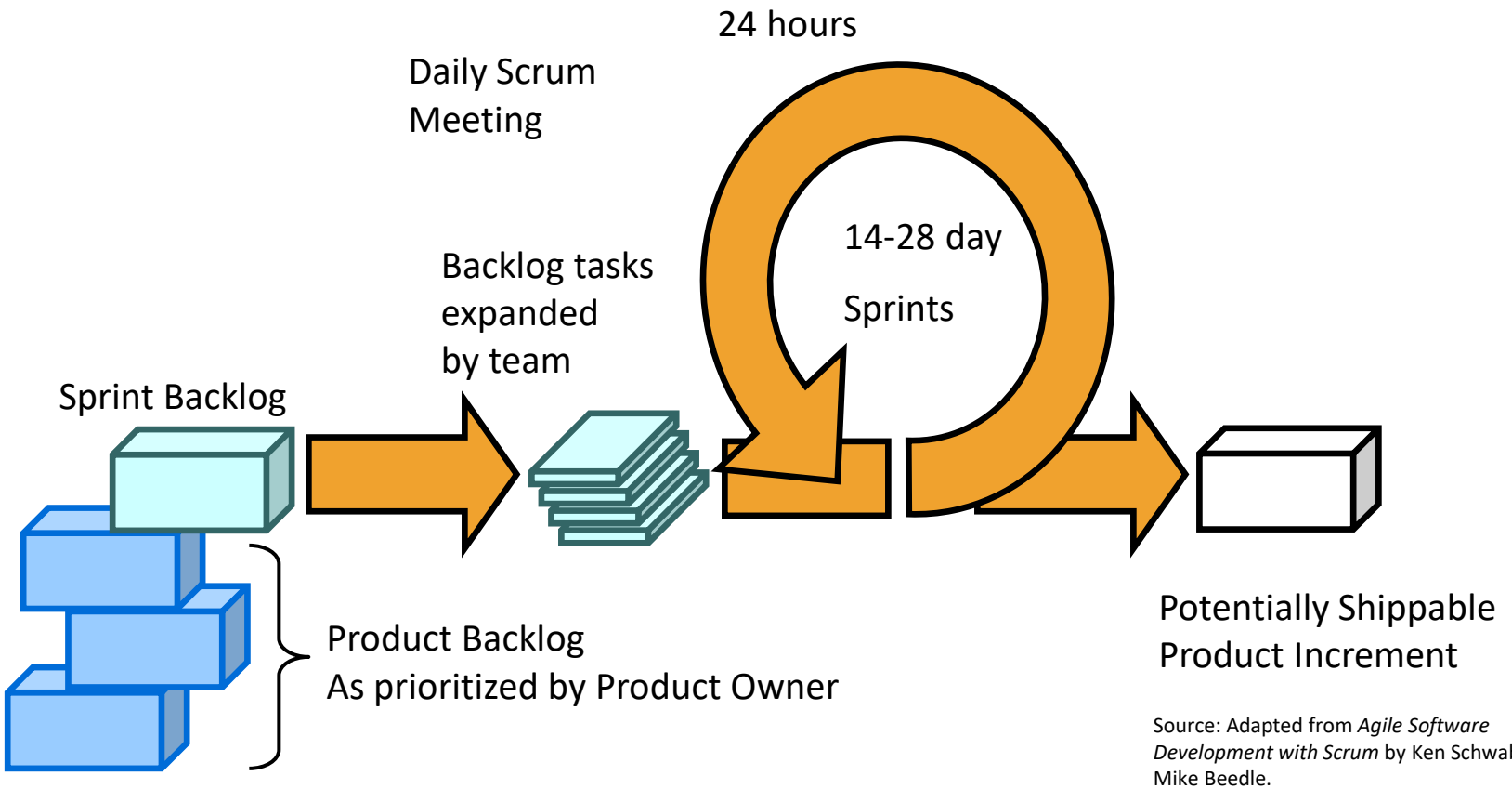
- Transparent
- Seamless communication
- Driven by trust and teamwork
- Minimal documentation
- Pair programming
- Rough estimates
- Fail fast

Agile Methodology: Variations

- Scrum
- Kanban
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Lean Software Development (LSD)

Agile Scrum Process

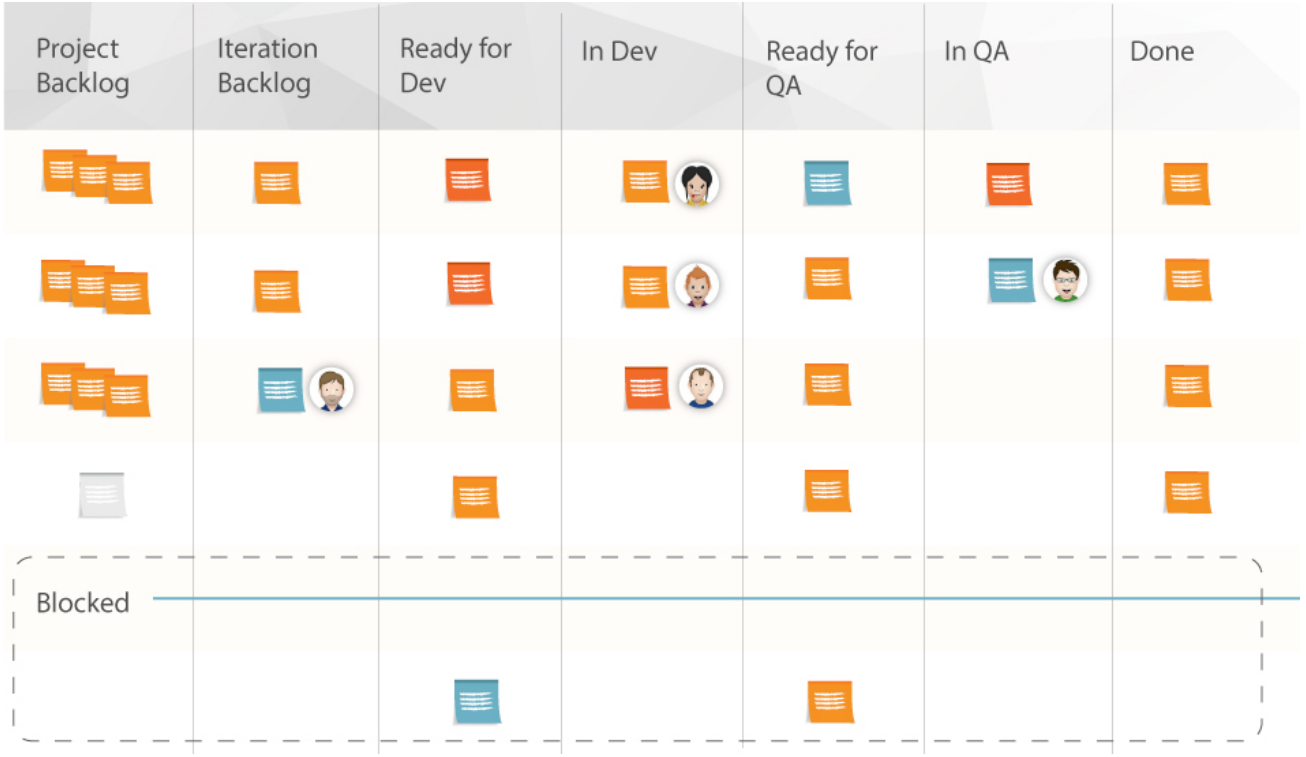
- Hands-on system consisting of simple interlocking steps and components



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Agile Scrum Features

- Daily standup
- Sprint planning meetings
- Team review (show and tell)
- Retrospective meetings
- End-of-phase retrospectives
- User stories
- The backlog
- Team walls



The Scrum Team

- Typically 5-10 people
- Cross-functional
  - Business, Data Engineering, Data Science, QA, Developers, UI Designers, etc.
- Members should be full-time
  - With some exceptions (e.g., System Admin, etc.)
- Teams are self-organizing
  - What to do if a team self-organizes someone off the team??
  - Ideally, no titles but rarely a possibility
- Membership can change only between sprints

Primary Roles

- Scrum Master
  - Represents management to the project
  - Typically filled by a Project Manager or Team Leader
  - Keeps the team focused on the goal.
  - Main job is to remove impediments
- Product Owner
  - Represents the clients/stakeholders
  - Responsible for product features and requirements
  - Typically filled by a Business domain expert
  - Main job is to build out product backlog, prioritize it and answer questions
- Team Member
  - All other members of the agile team

Business Requirements

- User Story
  - Smallest unit of work in an agile framework.
  - Informal, general explanation of a software feature written from the perspective of the end user or customer.
  - Example: As a *Risk manager* (role) I would like the lending application to *predict the chances of loan default* (what) for a given individual based on several factors so that *I can make an informed lending decision* (why)
- Epic
  - A collection of related user stories
  - Example : “Customer Risk Evaluation Epic”

Business Requirements

- Acceptance Criteria
  - Every user story should include a detailed acceptance criteria
  - The list of items that if delivered would complete the user story
  - Example
    - Application should make a prediction within a day (SLA)
    - The risk model and decisioning process should be available for audit
    - The risk rating should be displayed to the lending agent on the UI as a percentage
    - Etc.
- User Story Estimation
  - Team members provide estimates
  - Typically uses the Fibonacci series to measure degree of work
    - 1, 2, 3, 5, 8, 13,..
    - If a story is too big create an epic and break down into multiple user stories

Product Backlog

- The product wish list
  - Usually a combination of
    - story-based work (“let user search and replace”)
    - task-based work (“improve exception handling”)
- Prioritized by the Product Owner
  - Typically a Product Manager, Marketing, Internal Customer, etc.

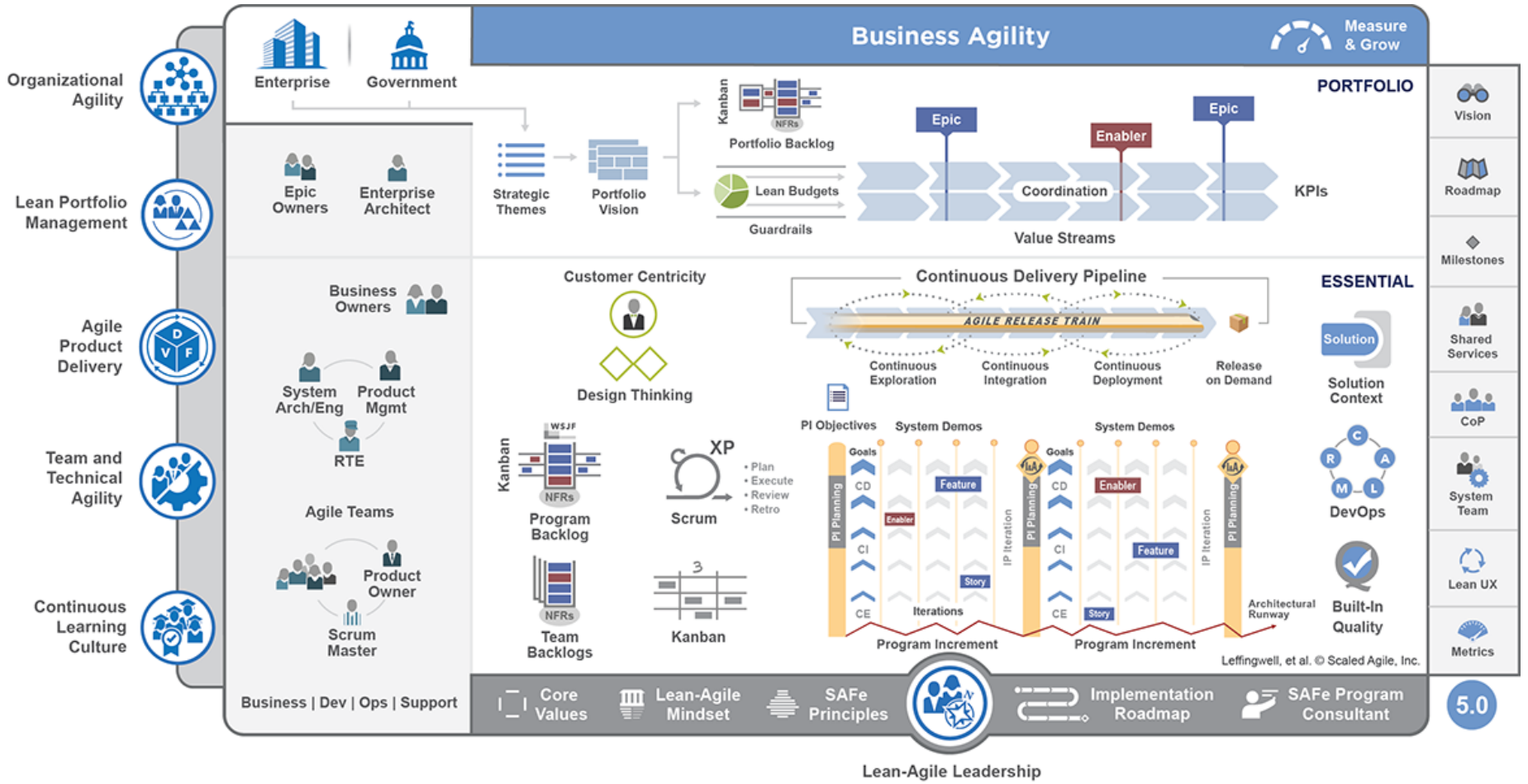
	Item #	Description	Est	By
Very High				
	1	**Finish database versioning**	16	KH
	2	**Get rid of unneeded shared Java in database**	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Scrum Calendar

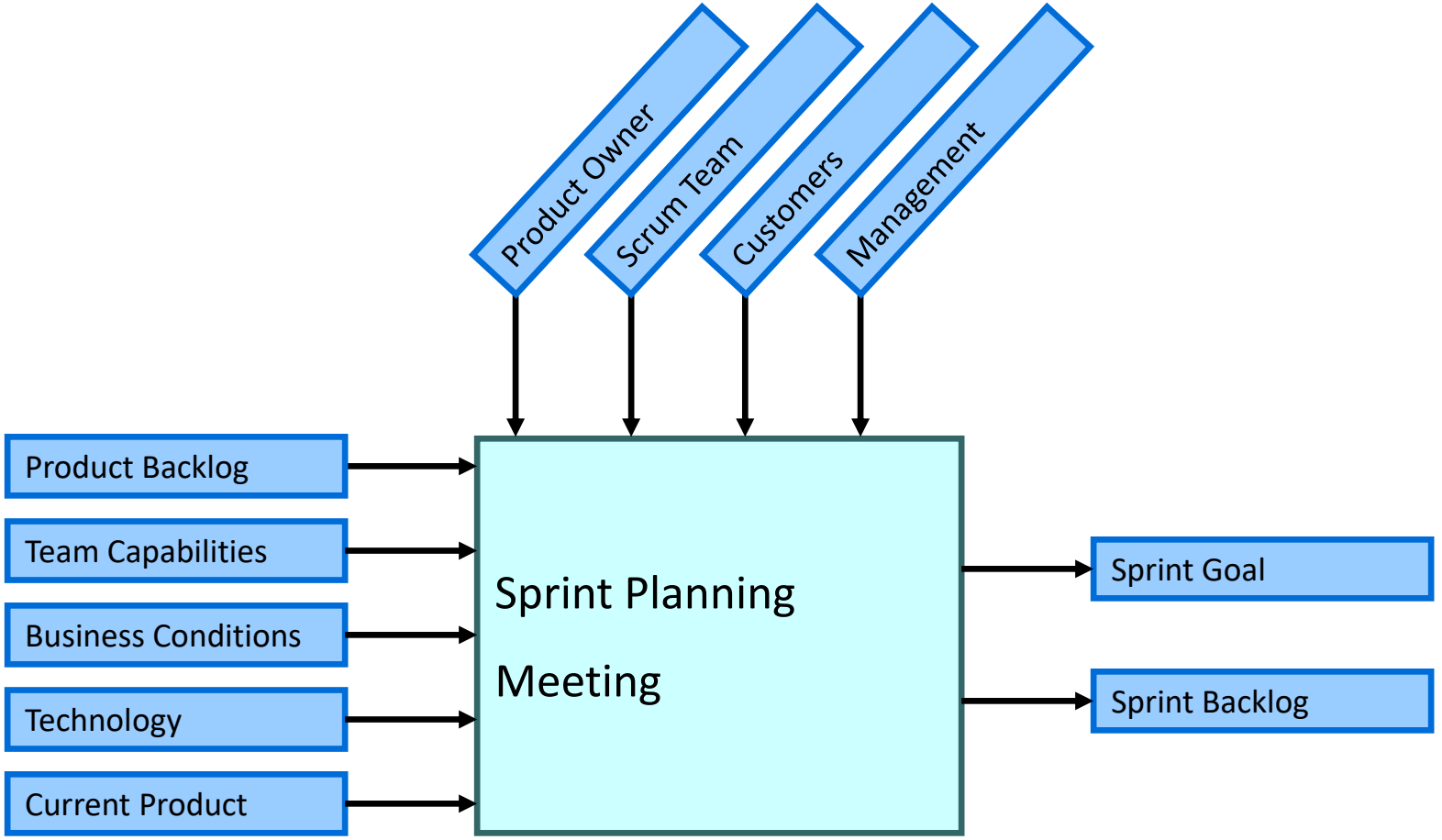
Monday	Tuesday	Wednesday	Thursday	Friday
16	17	18 **Sprint Planning (Sprint 1)**	19 Daily standup	20 Daily standup
23 Daily standup	24 Daily standup	25 Daily standup **Backlog Refinement**	26 Daily standup	27 Daily standup
30 Daily standup	31 Daily standup	1 Daily standup **Backlog Refinement**	2 Daily standup	3 Daily standup
6 Daily standup	7 **Sprint Review (Sprint 1)** **Sprint Retrospective (Sprint 1)**	8 **Sprint Planning (Sprint 2)**	9 Daily standup	10 Daily standup

http://www.agileuprising.com/2017/01/01/sprint-schedule-template/

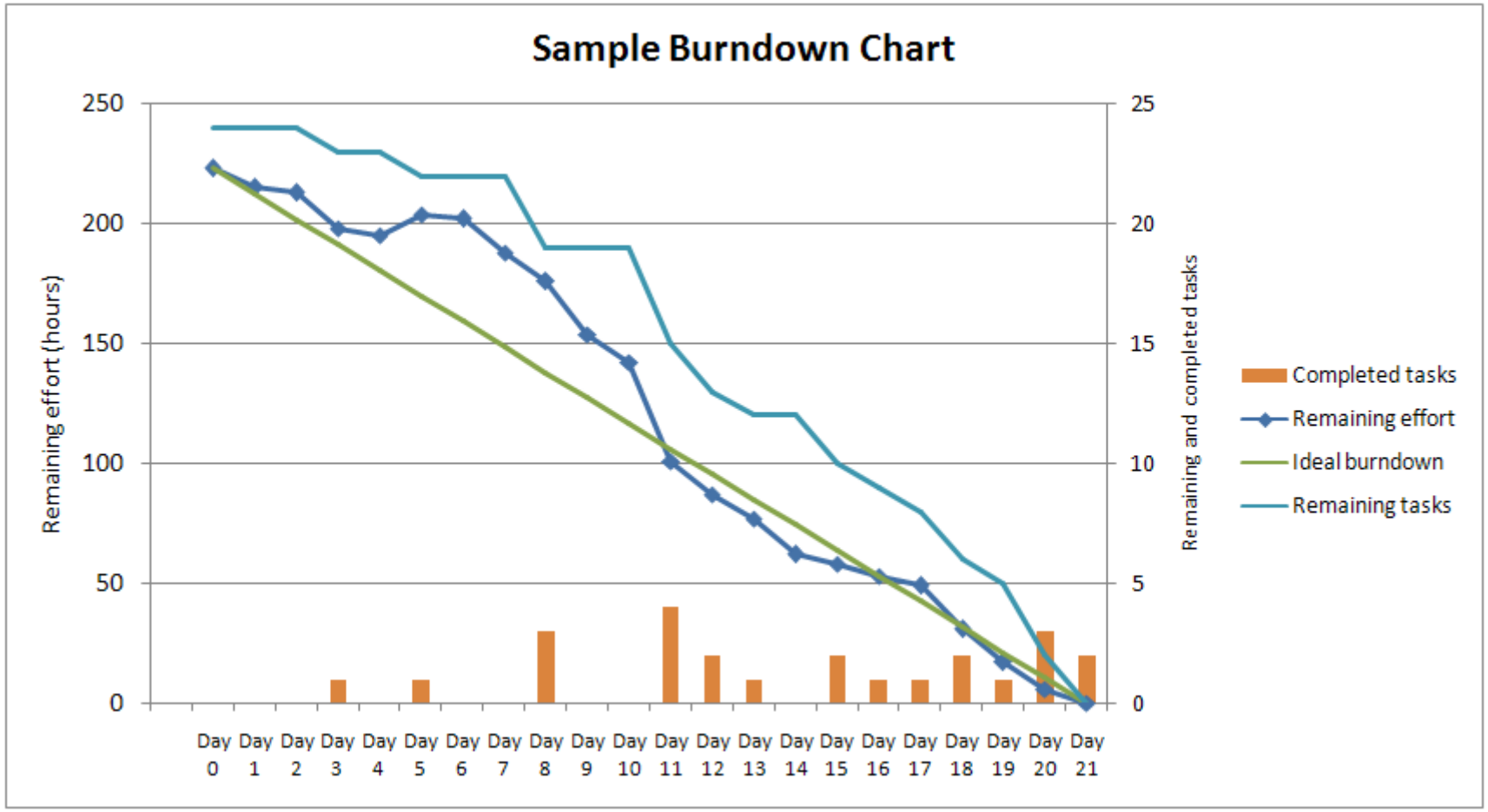
SAFE Agile – Scaling Agile



Sprint Planning

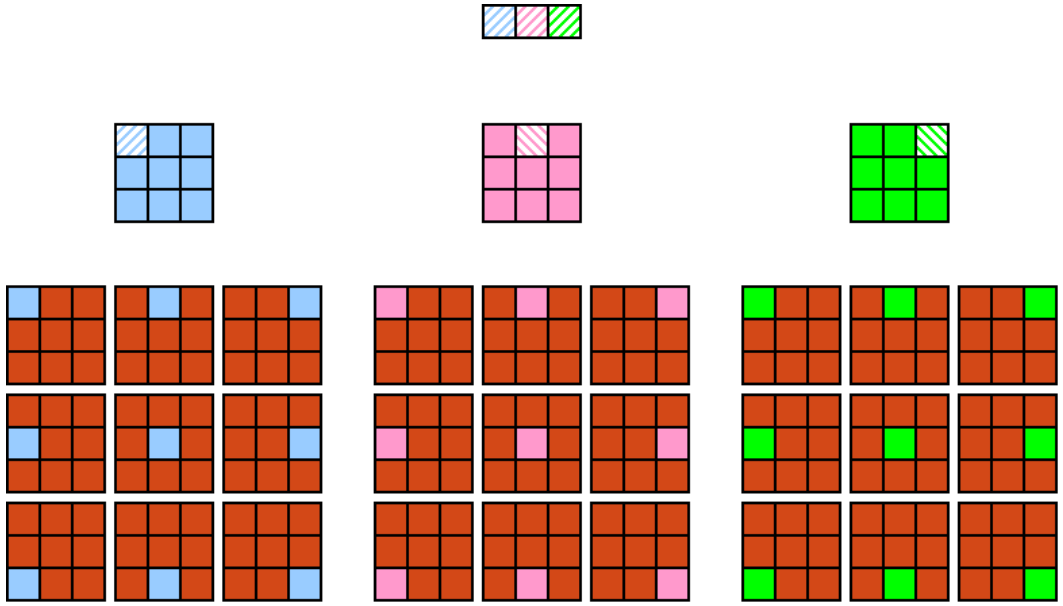


Measuring Sprint Progress

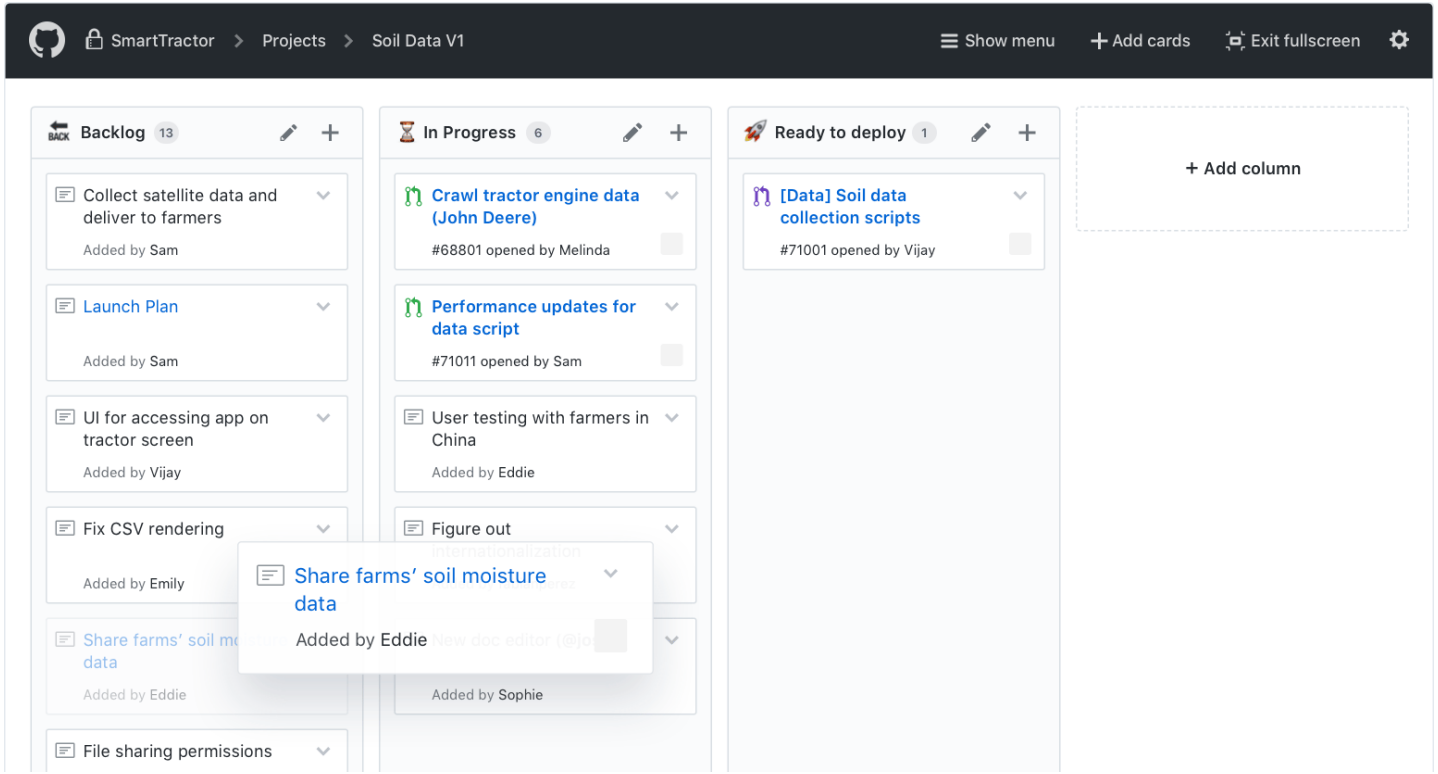


Scalability of Scrum

- Typical Scrum team is 5-10 people
- Scrum has been used with very large groups (600+)
- Better to have several small teams - “Scrum of Scrum” appraoch
- SAFe Agile



Agile Tools



CollabNet | Axosoft | Wrike | ActiveCollab, LLC | Kanbanize | Asana | Trello | Zoho Corporation | LeanKit Inc. | Planbox Inc. | Backlog | ProofHub, LLC | Targetprocess Inc. | Assembla | Git