

Sailing in cross currents:
setting your course amidst the
many current approaches to
requirements

Daniel Moul & Keith Collyer

Requirements Product Delivery Team

dmoul@us.ibm.com & keith.collyer@uk.ibm.com

RDM 2023B

Innovate2010

The Rational Software Conference

Let's **build** a smarter planet.

The premiere software and product delivery event.

June 6–10 Orlando, Florida



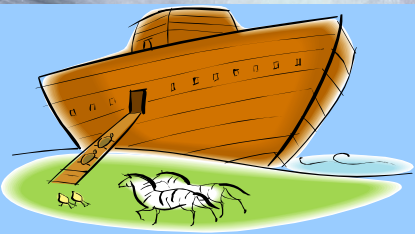
Abstract

Web development? Manufactured products? Packaged applications? SOA? Agile? System requirements specifications? Use cases? User stories? Faced with the many competing types of projects and approaches to creating and managing requirements, how can you determine which are right for your organization?

This session will survey major approaches, provide a framework to help you assess them, and offer some keys to successful implementation based on years of IBM Rational and Telelogic experience.



In your current project ... which ship are you sailing?



On terminology

- For today let's consider all of these "RM"
 - ▶ Requirements Management
 - ▶ Requirements Engineering
 - ▶ Requirements Definition

- Sometime we'll use "practices" to mean "process"



Why do you need a requirements process? *Isn't a requirements tool enough?*

- Effective RM is about people and process
- Process is whole-team behavior
- Tools enable and accelerate process



The Business Environment Determines Project Parameters

- Are you the customer or contractor?
- What are the financial terms?
- Who is responsible for what?



Projects are commissioned to meet business goals

Program & Portfolio Management

Business strategy
Objectives & priorities
Business cases
Project proposals

Project selection
Project charters

Vision documents
Requirements identified
Project execution

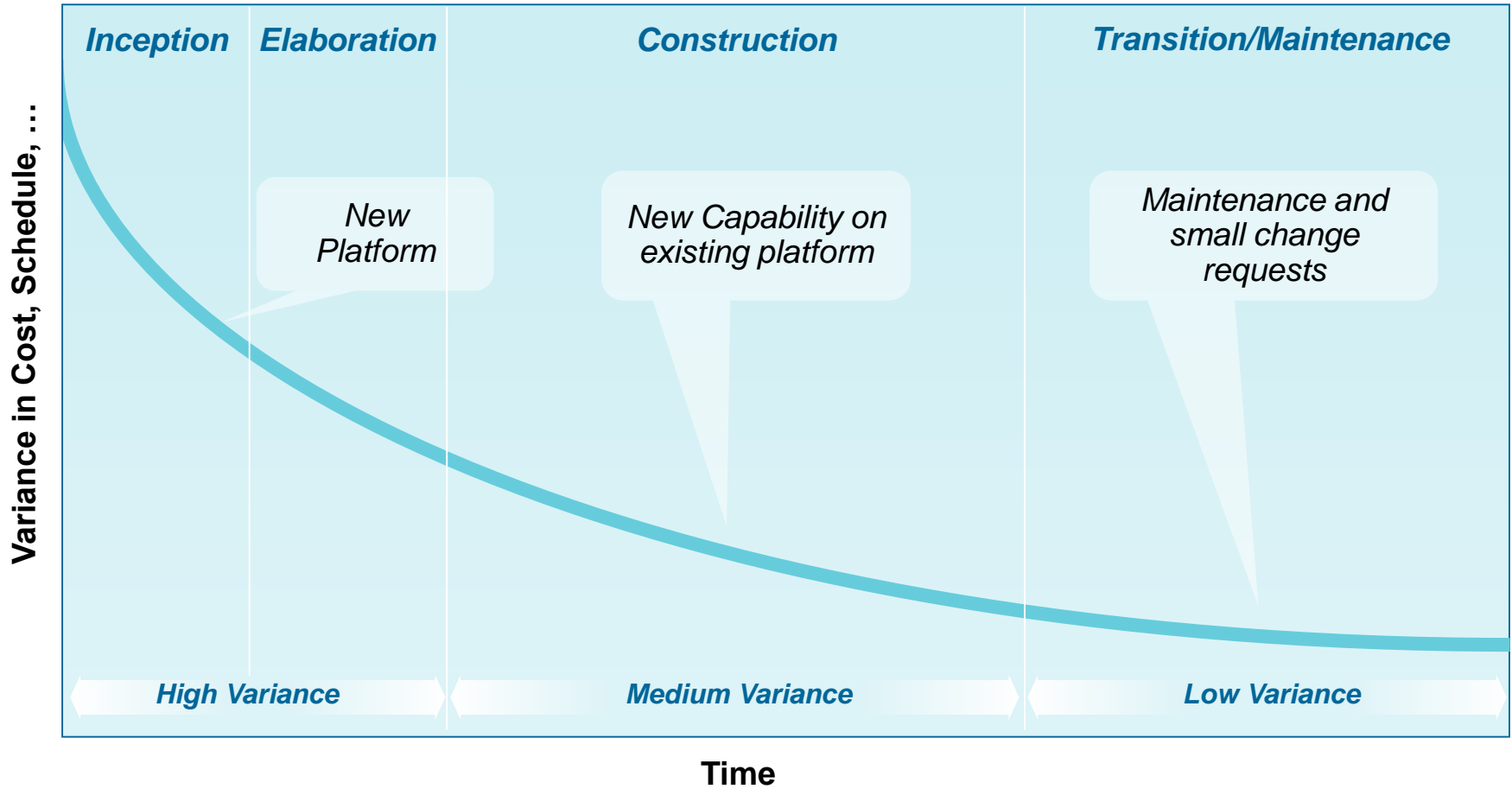


Project Management

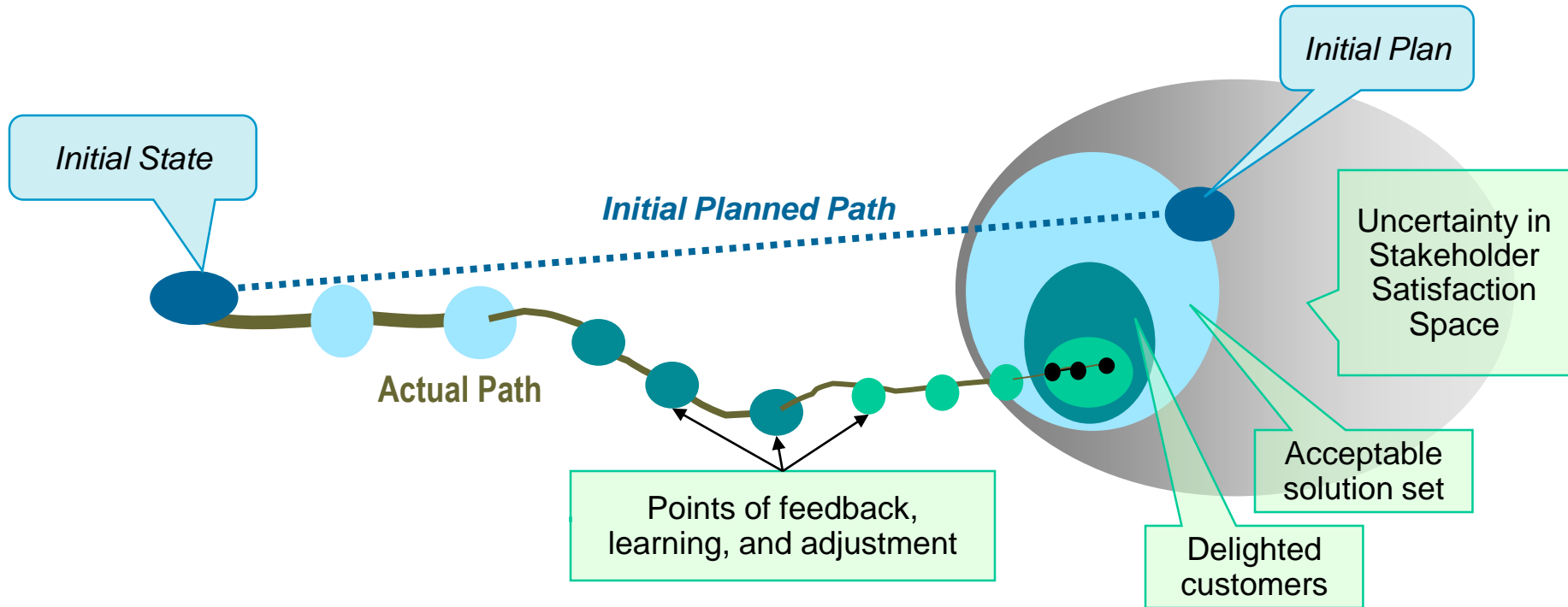


Different projects need different governance

Uncertainty and risk are the key discriminators



Project delivery is an exercise in removing uncertainty



Feedback → course corrections → better outcomes

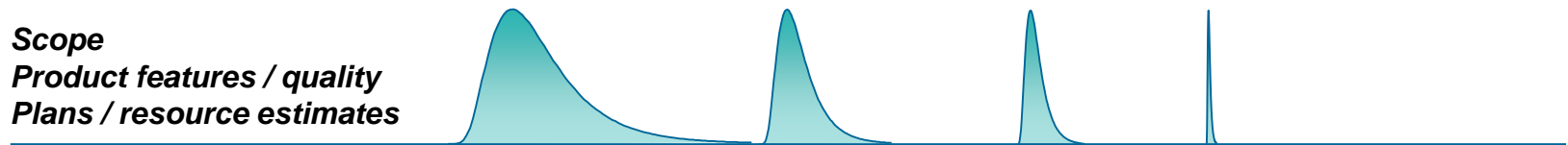


Managing Uncertainty = Managing Variance

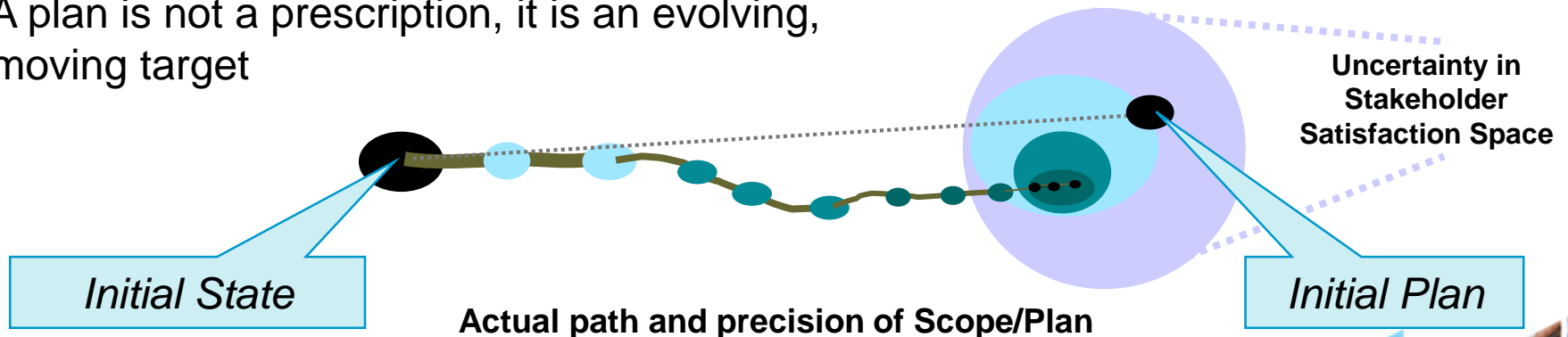
- A completion date is not a point in time, it is a probability distribution



- Scope is not a requirements document, it is a continuous negotiation

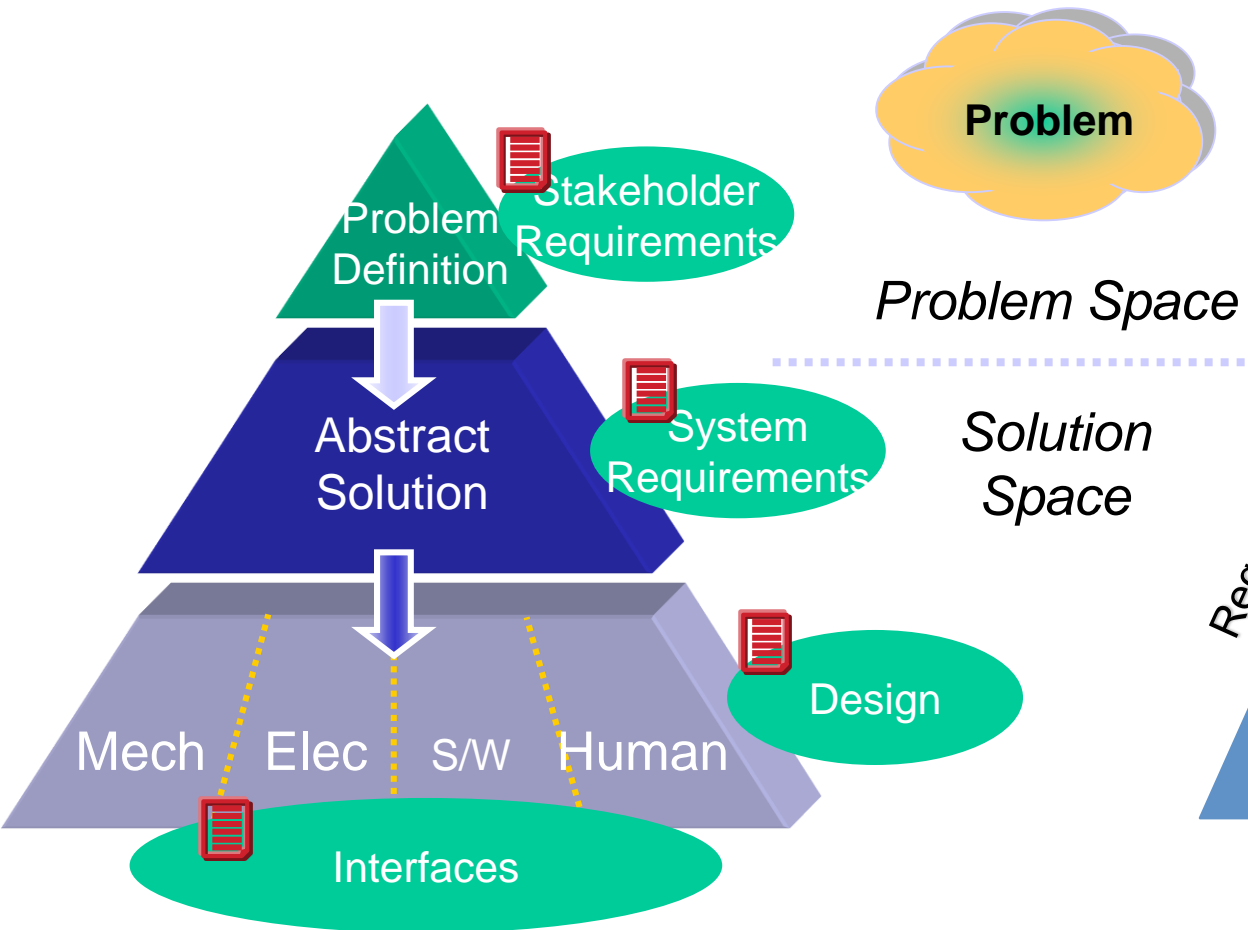


- A plan is not a prescription, it is an evolving, moving target

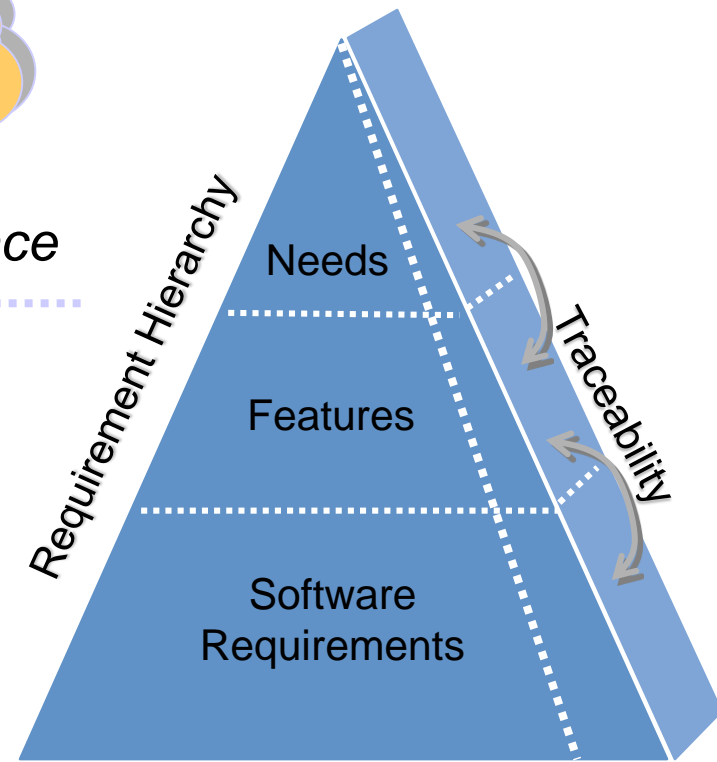


Typical Requirements Hierarchy – two views

Typical Systems View



Typical IT View



Not every requirement is called a requirement

Gaps in
Packaged
Applications



Capability
Gap
(military)



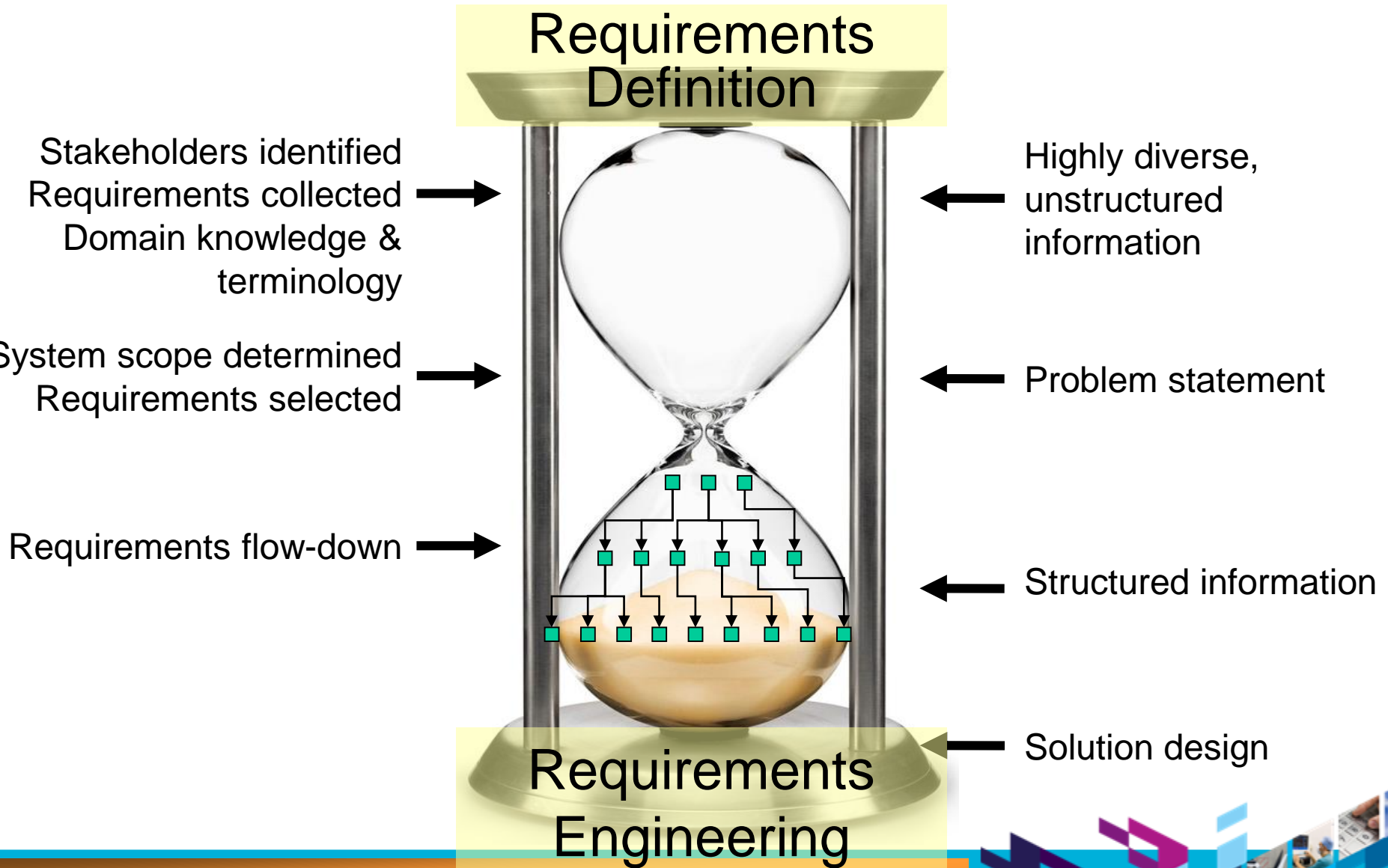
Regulations
Policies
Standards



Business
Rules



Progressive removal of uncertainty

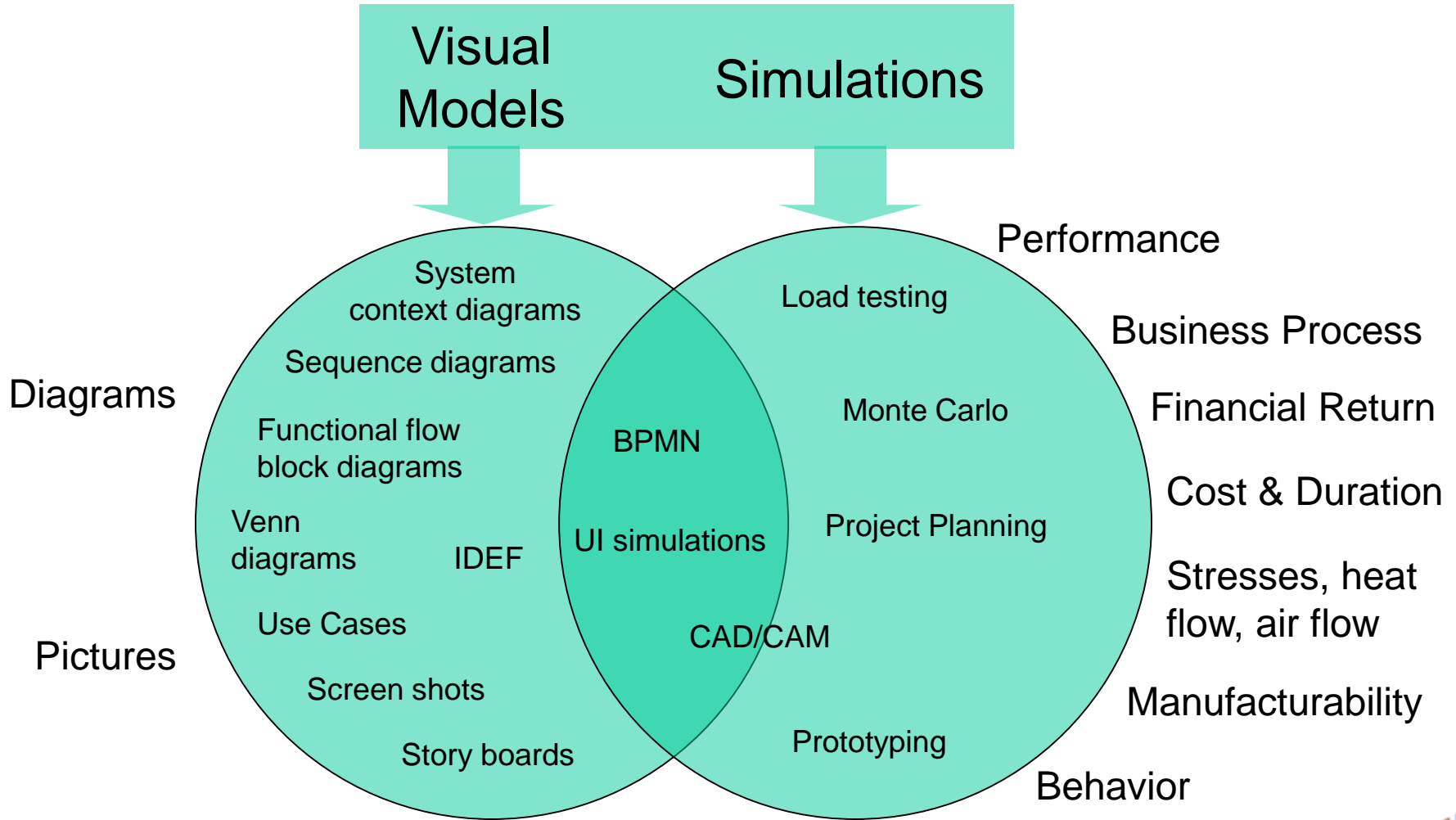


Like sand in an hourglass, this can be a continuous process



Models: Low-cost ways to learn early

Optimize for learning / adjusting



Economics determine many of the possible optimizations

Optimize for learning and adjusting as much as possible within your constraints

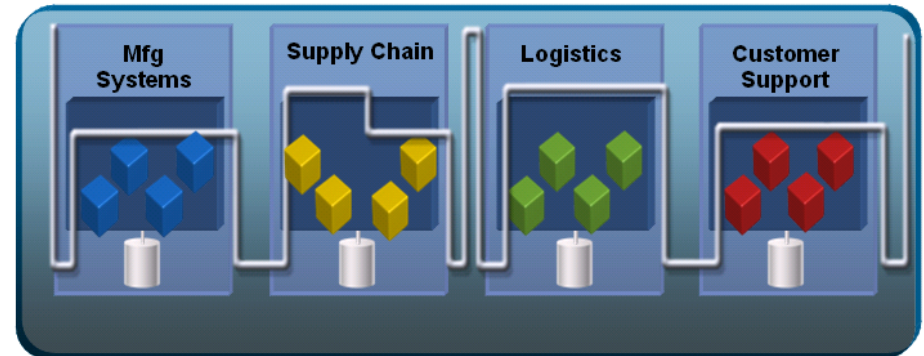
- What can you optimize?
- What are the underlying economics?
 - Value of being fast-to-market
 - Cost of failure
 - Cost of change
 - Cost of communication
 - Cost of non-compliance
 - Cost of design and manufacture



Industry patterns reveal optimized groupings

- Unique industry characteristics
 - ▶ Government / private
 - ▶ Manufactured systems / IT

- Patterns are reflected in culture



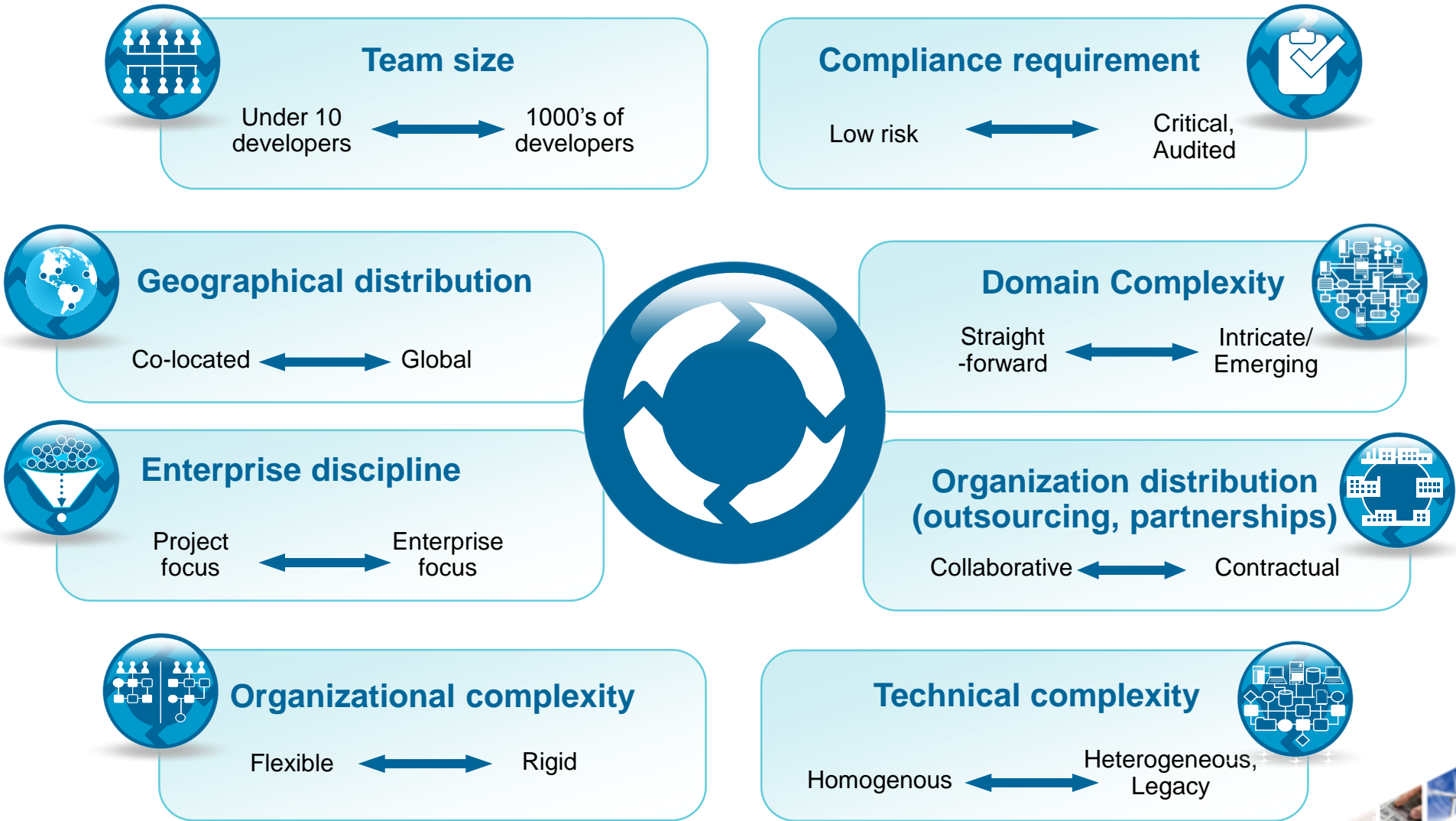
Project have various cultures

Group	Requirements Focus
<p>Engineering & Compliance culture <i>Good outcomes are the result of good, controlled processes. “Have we missed anything?”</i></p>	<p>RM in an engineering process</p> <ul style="list-style-type: none"> • Formal process • Manufactured systems • Mission-critical systems • Regulated, compliance, and contract-driven industries
<p>Market-driven culture <i>Balance process and expedience. “How can we get this out faster with good quality?”</i></p>	<p>Effective teams, efficient tools</p> <ul style="list-style-type: none"> • Business-oriented software applications • Fast-to-market manufacturers
<p>ALM minimalist culture <i>“We use our main tools for requirements too”</i></p>	<p>Use development and test tools</p> <ul style="list-style-type: none"> • Requirements by and for dev and test • Typically business analysts are not involved
<p>Ad-hoc culture <i>“What is RM?”</i> <i>“We don’t do RM”</i> <i>“We get by with office docs”</i></p>	<p>Using general-purpose tools at hand</p> <ul style="list-style-type: none"> • May employ some RM, “pure agile” methodologies or no defined methodology at all

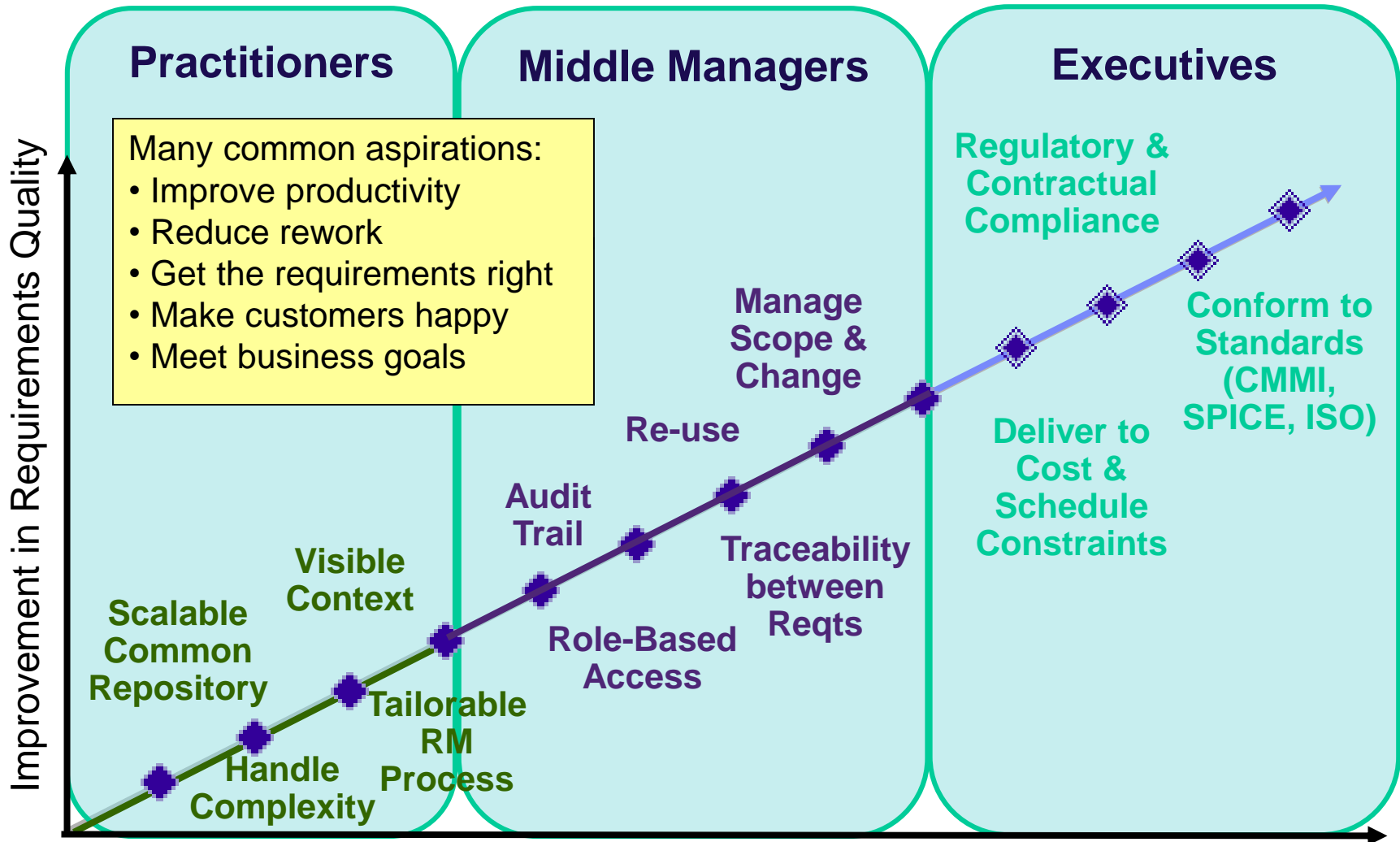
Management Techniques in Organization Culture (extract)

Base Lifecycle	Technique	Engineering - Compliance	Market-driven	ALM minimalist	Ad hoc
Waterfall	BDFU	Y			
	Complete stages	Y			
	Single release	Y			
	Complete plan	Y			Plan, what plan?
Incremental	BDFU	Y			
	Parallel implementation	Y	Y	Y	
	Early release		Y	Y	
Iterative / Evolutionary	Large-scale iteration	Y	Y		
	Plan complete at high level	Y	Y		
	Cost-benefit driven	Y	Y	Y	
	Value at every release		Y	Y	
	Each release "complete"		Y	Y	
	Small releases		?	Y	Y
	Just enough planning		Y	Y	
Agile	User stories		Y	Y	Y
	Time-boxing		Y	Y	
	Test-driven		Y	Y	
	Work-item list		Y	Y	Y
Common	Evaluate against plan	Y	Y	Y	
	Compliance-checking	Y	?		
	Feedback	Y	Y	Y	Y
	Consistency check	Y	Y	Y	

Consider other constraints – Example: IBM agility@scale™



Value to stakeholders should determine RM priorities



The RM Vision Iceberg

Buying RM Tool

Process Definition

Enterprise Tool Deployment

Tiered Support

Tailoring

Promoting Adoption

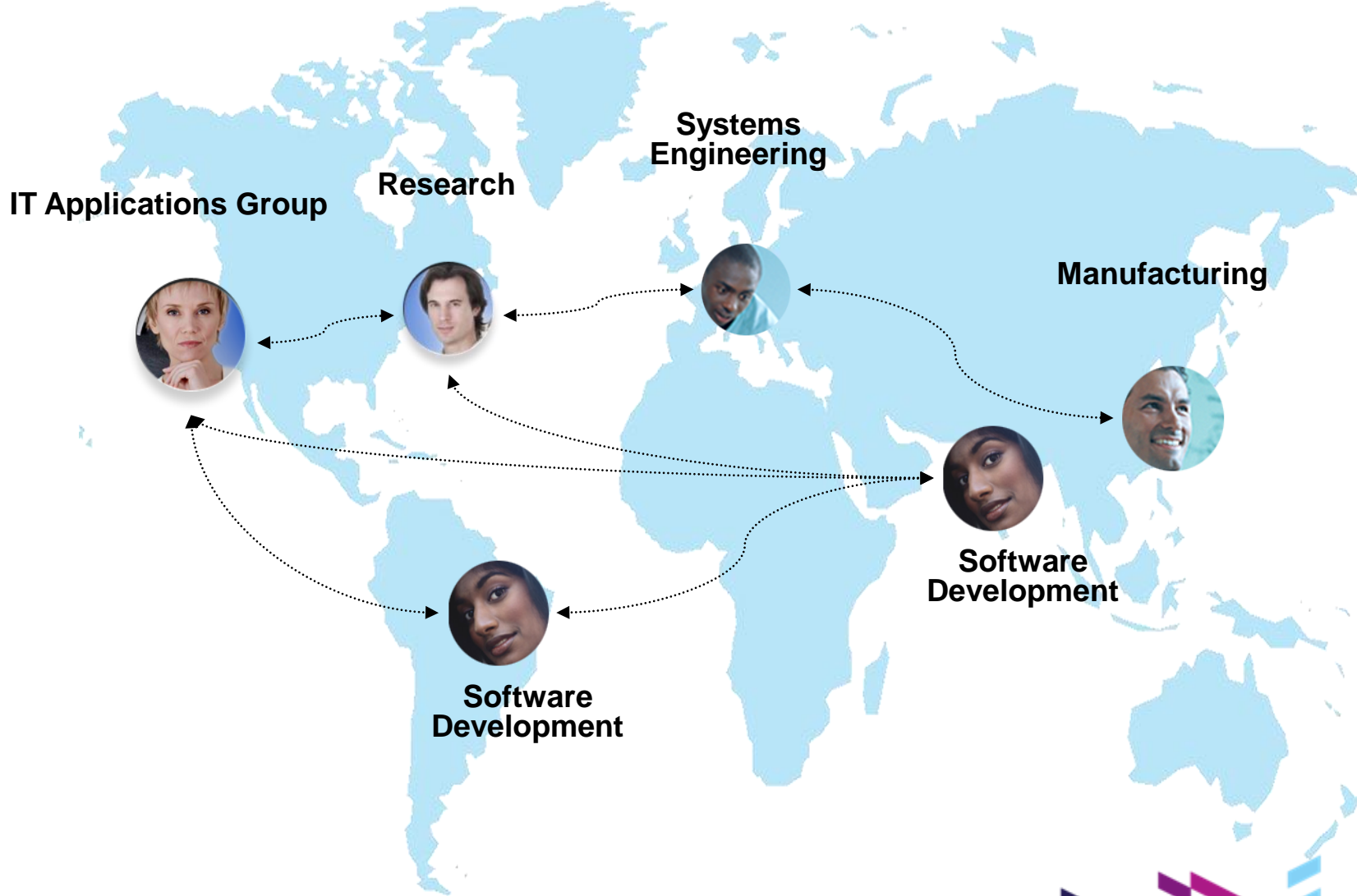
Training

Metrics



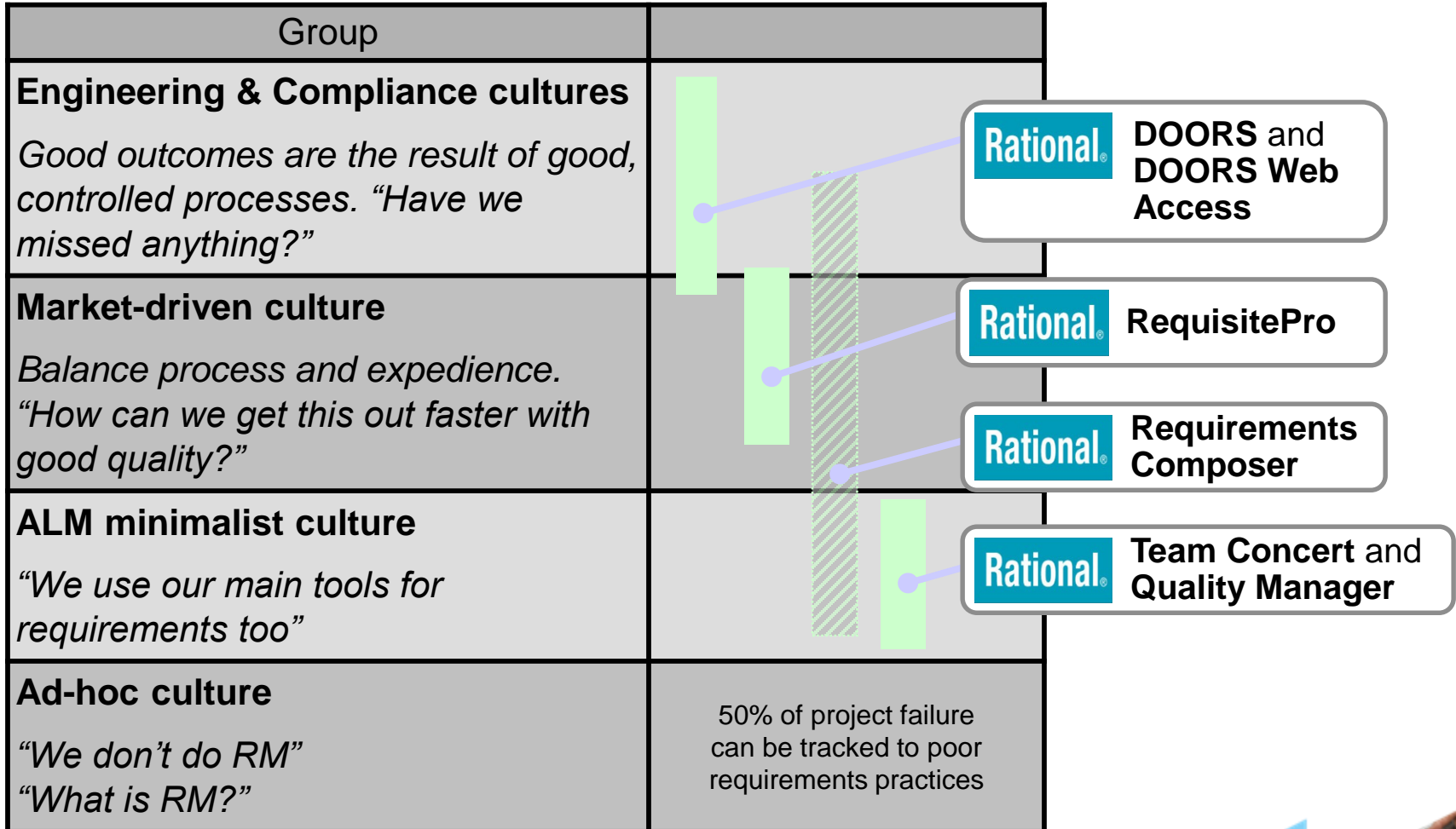
People impose significant constraints

And in one company there are many project team cultures



Rational RM portfolio today

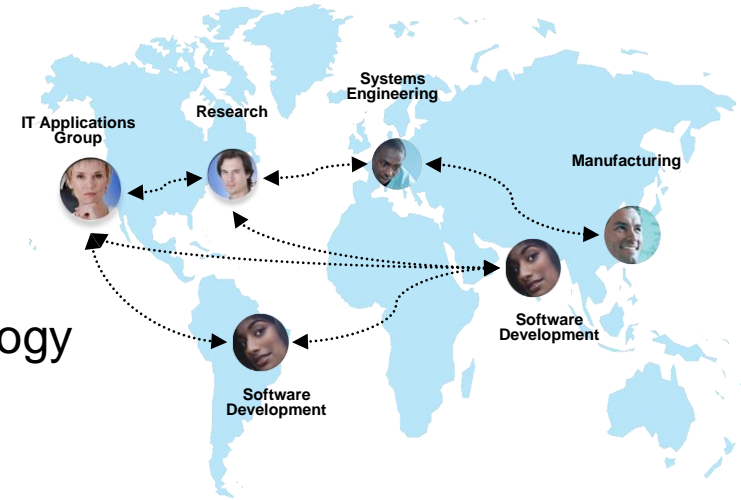
Addressing different cultures and different needs



Summary

An effective requirements process will ...

- Fit the business environment and project methodology
- Recognize where there is scope for optimization (and where there isn't)
- Recognize (and reduce) the degree of uncertainty in the solution throughout the project lifecycle
- Be adaptable ... not "one size fits all teams"
- Be well communicated, understood, and followed
- Be relevant to the entire project lifecycle
- Include measurement, reflection and continuous improvement
- Be supported by (embodied in) tools



Questions





www.ibm/software/rational

© Copyright IBM Corporation 2010. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

