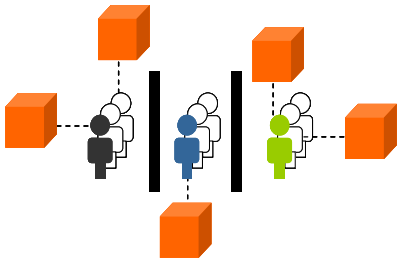
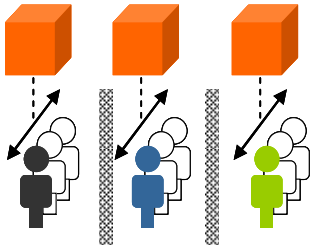
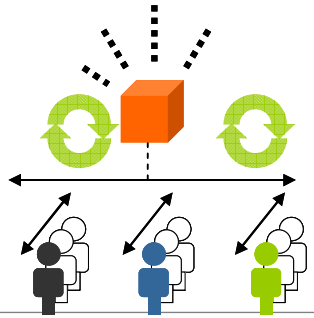
Measured Improvement in Software Economics

Walker Royce
VP, Chief Software Economist
IBM Software Group, Rational

An Evolving Software Delivery Model

Collocated development	Outsourcing/Co-sourcing	Integrated Software Supply Chains
		
Country-based delivery	Onshore / Offshore	Networked Global Centers
Utilization based	Deliverables based	Outcome based
Team collaboration	Project collaboration	Community Collaboration
No workflow management	Manual workflow management	Automated workflow management
No reuse	Ad hoc reuse	Systematic reuse
Limited measurement	Visible project metrics	Pervasive transparent metrics
Waterfall governance	Engineering governance	Economic governance
Technology Platform		
Standalone tools, process	Limited tool integration	Consistent platform Integrated, Collaborative, Optimized

Accelerated delivery demands a quid pro quo

Engineering Practitioners

Embrace Measurement

- Design, create, test
- Reuse knowledge, best practices
- Address uncertain things first
- Be adaptive to change

The Speed Of Trust

Governance Stakeholders

Enable Agility

- Achieve predictable outcomes
- Manage risk
- Ensure compliance
- Improve software economics
- Visibility and transparency

Software Delivery is an Economic Discipline

Level 5: Completely irreducible uncertainty

Level 4: Partially reducible uncertainty

Level 3: Fully reducible uncertainty

Level 2: Risk without uncertainty

Level 1: Complete certainty

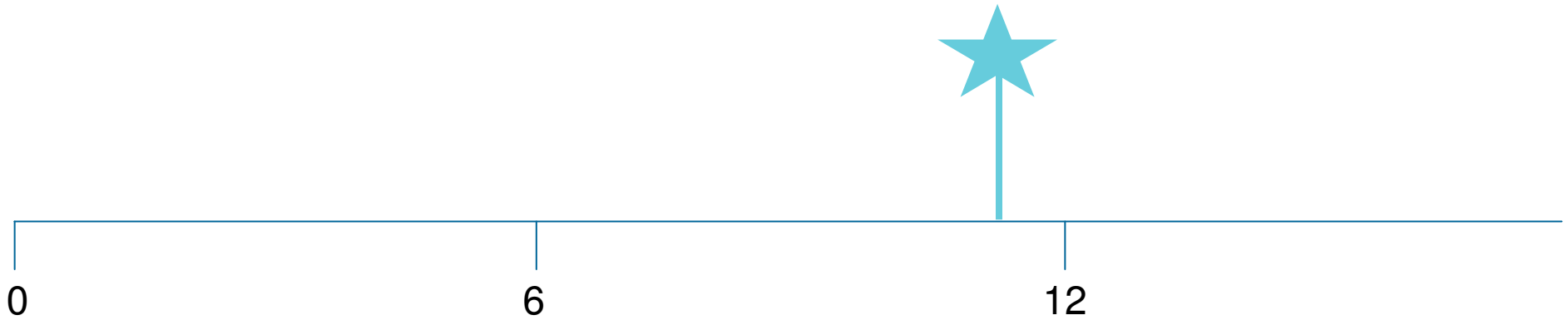
Engineering → Software →
Engineering → Software →
Engineering → Software →
Engineering → Software →

Religion
Philosophy
History
Economics
Biology
Chemistry
Physics
Mathematics

Lo, Andrew, and Mark Mueller. MIT Sloan School of Management, Moody's/NYU 6th Annual Credit Risk Conference, New York, March 2010.

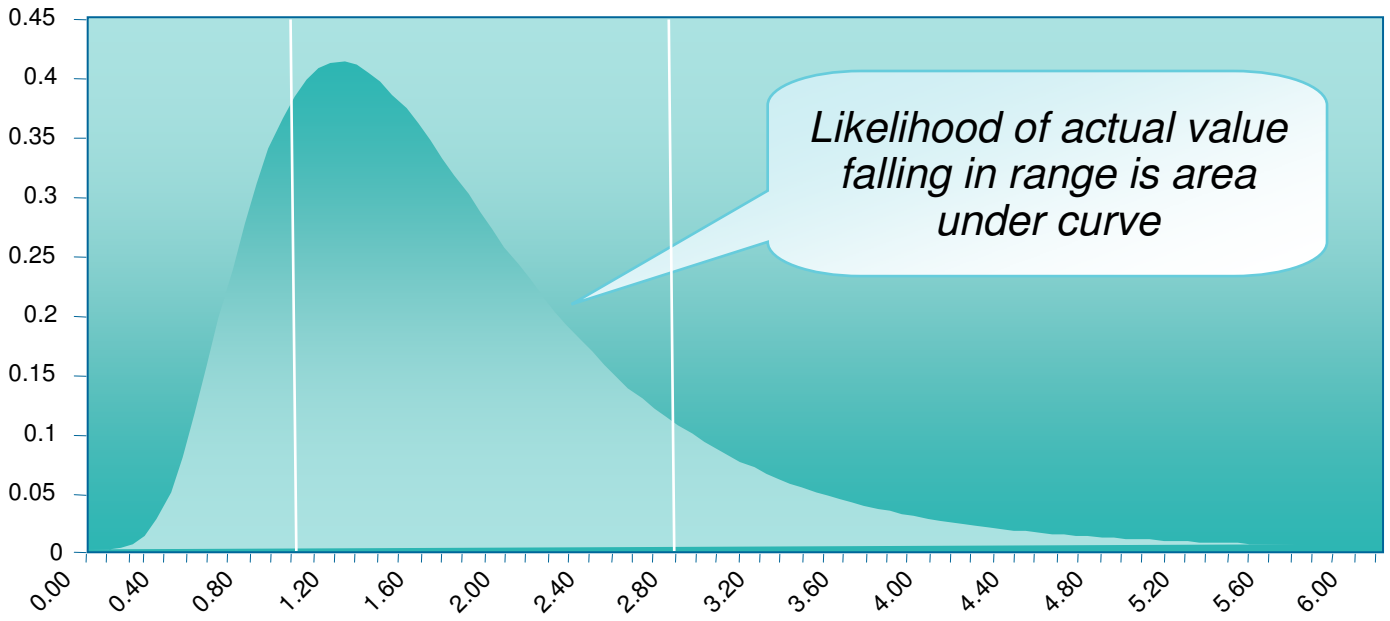
Schedule risk: Imagine you have 12 months to deliver a business critical system

- Your estimators tell you it will be done in 11 months
- What do you do with the information?
  - Rest easy, believing there is no risk?



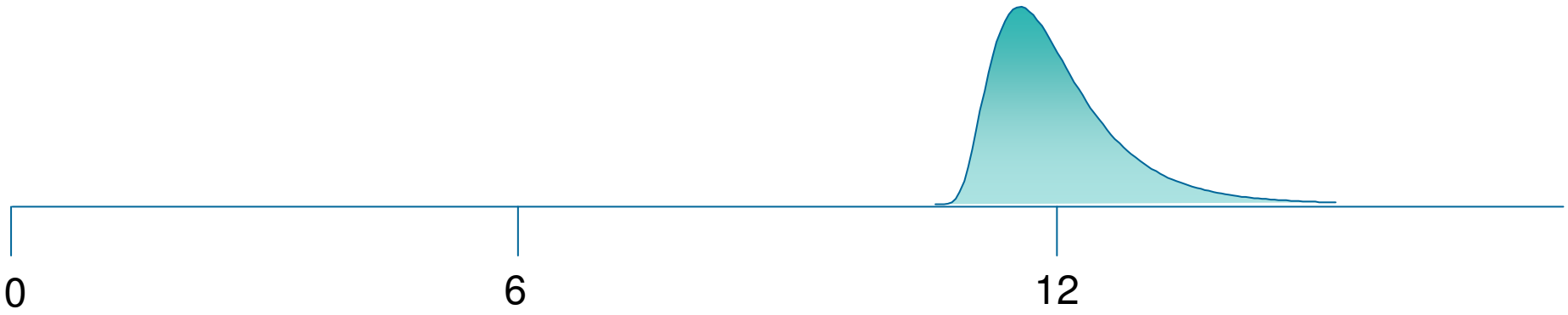
Maybe you realize that program parameters (cost, schedule, effort, quality, …) are random variables

- Area under curve describes probability of measurement falling in range



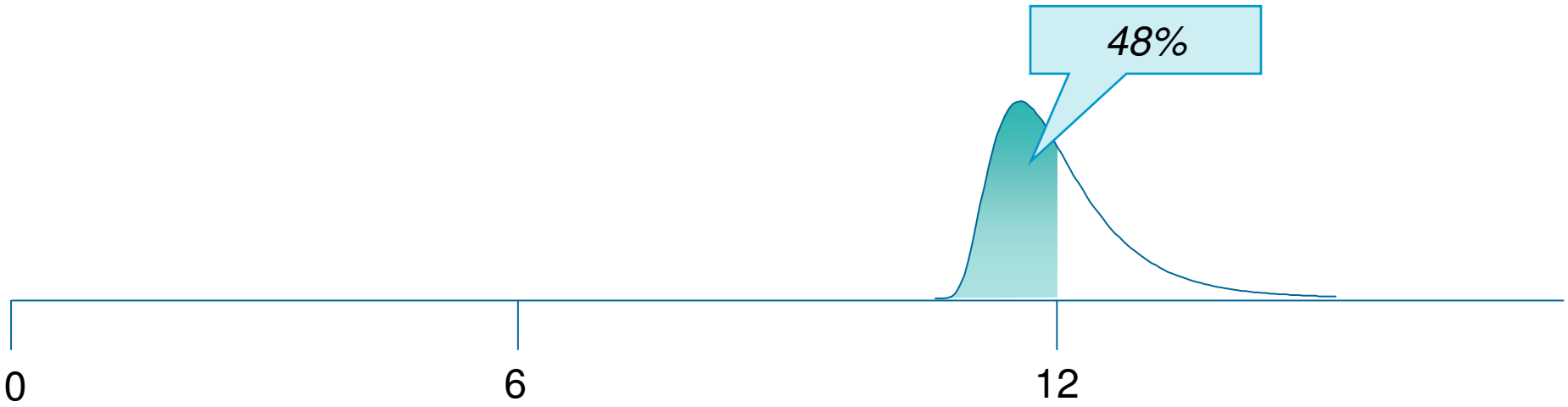
Imagine you have 12 months to deliver a business critical systems

- So you ask for the distribution and discover there is some uncertainty



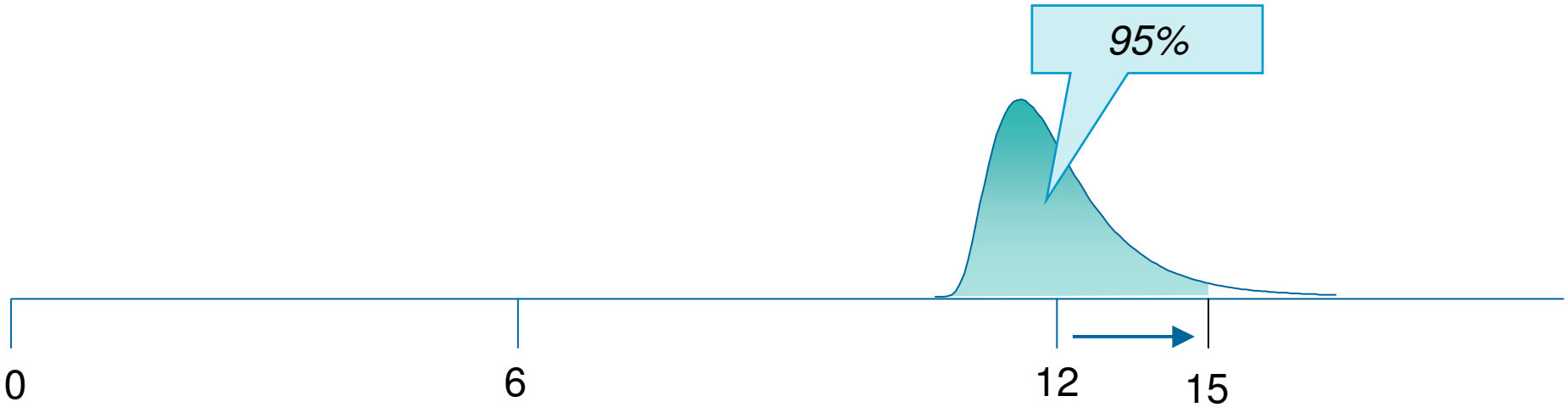
Imagine you have 12 months to deliver a business critical systems

- In fact there is less than 50% chance of making the date



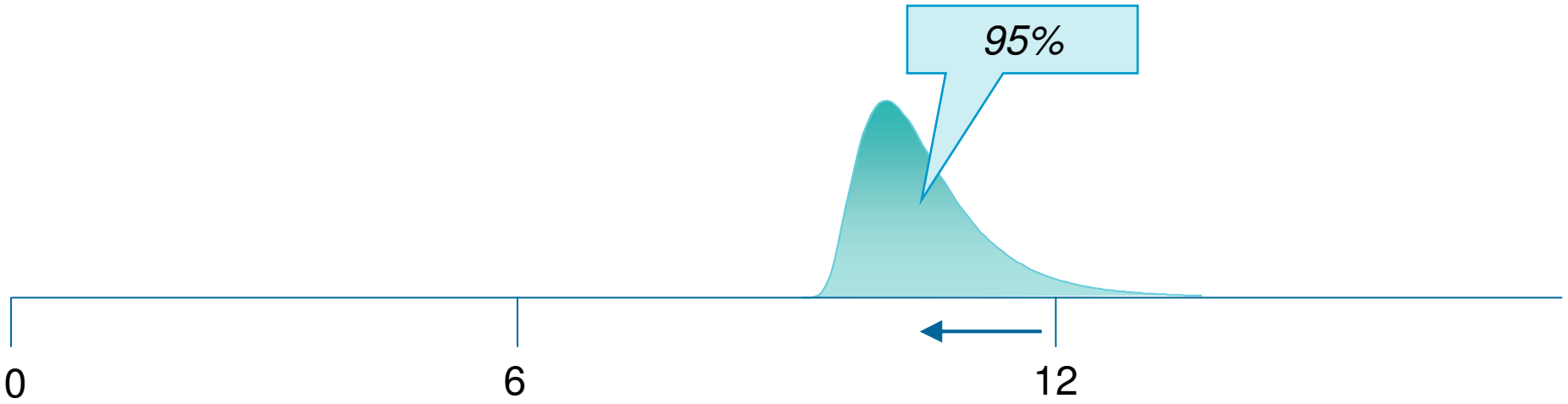
Then what?

- Move out the date to improve likelihood of shipping?



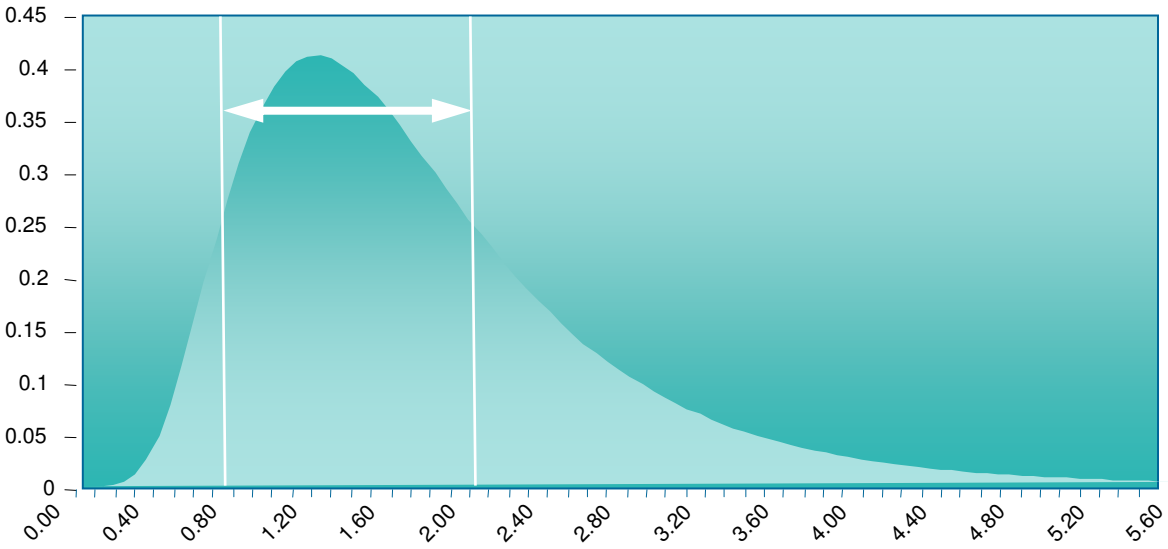
Then what?

- Or move in the estimate by sacrificing quality or content?



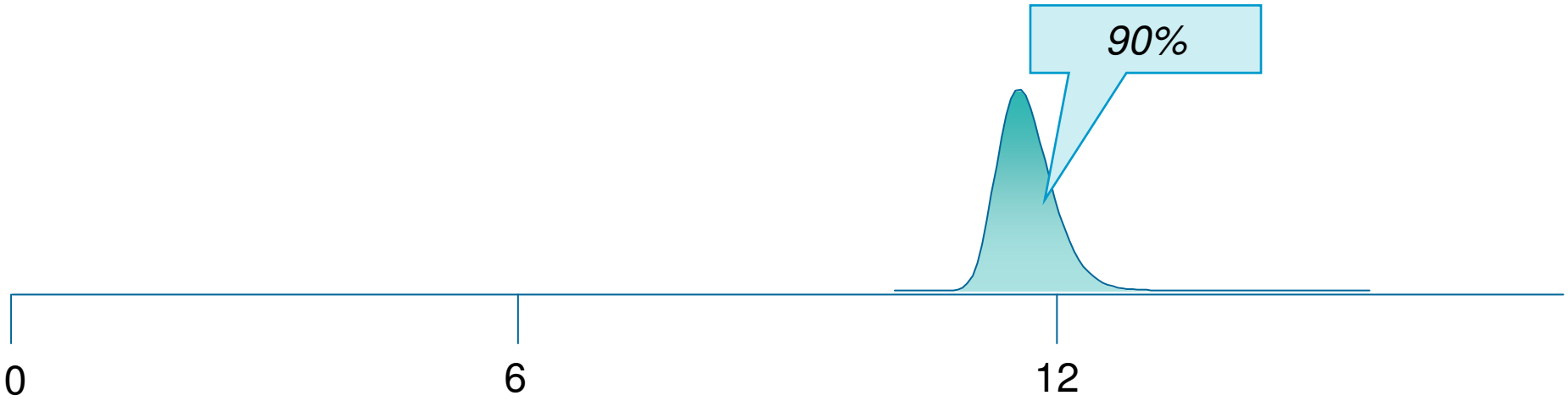
Managing variances in scope, solution, plans: The real key to improving software economics

- Sources of uncertainty and variance
  - Lack of knowledge
  - Lack of confidence
  - Lack of agreement
- Reduction of variance reflects
  - Increased predictability of outcome
  - Increased knowledge about
    - Client needs
    - Technology capability
    - Team capability
  - Good decisions



Then what?

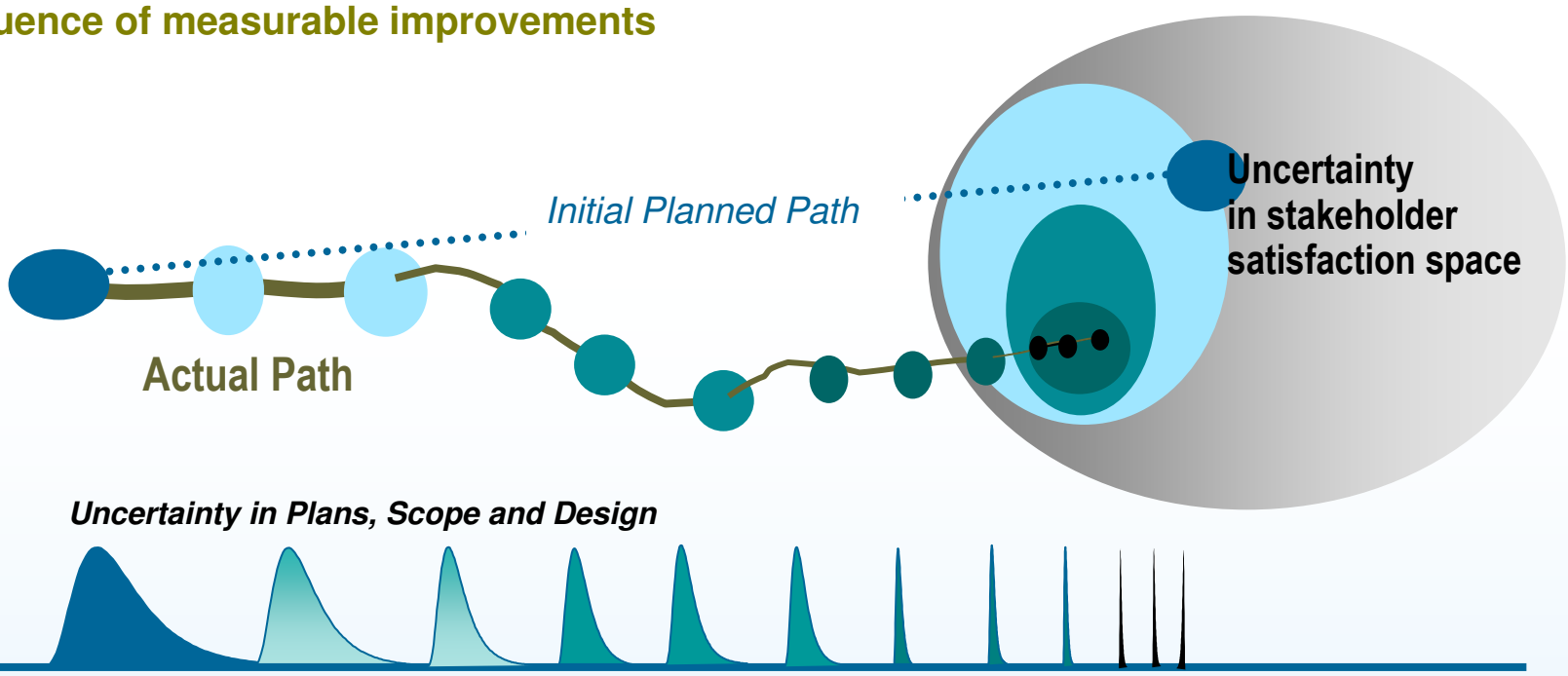
- Determine the source of the variance
- Over the project lifecycle, reduce the variance to improve likelihood of shipping



Then what?

- Over the lifecycle, reduce the variance further to improve likelihood of shipping

Economic Governance: Measurement and Steering

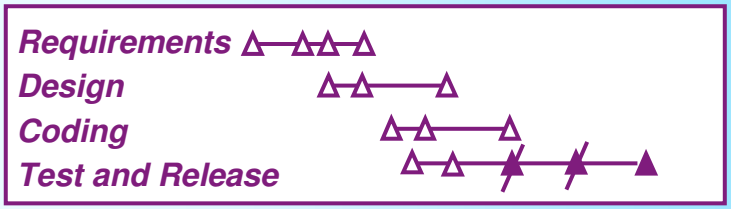
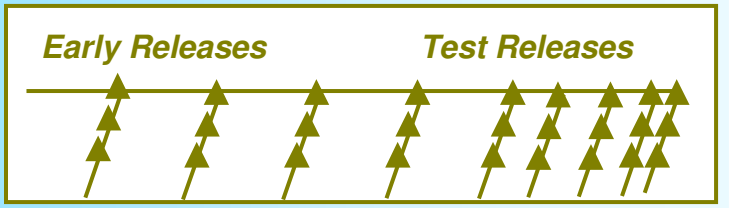
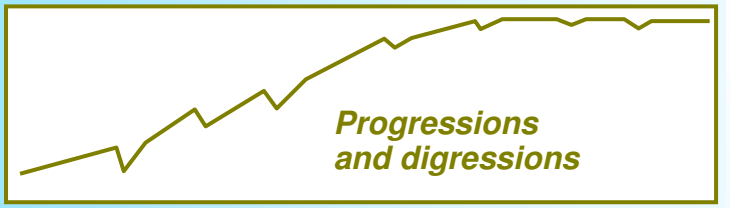
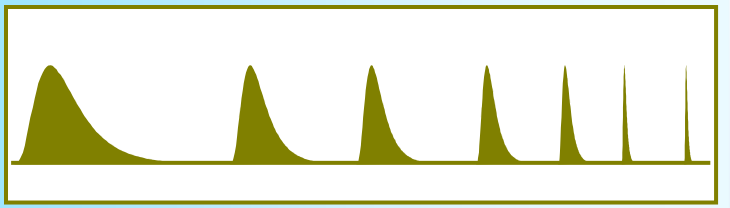


Managing uncertainty requires **MEASUREMENT** → *Measurement builds* **TRUST** → *Trust improves* **EFFICIENCY**

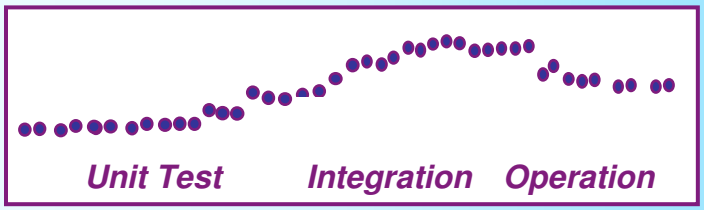
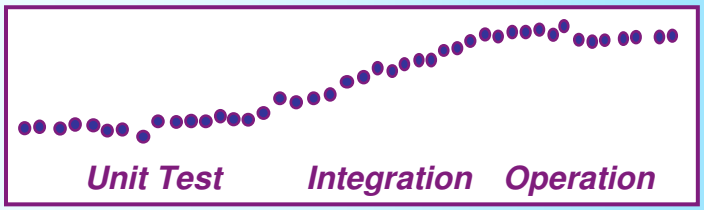
Pivotal Culture Shifts

Integrate	Collaborate	Optimize
Plans/management Plan for integration to precede unit testing	**Progress measures** Quantify progress trends from the integrated code and test base	**Quality measures** Quantify cost-of-change trends to demonstrate true agility
Avoid false precision in plans and requirements	**Don't** rely on subjective and speculative measures	**Don't** attack the easy things first

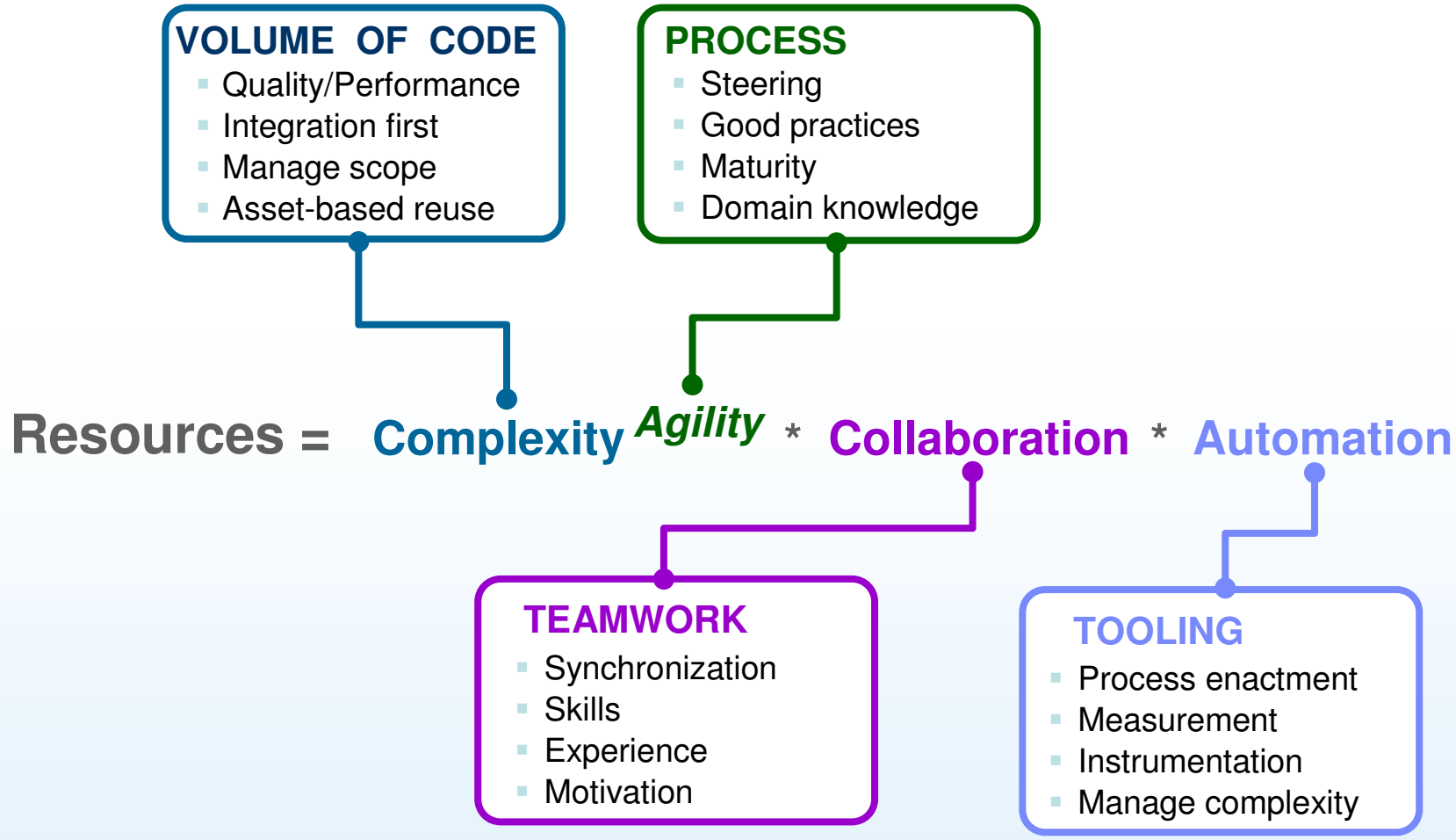
Measured Improvement: *Progress Econometrics*

Conventional Engineering Governance		Modern Economic Governance
	**Planning Progress**	
	**Technical Progress**	
	**Economic Progress**	

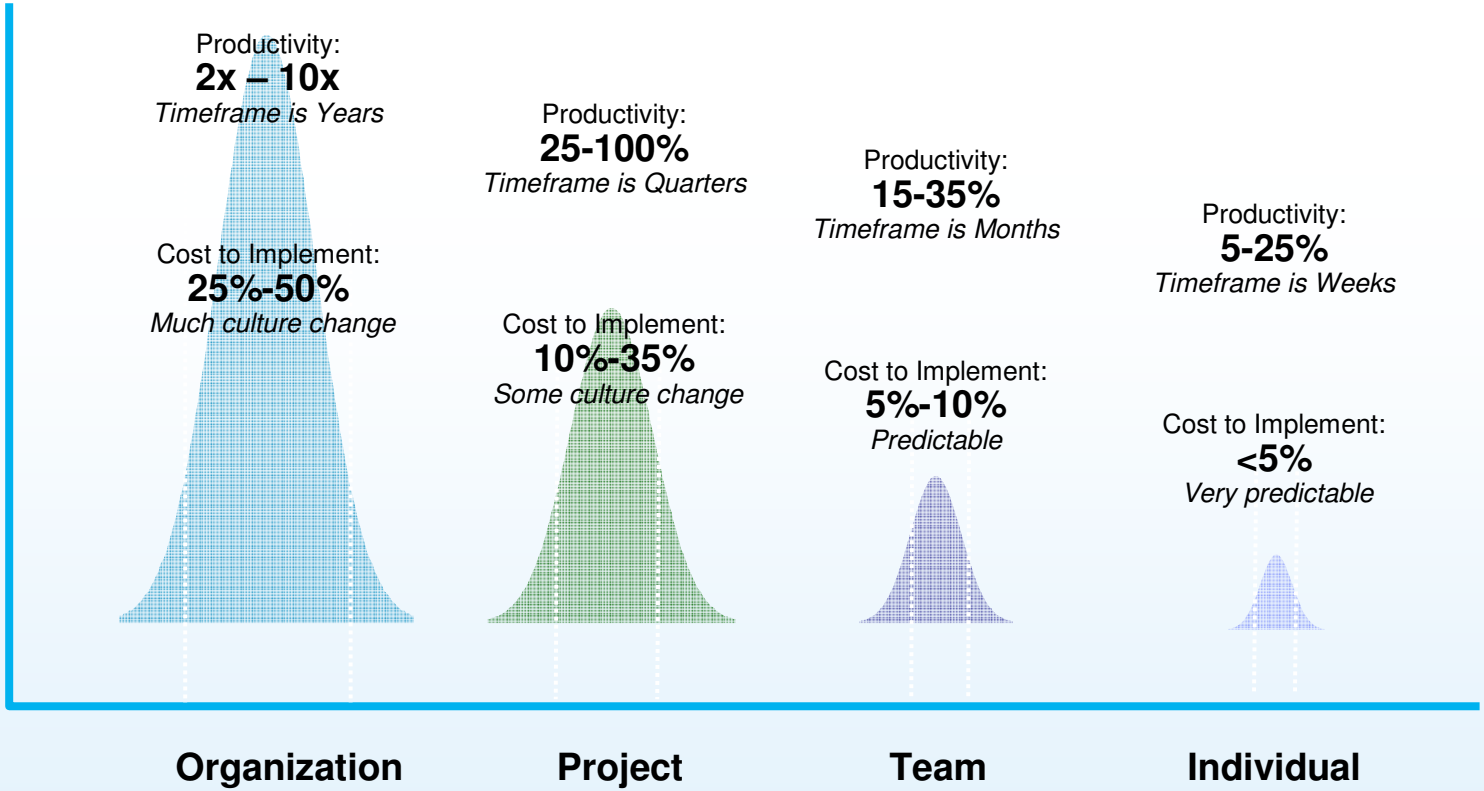
Measured Improvement: *Quality Econometrics*

Conventional Engineering Governance		Modern Economic Governance
	**Maturity** *Defect Trend*	
	**Modularity** *Change Volume Trend*	
	**Adaptability** *Cost of Change Trend*	

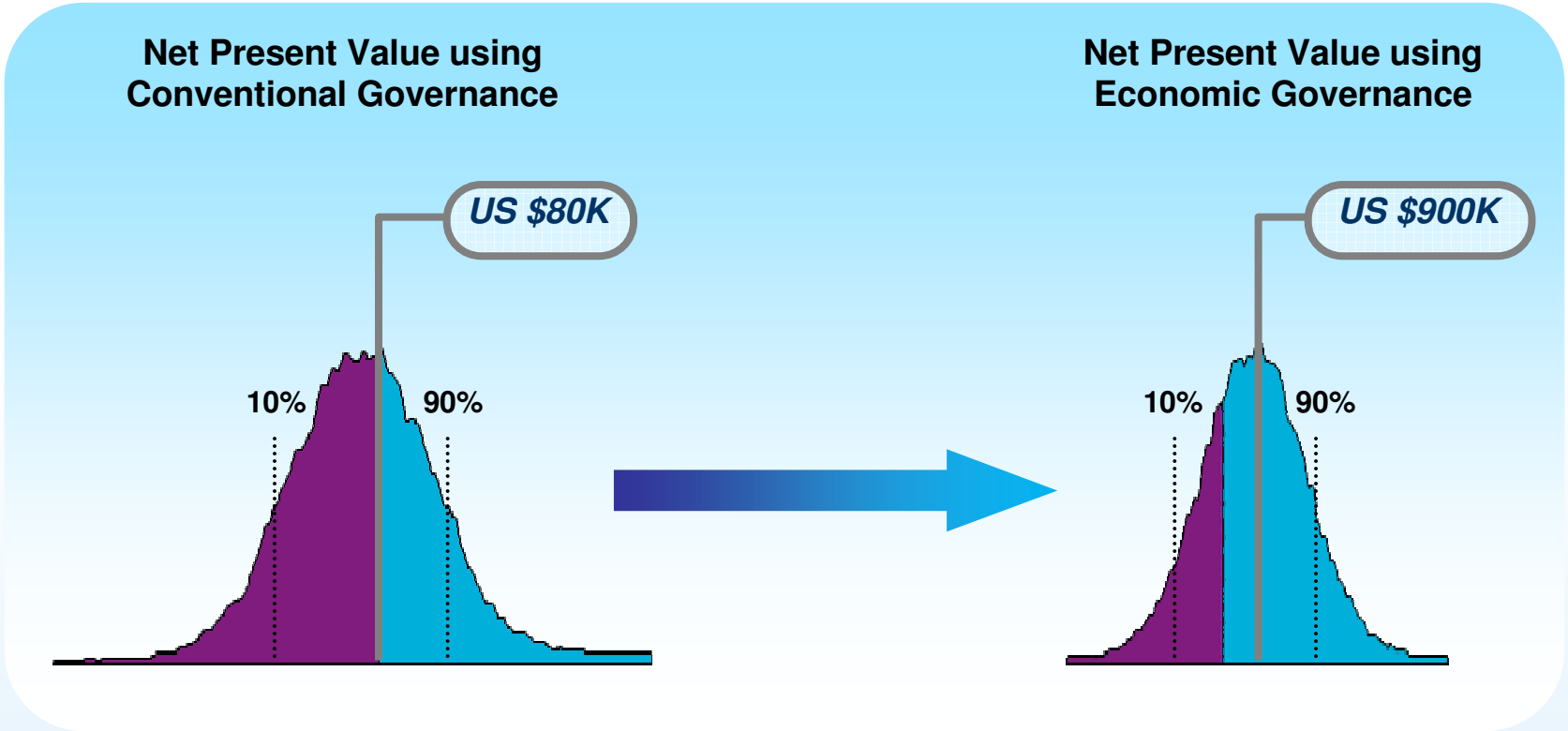
Improving Software Economics



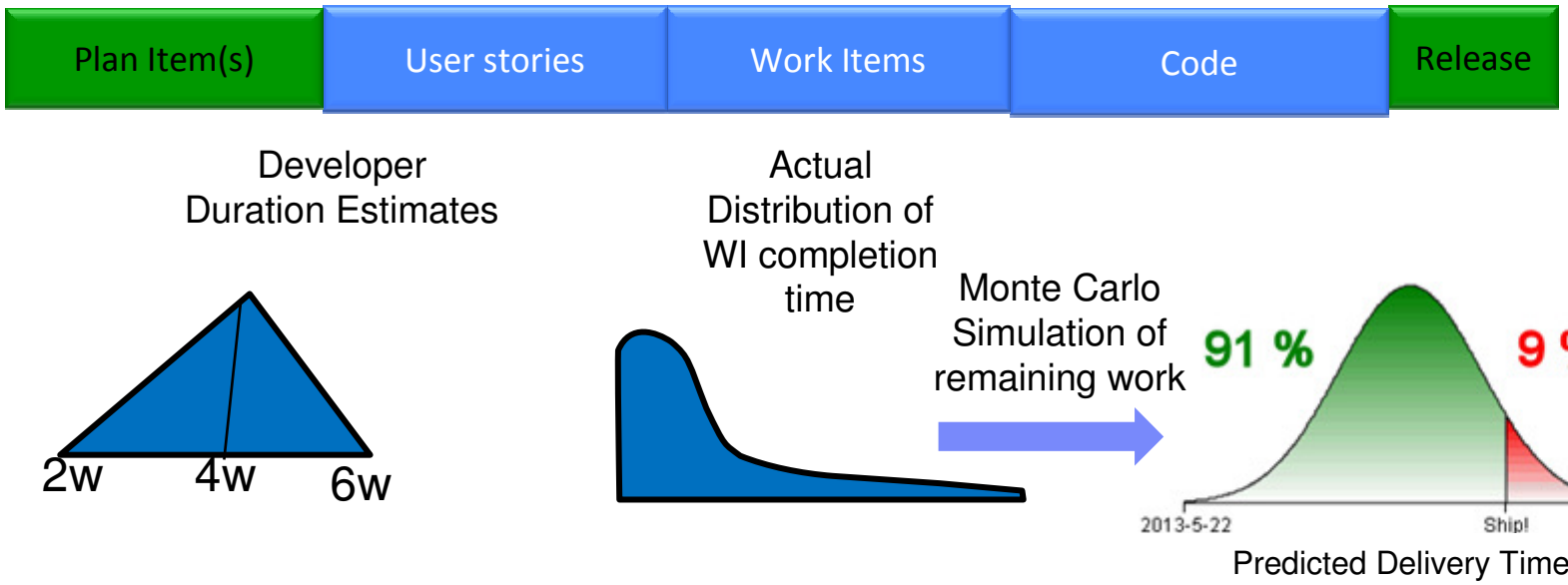
Productivity Improvement Leverage



Measured Improvement: *Quality Econometrics*



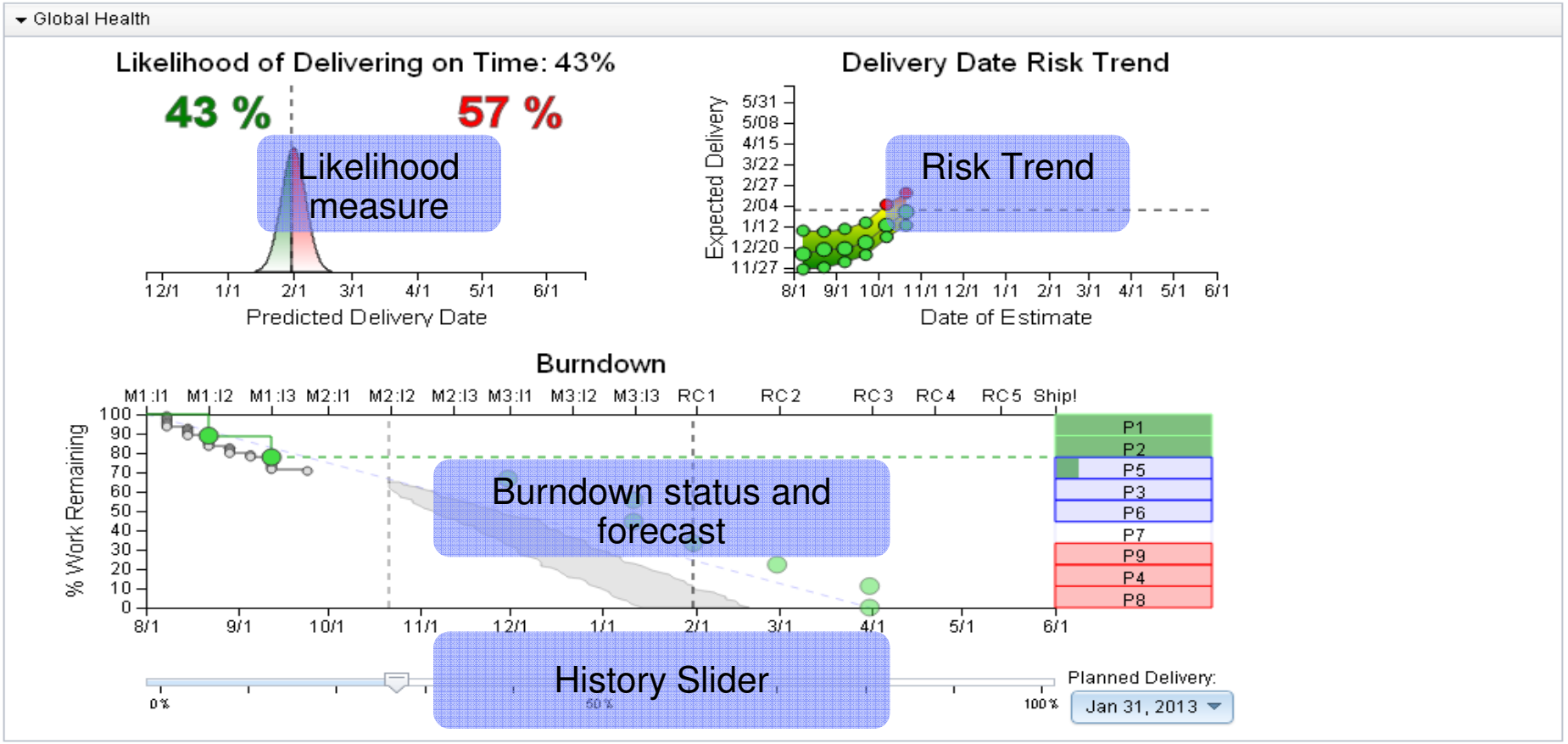
AnDeS is a Monte Carlo simulation to lifecycle of project scope items



Uncertainty is reduced as

- Higher risk work is completed
- Plan items are scoped as work items

The Andes Solution



The Moral of This Story

Better software economics is a result of:

1. Measured improvement for improved predictability

- The foundation of economic governance
- Measurement helps you manage uncertainty

2. Agility for improved operational efficiency

- Best measured by cost of change trends
- Best achieved by accelerating integration testing

If you play better defense you can play more offense!