IBM Software Group

Improving Software Economics and Getting to Painless Governance

Walker Royce

Vice President, Rational Worldwide Services

Rational. software Software

e business on demand software

Improving Systems/Software Governance

	1960s-1980s	1990s-2000s	2005+
Complexity	100% Custom	30% Reused Assets 70% Custom	70% Reused Assets 30% Custom
Process	Ad-hoc	Repeatable	Managed and Measured
Team	Collocated OJT	Collocated Software Skills	Distributed Systems/Software Professionals
Tools	Proprietary Not Integrated	Mix of Proprietary and Commercial Not Integrated	Commercial Integrated Processes-Tools
Project Performance	Predictable *over budget, over schedule*	Unpredictable *Infrequently on budget, on schedule*	Predictable *Frequently on budget, on schedule*
Success Rate	10%	25%-33%	50% +

Transitioning from the Old Way to the New

Conventional Governance

Activity-based management
- Mature processes, PMI/PMBOK
- Plan in detail, then track variances

Adversarial relationships
- Paper exchange, speculation

Requirements first
- Assumes certainty in desired product
- Avoid change

Early false precision
- "More detail = higher quality"

Apply too much or too little process
- Process is primary, blind adherence

Modern Governance

Results-based management
- More art than engineering
- Plan/steer/plan/steer….

Honest collaborative communication
- Progressions/digressions, facts

Architecture (*risk mitigation*) first
- Admits uncertainties
- Manage change

Evolving artifacts
- Scope (Problem specs)
- Design (Solution specs)
- Constraints (Planning specs)

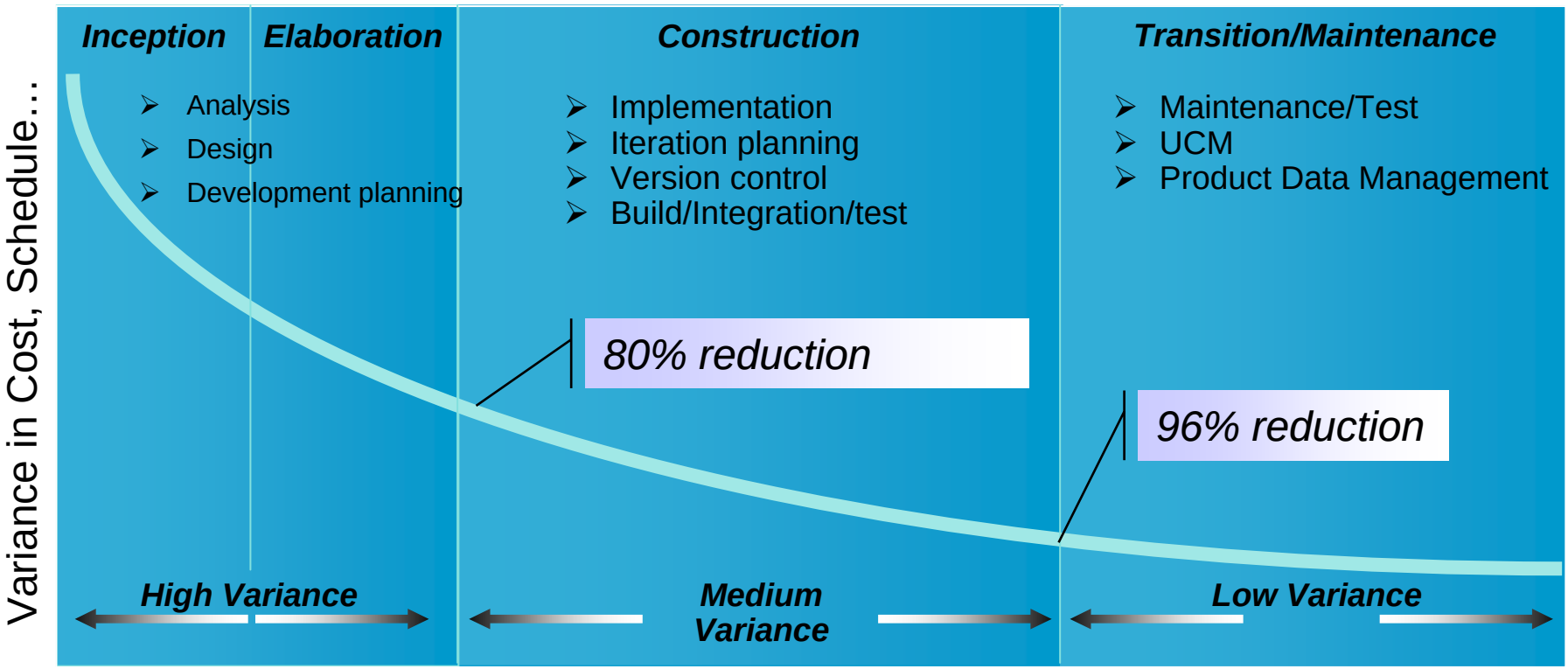
Right-size the process
- Desired results drive process

Development Governance

Agile
"Steering"

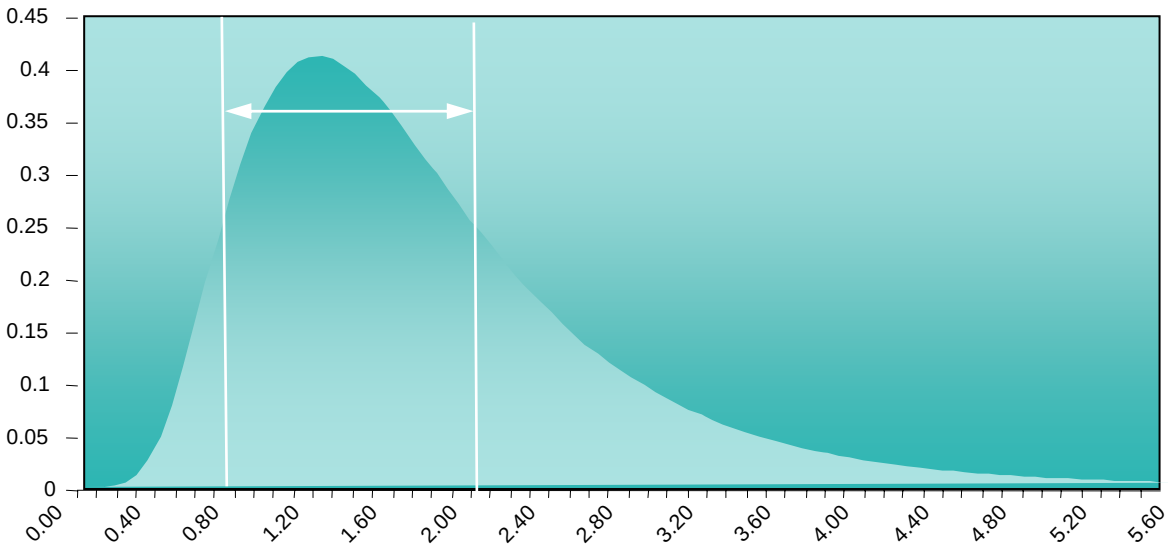
Methodical
"Production"

Rigorous
"Quality Controlled"



Managing Variance

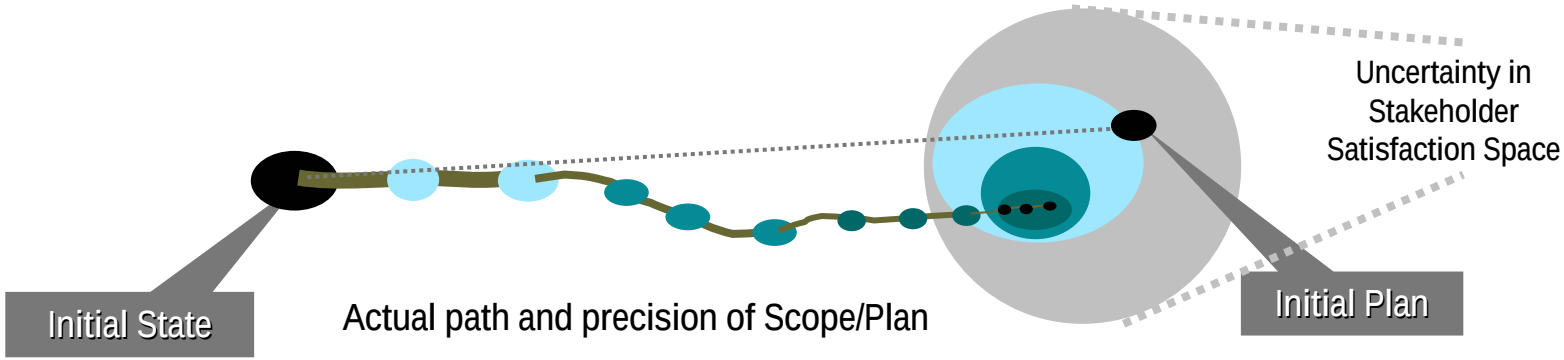
- Sources of uncertainty and variance
  - Lack of knowledge
  - Lack of confidence
  - Lack of agreement
- Reduction of variance reflects
  - Increased predictability of outcome
  - Increased knowledge about
    - Client needs
    - Technology capability
    - Team capability
  - Good Decisions



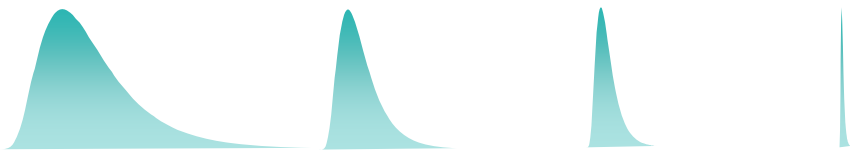
Governance = Managing Uncertainty = Managing Variance

- 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



Plans/Resource estimates
Scope
Product features/quality



Four Patterns of Success

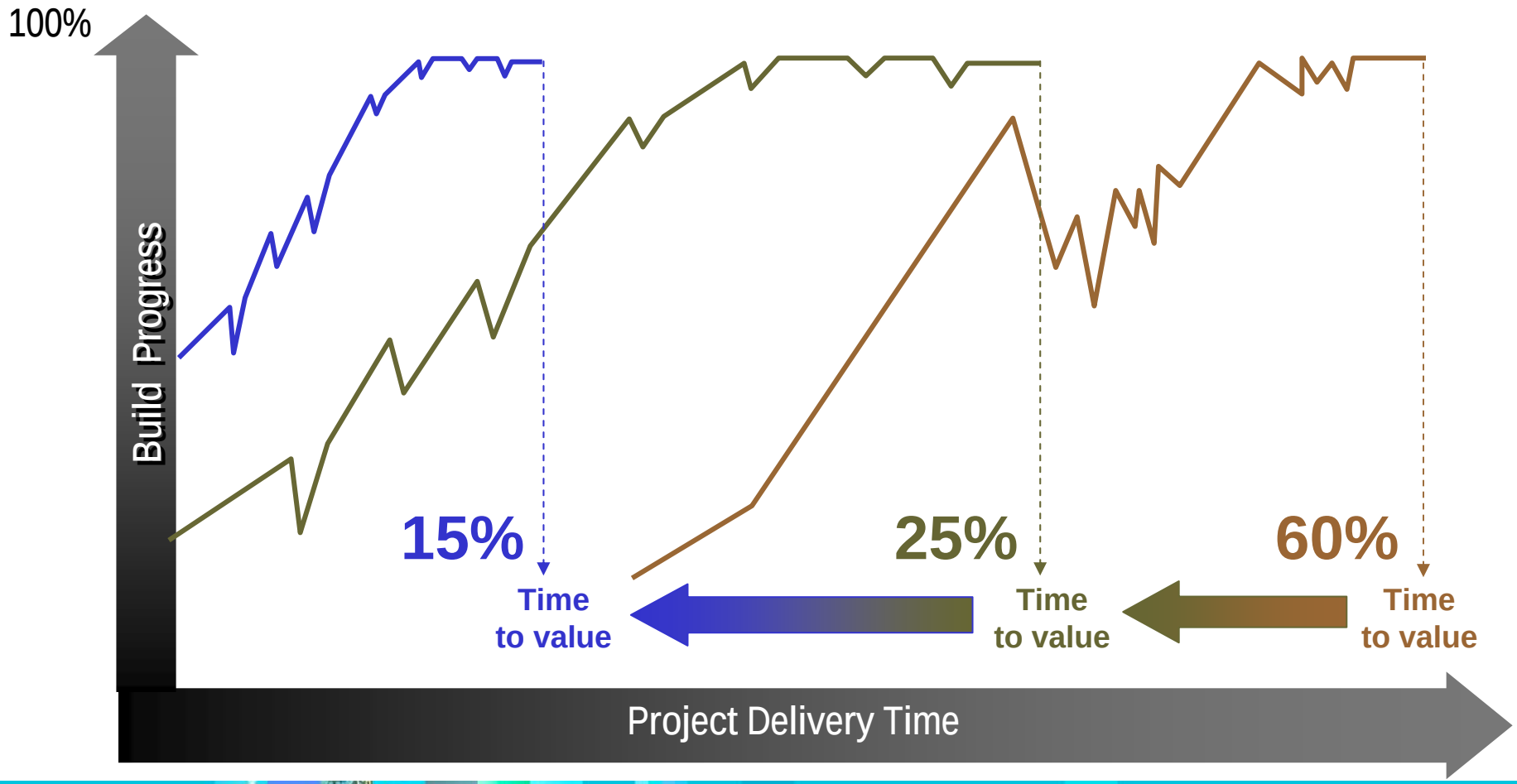
- **Scope management ➔ *Asset based development***
  - **Solutions need to evolve from user specifications AND user specifications need to evolve from candidate solutions.**
    - **As opposed to getting all the requirements right up front.**
- **Process management ➔ *Rightsize the process***
  - **Process and instrumentation rigor evolves from light to heavy.**
    - **As opposed to the entire project's lifecycle process should be light or heavy depending on the character of the project.**
- **Progress management ➔ *Honest assessments***
  - **Healthy projects display a sequence of progressions and digressions.**
    - **As opposed to healthy projects progress to 100% earned value with a monotonically increasing and predictable plan.**
- **Quality management ➔ *Incremental demonstrable results***
  - **Testing needs to be a 1st class, full lifecycle activity.**
    - **As opposed to a subordinate, later lifecycle activity.**

Improving Time to Value

Effective governance
SOAs
Integrated environments

Iterative processes
Middleware components
Mature commercial tools

Conventional processes
Stovepipe architectures
Proprietary tools/methods

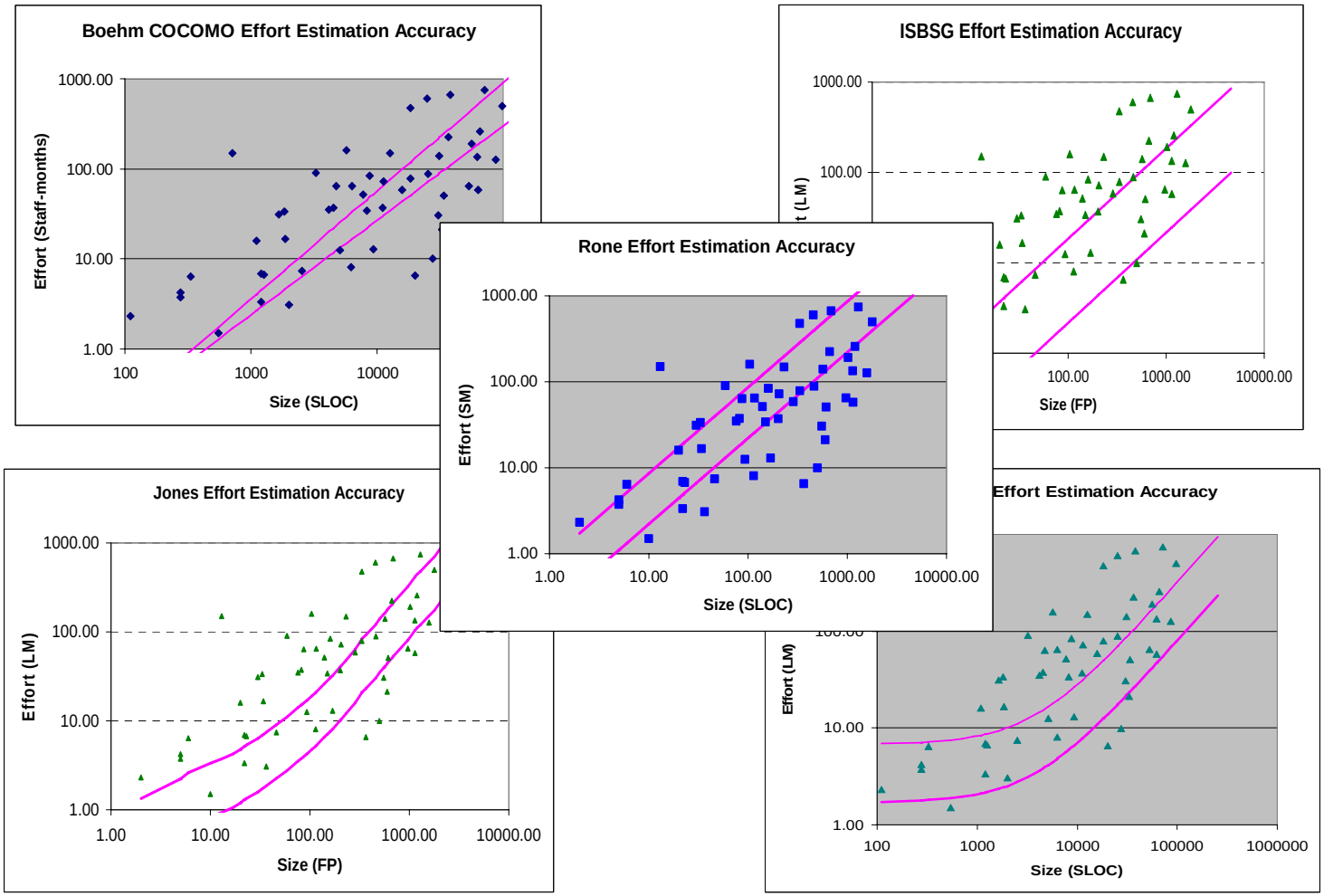


A Discriminating Macro-Level Metric: Activity Mix Trends

Workflow	Future Process	Iterative Process	Waterfall Process
Management	12%	10%	5%
Requirements	12%	10%	5%
Design	20%	15%	10%
Implementation	14%	25%	30%
Test &Assessment	18%	25%	40%
Deployment	12%	5%	5%
Environment	12%	10%	5%
	100%	100%	100%

More balance; less waste during integration and test

Software Cost Models



Improving Software Economics

Legacy system upgrades
e-business, Web applications
SW Maintenance

New Developments
New Releases
Packaged applications

$$\text{Time or Cost To Build} = (\text{Complexity})^{(\text{Process})} * (\text{Team}) * (\text{Tools})$$

Complexity → Volume of human-generated code

Process → Methods, notations, maturity

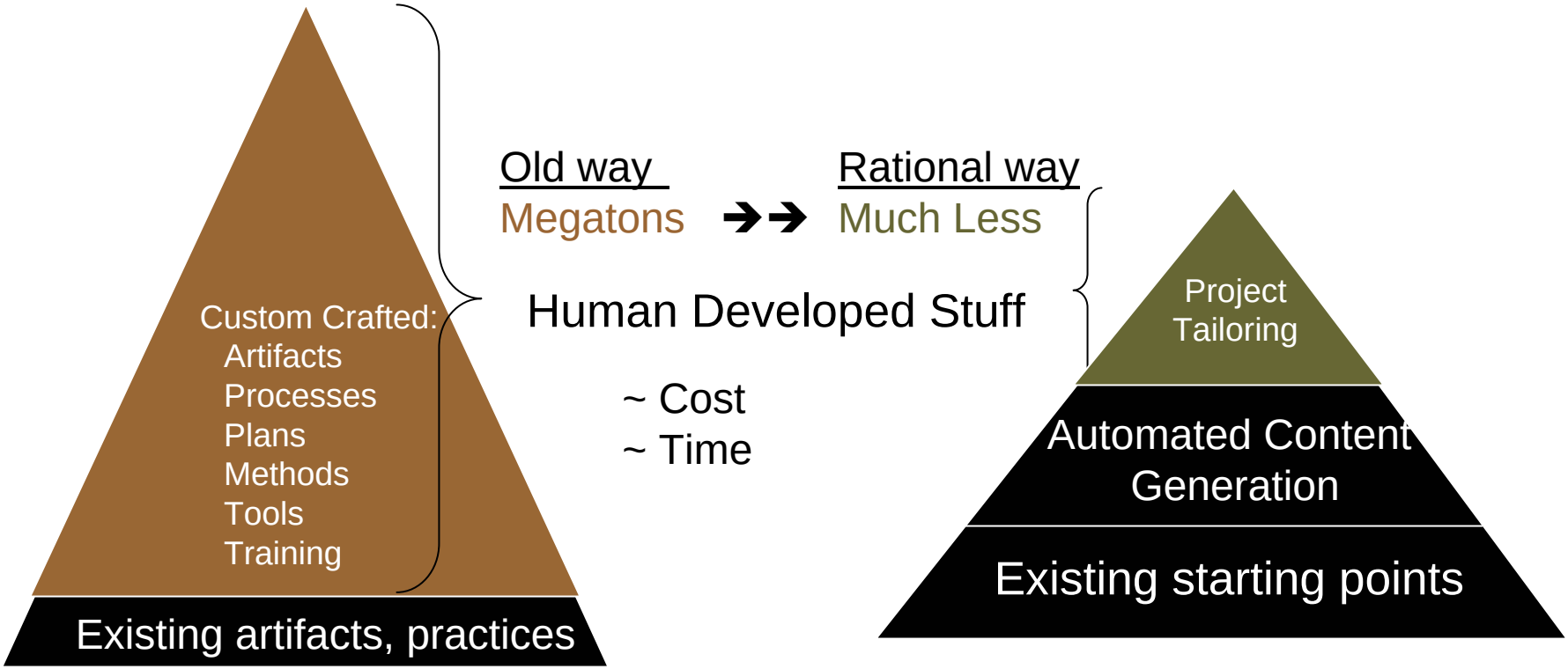
Team → Skill set, experience, motivation

Tools → Process automation

The Value of IBM/Rational's Development Platform

Business Driven Development

- **Reduced Complexity**
  - Semantic knowledge of SOA patterns, components, usage
- **Improved Process**
  - Governance, guidance and variance management
- **More Efficient Teams**
  - Accelerated proficiency, more engineering, less overhead
- **More Automation**
  - Content generation, Instrumentation, change management



Measured capability improvement

Map business value to software delivery best practices

Target: Phase 1

Already implemented

Outside scope

Example: Financial Service Company

Customer Business Challenges

- Create financial products more quickly
- Functionality of customer web falling behind competition
- Inconsistencies with integrated financial reporting
- Recent SOX audit failure

Operational Objectives

- Reduce time-to-market
- Improve productivity
- Increase innovation
- Improve consistency/predictability
- Improve oversight
- Enable flexible/global resourcing
- Satisfy compliance mandate

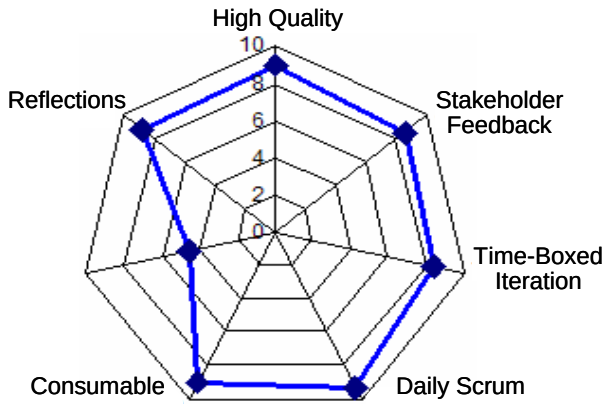
Software Delivery Best Practice

- Use-case driven development
- Continuous integration
- Shared vision
- Whole team
- Staged integration
- Multi-team management
- Risk-value lifecycle
- Asset-based development
- Asset governance
- Iterative development
- SOA modeling
- Enterprise SOA
- SOA governance
- Architecture modeling
- Test driven development
- Functional testing
- Test management
- Structured testing
- …

Business Metrics

Project	Time to Market (M)	Quality (Defect Density)	Productivity (SLOCS / PM)
A	22	2.3	200
B	14	1.4	160
C	18	1.6	180
D	9	0.3	150
E	6	0.4	205

Measured Iterative Practice Adoption

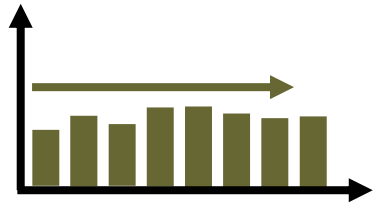


Improve Organizational Capability

1. Ad hoc processes, methods, tools

Ungoverned

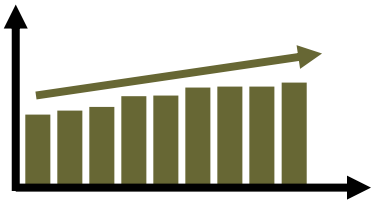
- Custom defined on each project, no ROI



2. Foundation project disciplines

Repeatable practices

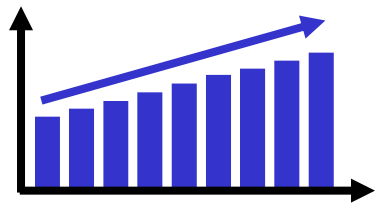
- Analysis & design, configuration and change management, planning, scope management, testing



3. Organizational process discipline

Governance

- Common methods, tools and training, Objective metrics collection



4/5. Software economies of scale

Business driven development

- Business performance optimization, Quantitative process management

Getting to Painless Governance

- What is “painless” governance?
  - An integrated environment (process-methods-tools-assets) that ensures artifact integrity and frees practitioners to design, code and test
- What barriers to adoption exist?
  - Middle level (Project) management is where this war is won
  - Execs and practitioners are easy, PMs are challenging
- What measures/metrics actually work?
  - Metrics extracted from the CM system
  - 1st derivatives on the design base, code base, test base
- How do you balance long-term enterprise needs with short-term project needs?
  - 80% Lifecycle framework (RMC, RUP, principles, common metrics & tools)
  - 20% Project tailoring (context specific details, tactics, standards, measures)
  - Start with minimal, agile process, evolve to rigor

Presenting Governance to Practitioners

- Good Governments frame governance around protecting freedoms of citizens.
- Here are the sorts of freedoms that would raise practitioner eyebrows:
  - Freedom of speech —open honest communication of progress and issues
  - Freedom to change —platform tools and process permit software change freedom
  - Freedom to experiment —process supports trial and error activities and prototyping
  - Freedom to create —automate bookkeeping, instrumentation, change propagation, traceability
    —self documenting designs/code/tests trump all other artifacts
  - Freedom to adapt —right-size the policies and processes to the project specific context
  - Freedom to distribute —platform permits untaxed collaboration
  - Freedom of computing power —platform hardware does not impose artificial constraints
  - Freedom to demonstrate —progress metrics are derived from executable software baselines

Our Platform of process/methods/tools needs to provide for those freedoms
Paint that sort of picture for practitioners and governance is a positive

What is the Value of a Governance Platform?

- Accelerate **time to value**
  - Steer projects iteratively and integrate/test continuously to cut downstream scrap and rework by 50%
- Improve **project predictability**
  - Real time instrumentation of changing work products for lifecycle assessment of progress and quality
- Improve **software economics**
  - Balance existing reusable assets with evolving user needs to optimize economic outcomes
  - Employ know-how and skills from anywhere in a collaborative development environment
- Right-size **development governance**
  - Dynamically adapt process agility & rigor commensurate with the uncertainty in the estimate to complete