Construx®

Delivering Software Project Success

Three Things I Wish I Learned in School

www.construx.com

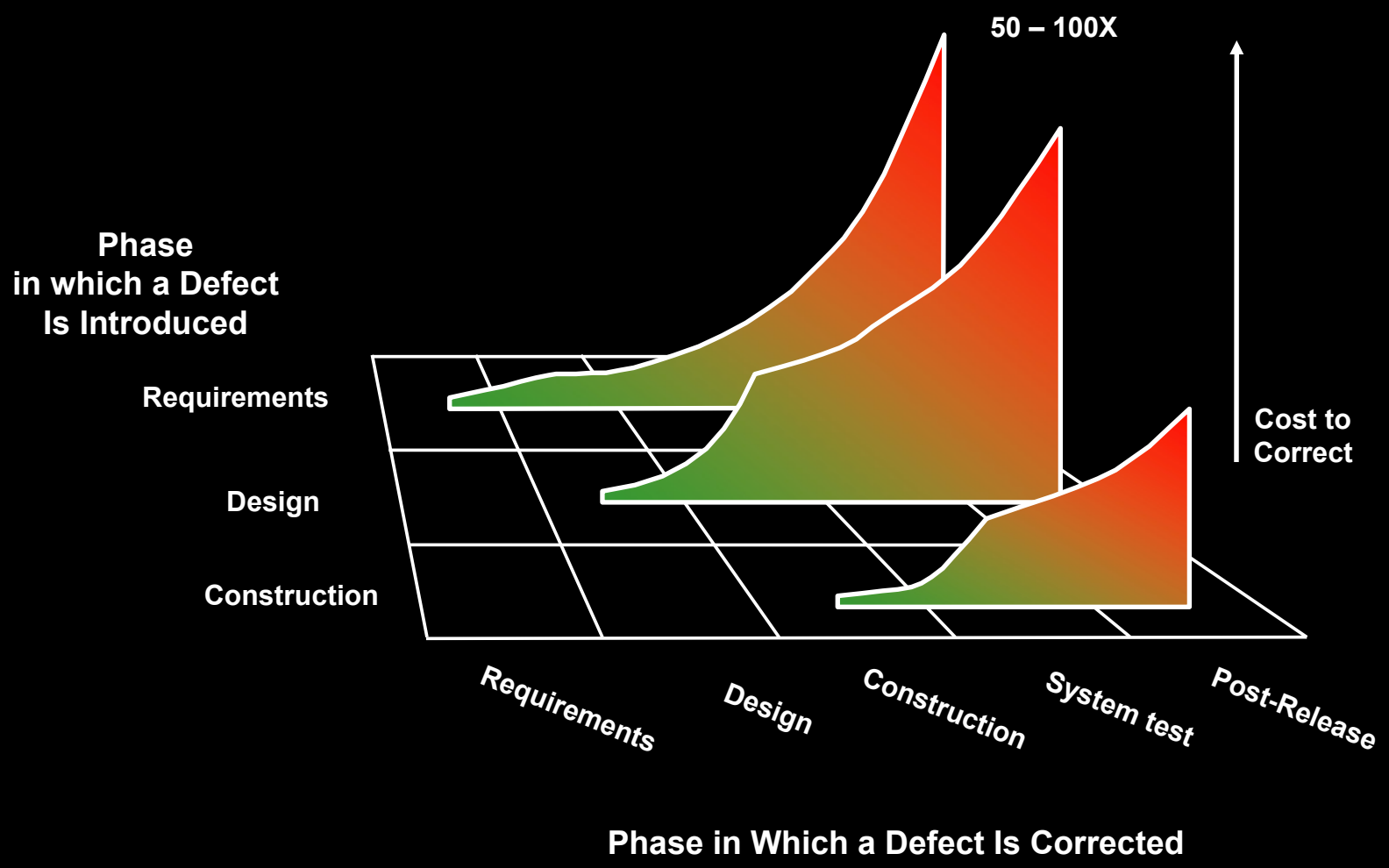
Construx®

Delivering Software Project Success

#1

“Motion” ≠ “Progress”

The Cost of Defects



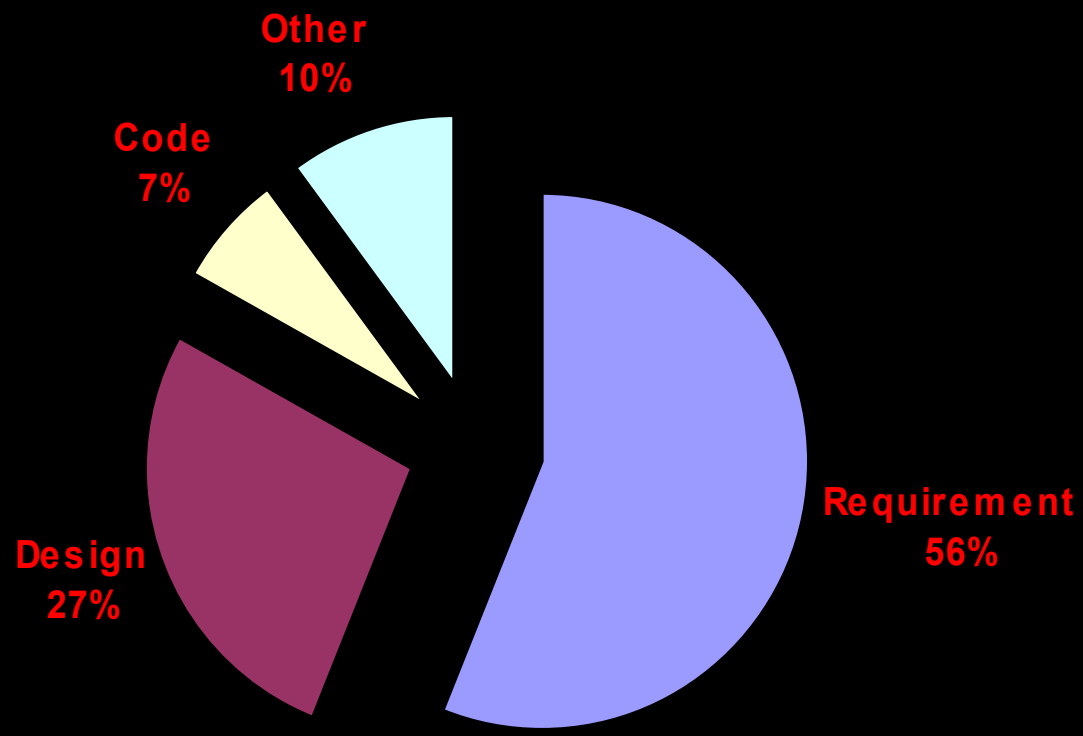
Reference: [McConnell98]

"Software Development Best Practices"

The Cost of Defects (cont)

- Decades of research confirm defect-cost increase
  - Fagan, Michael E. 1976. “Design and Code Inspections to Reduce Errors in Program Development.” IBM Systems Journal 15, no. 3: 182–211
  - Humphrey, Watts S., Terry R. Snyder, and Ronald R. Willis. 1991. “Software Process Improvement at Hughes Aircraft.” IEEE Software 8, no. 4 (July): 11–23
  - Leffingwell, Dean, 1997. “Calculating the Return on Investment from More Effective Requirements Management,” American Programmer, 10(4):13-16
  - Willis, Ron R., et al, 1998. “Hughes Aircraft’s Widespread Deployment of a Continuously Improving Software Process,” Software Engineering Institute/Carnegie Mellon University, CMU/SEI-98-TR-006, May 1998
  - Grady, Robert B. 1999. “An Economic Release Decision Model: Insights into Software Project Management.” In Proceedings of the Applications of Software Measurement Conference, 227-239. Orange Park, FL: Software Quality Engineering
  - Shull, et al, 2002. “What We Have Learned About Fighting Defects,” Proceedings, Metrics 2002. IEEE; pp. 249-258
  - Boehm, Barry and Richard Turner, 2004. Balancing Agility and Discipline: A Guide for the Perplexed, Boston, Mass.: Addison Wesley, 2004

Frequency of Defects



About 83% of all defects exist before a single line of code is written!

Reference: [Mogyorodi03]

Software Project Effort—in Theory

Requirements	w%
Design	x%
Construction	y%
Testing	z%
Total	100%

Rework Percentage (R%)

$$R\% = \frac{\text{Project effort spent on rework}}{\text{Total effort spent on project}}$$

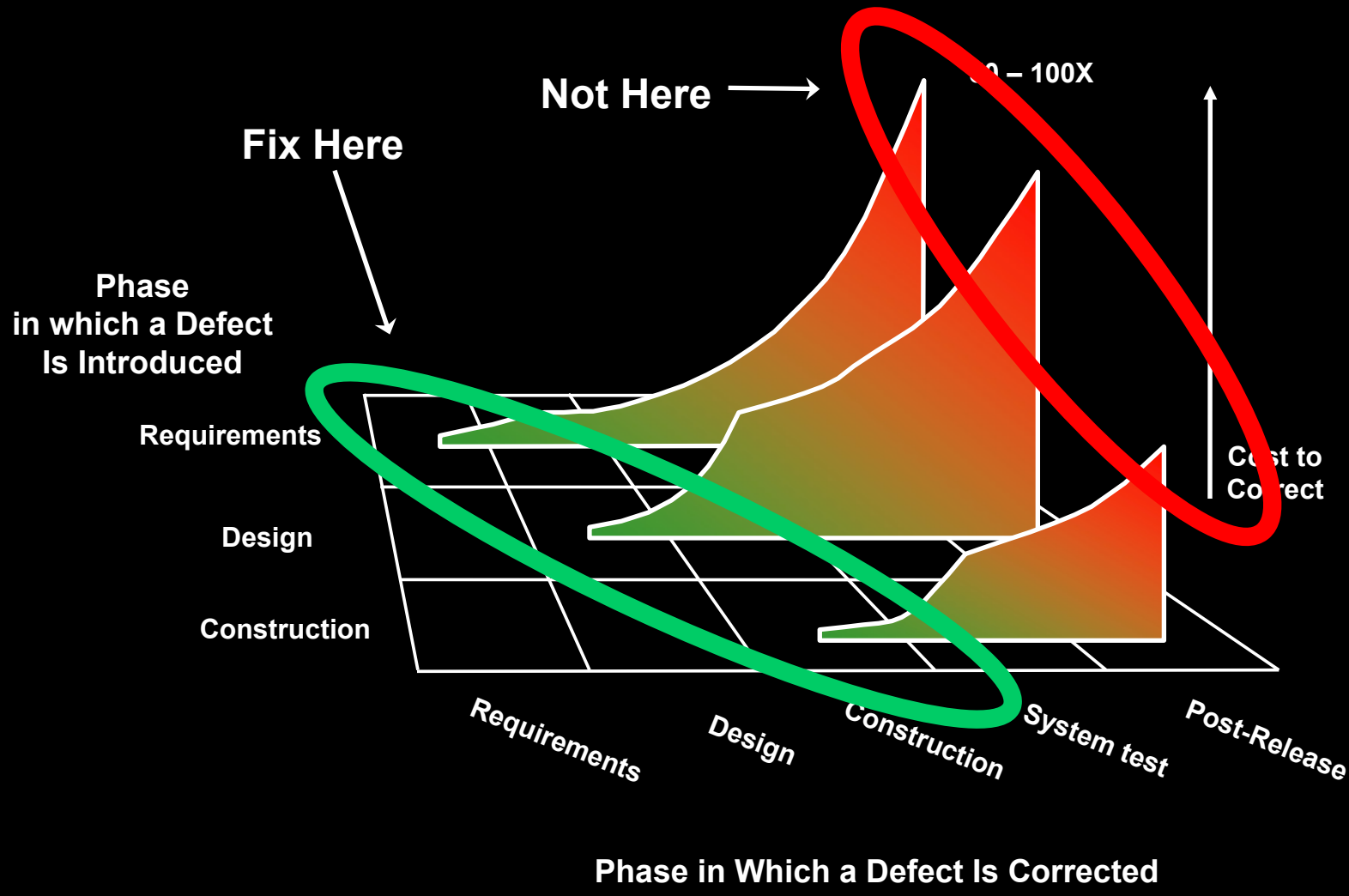
"Estimates of defect rework ranged from 30 to 80 percent of total development effort (when no form of peer review is used)" [Fagan96]

Software Project Effort—in Practice

Requirements	w%
Design	x%
Construction	y%
Testing	z%
Rework (R%)	51%
Total	100%

Rework is not only the single largest driver of cost and schedule on a typical software project—it is larger than all others combined!

Early Defect Detection is Key



Inspections

- Critical examinations of software deliverables
  - Performed by technically competent individuals who did not produce the deliverable being considered
- Identify defects as early in the software lifecycle as possible
  - Defects can be removed while the cost of removal is still low

Testing vs. Inspections

	Testing	Inspections
Effectiveness (%defects found)	60% → 70%	60% → 90%
Efficiency (hours/defect found)	6 → 8	0.8 → 1.2
Cost to repair (hours/defect fixed)	3.4 → 27.2	1.1 → 2.7

References: [Jones99], [O'Neill97], [Russell91], [Wagner06]
"Software Development Best Practices"

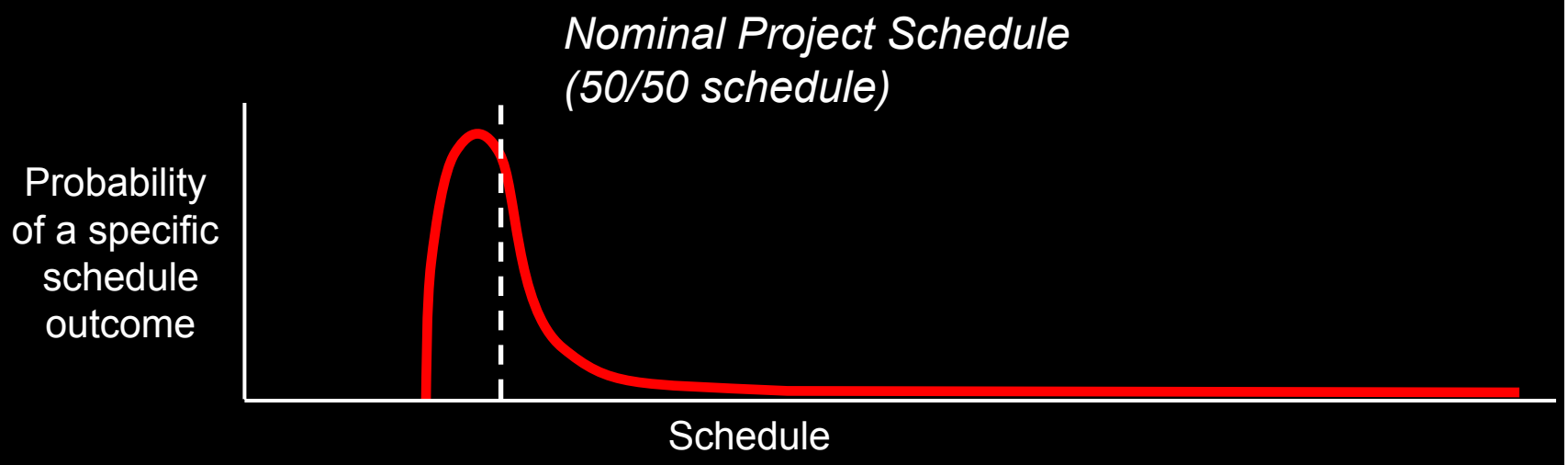
Construx®

Delivering Software Project Success

#2
“Estimate” ≠ “Commitment”

Estimates Have Probabilities

- There is no such thing as a "single-point estimate" that is correct/meaningful
  - All estimates include probabilities—stated or implied (even if the estimator doesn't know it)
  - Estimate probabilities are not symmetrically distributed about a mean



Reference: [Tockey05]

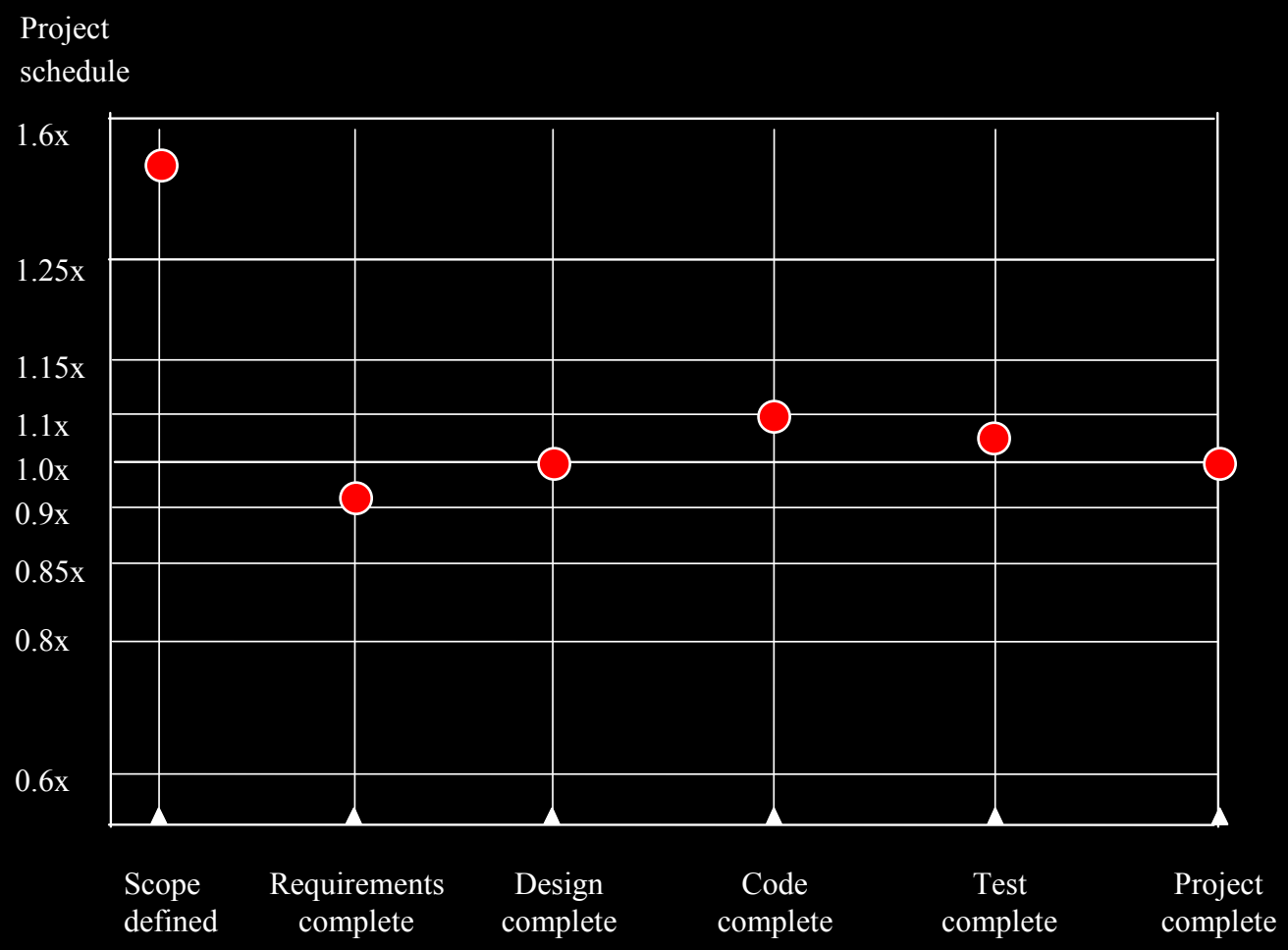
Sources of Estimation Uncertainty

- Will the customer want Feature X?
- Will the customer want the cheap or expensive version of Feature X?
- If you implement the cheap version of Feature X, will the customer later want the expensive version after all?
- How will Feature X be designed?
- How long will it take to debug and correct mistakes made in implementing Feature X?

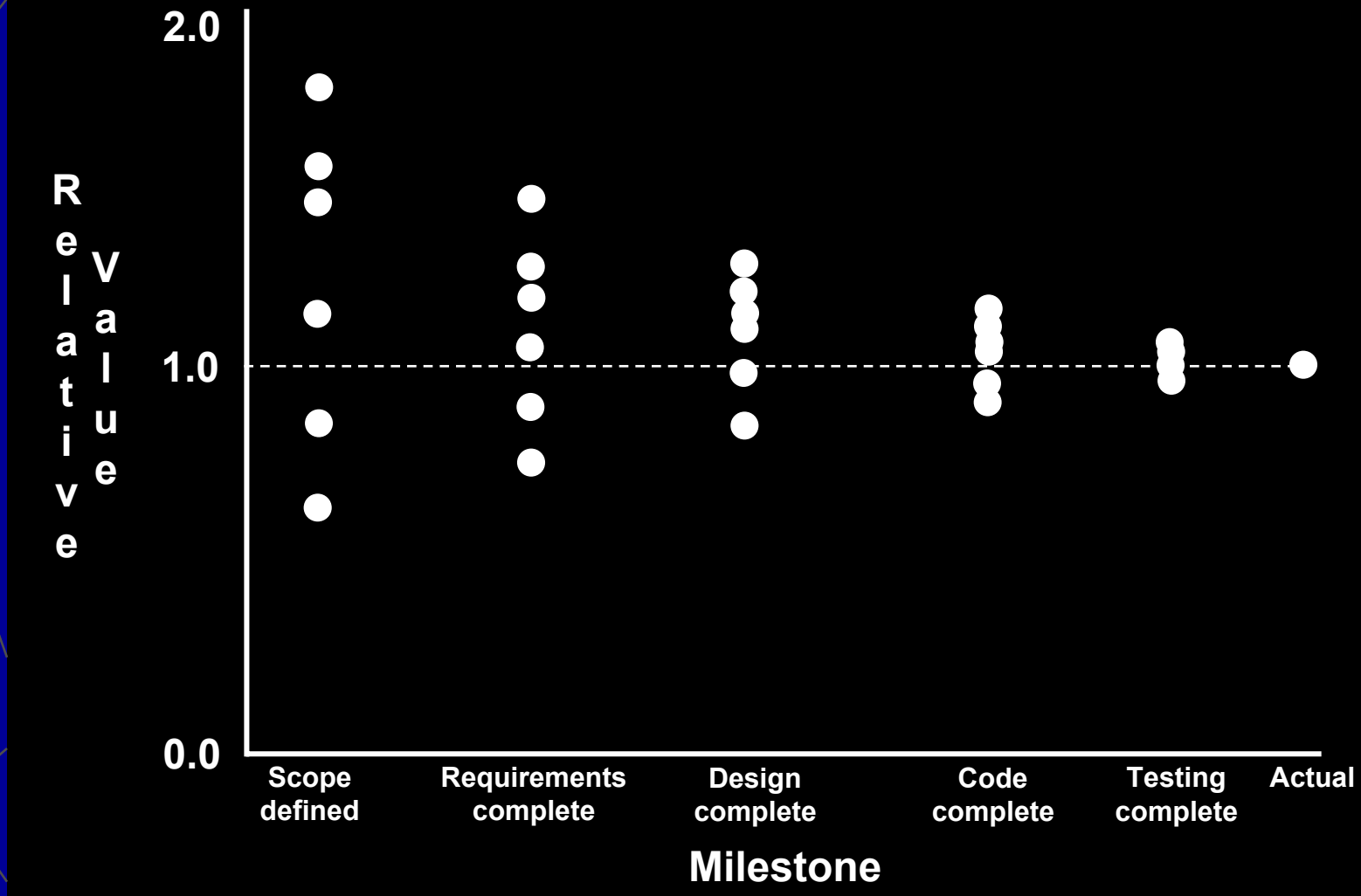
Measuring Estimate Uncertainty

Project: Framus	Estimate	Act/Est
Scope defined	*7 mos*	*1.43*
Requirements complete	*11 mos*	*0.91*
Design complete	*10 mos*	*1.00*
Code complete	*9 mos*	*1.11*
Test complete	*9.5 mos*	*1.05*
Project complete	*10 mos*	*1.00*

Relative Error in Estimates



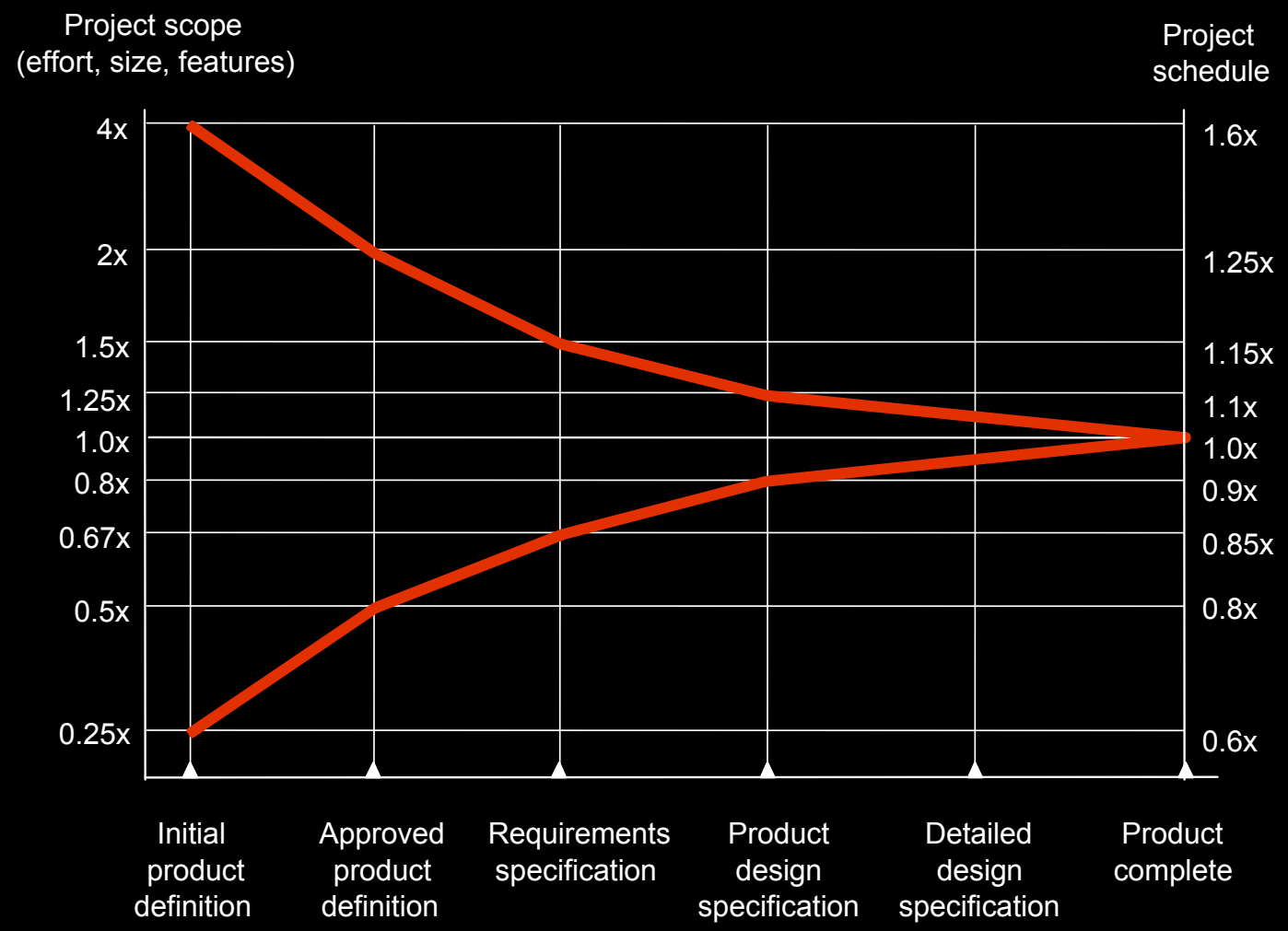
Relative Error in Estimates (cont)



Construx

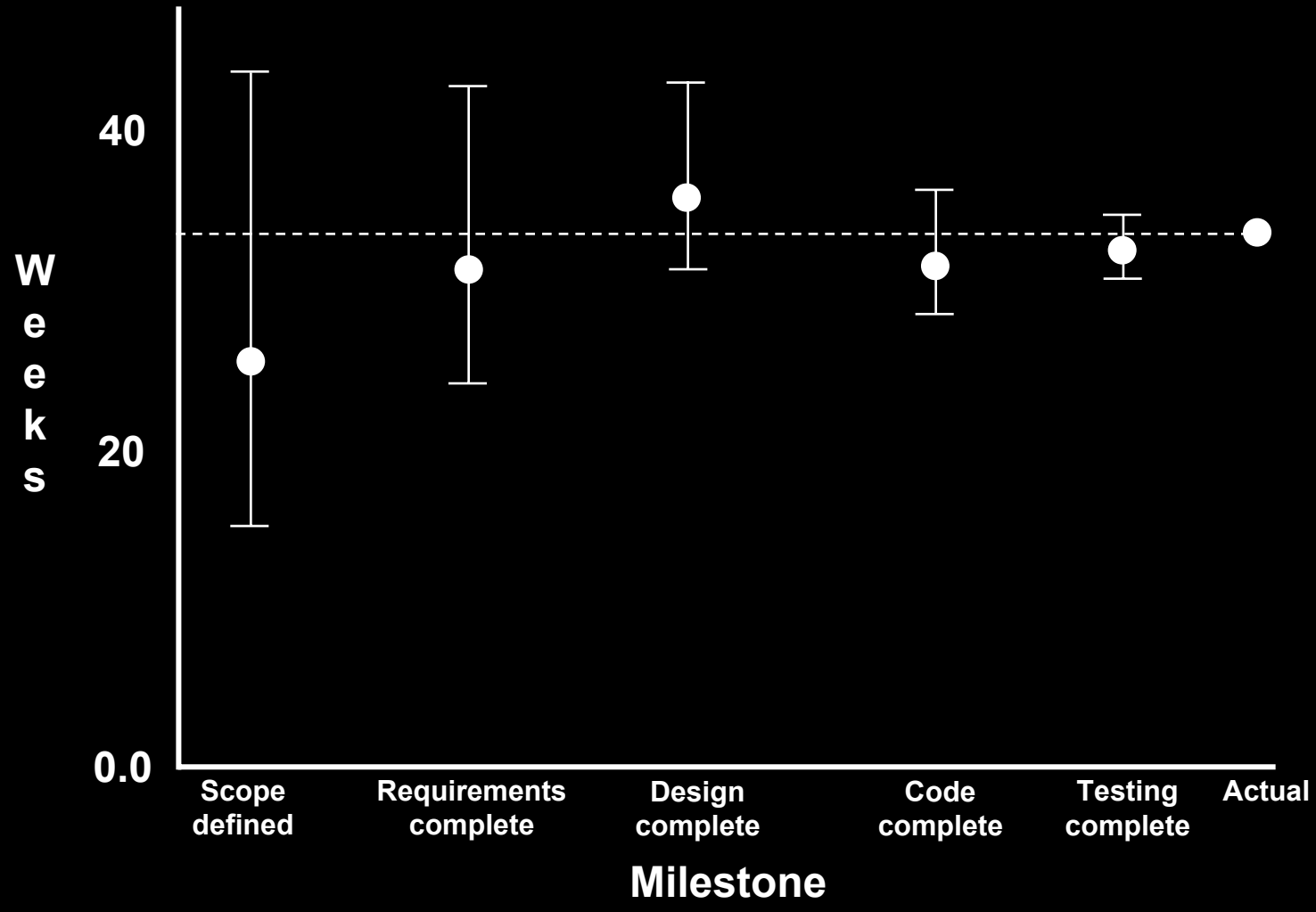
"Software Development Best Practices"

The “Cone of Uncertainty”



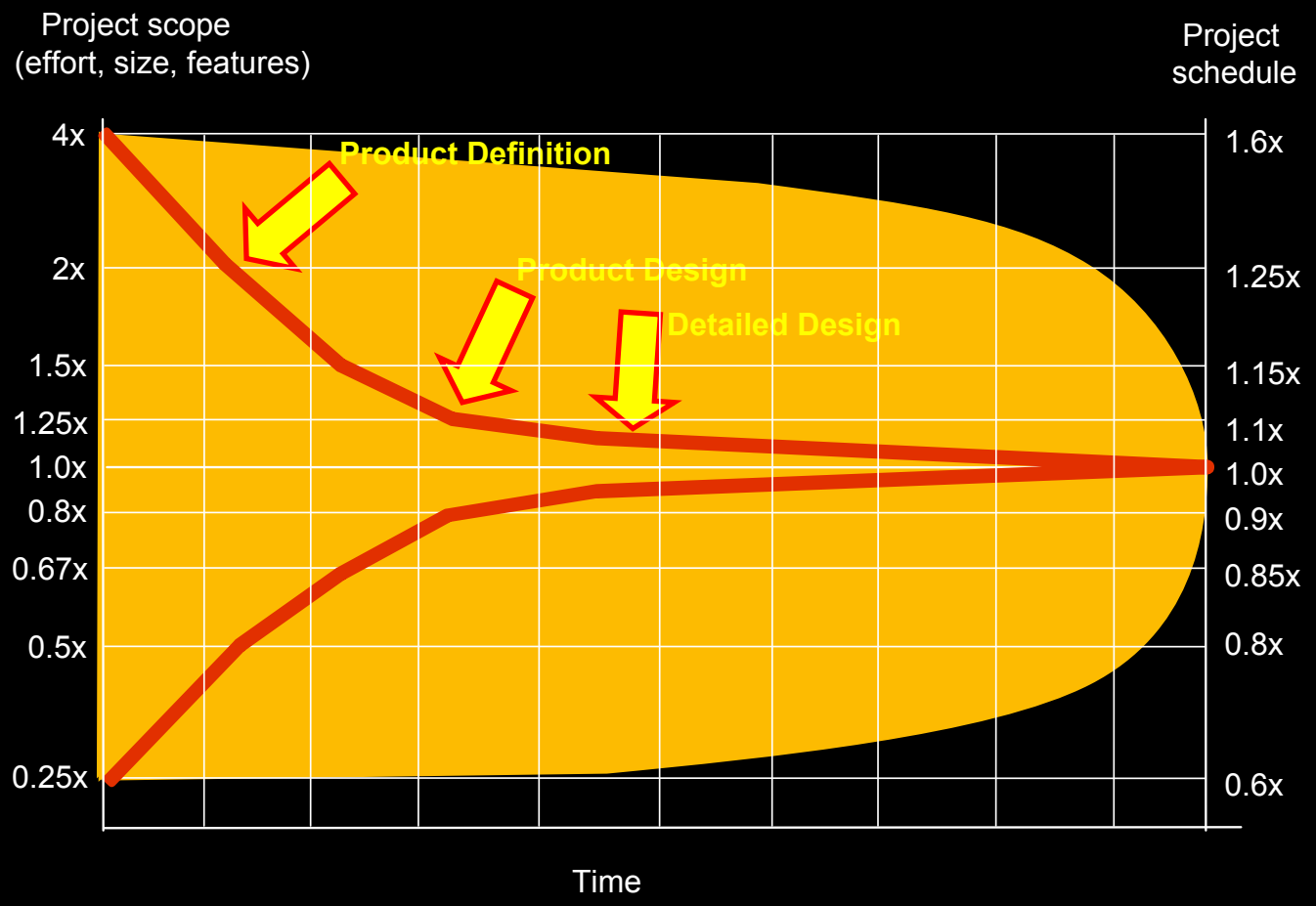
Source: *Software Cost Estimation with Cocomo II* (Boehm 2000)

Using Your Cone of Uncertainty



Construx

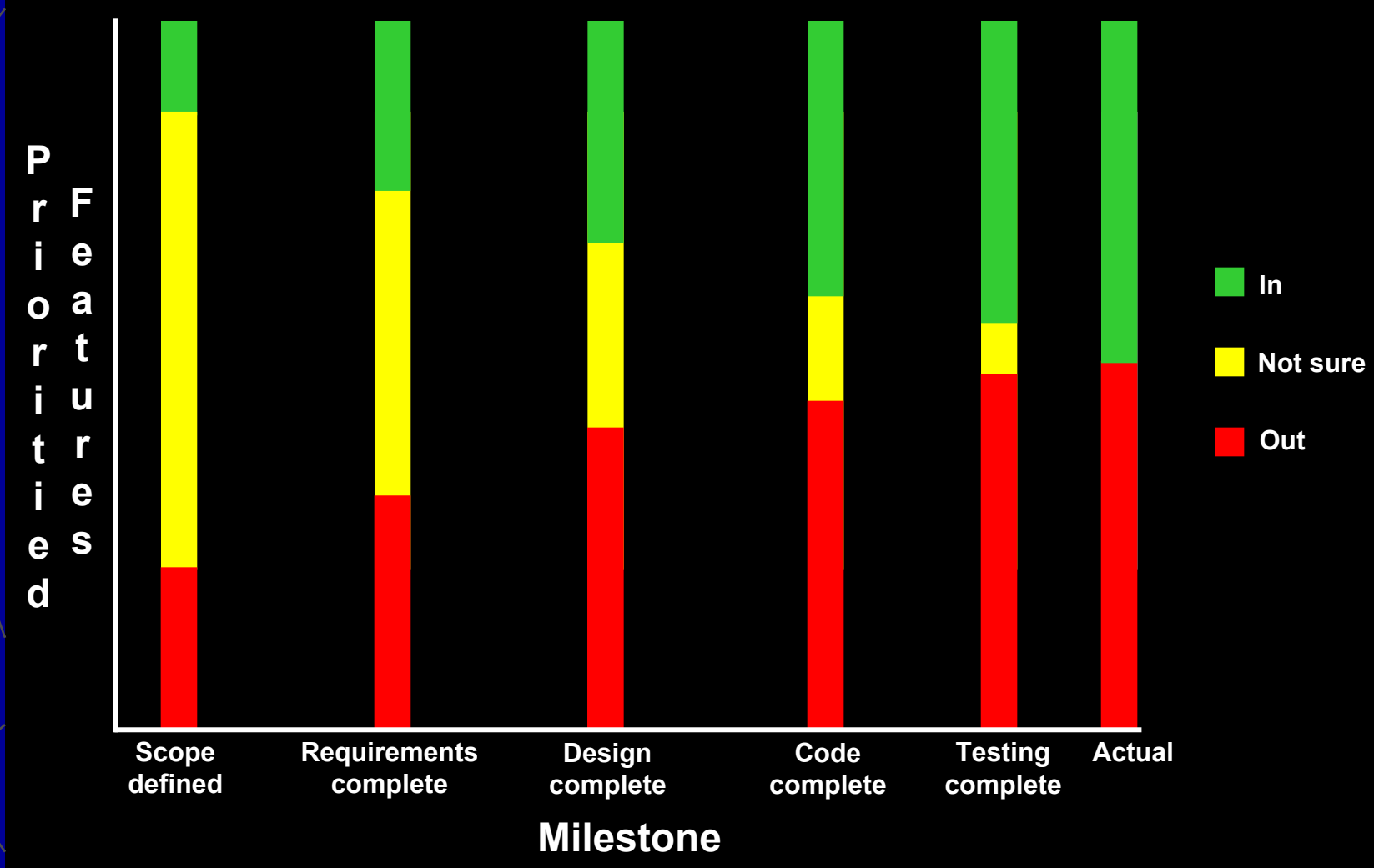
Cloud of Uncertainty?



Cone of Uncertainty Given Fixed Schedule

Priority	Feature	Best-case estimate	Worst-case estimate
1	F7	9	16
2	F3	5	9
3	F9	6	10
4	F2	1	3
5	F1	3	7
6	F5	4	8
7	F6	6	11
8	F10	2	6
9	F4	5	8
10	F8	6	10

Cone of Uncertainty Given Fixed Schedule (cont)



Construx

"Software Development Best Practices"

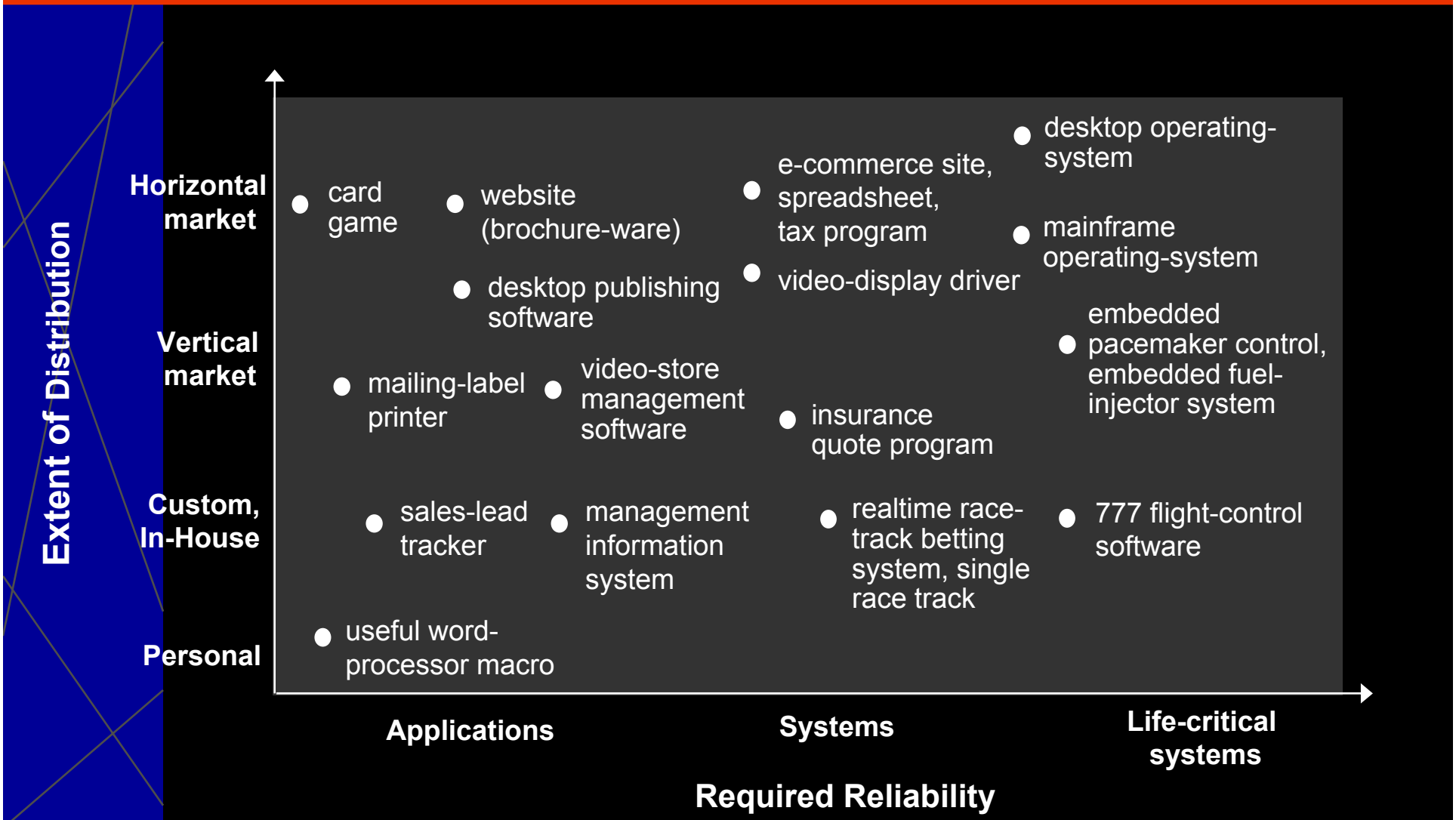
Construx®

Delivering Software Project Success

#3

One software process does not fit all projects

One Size Does Not Fit All

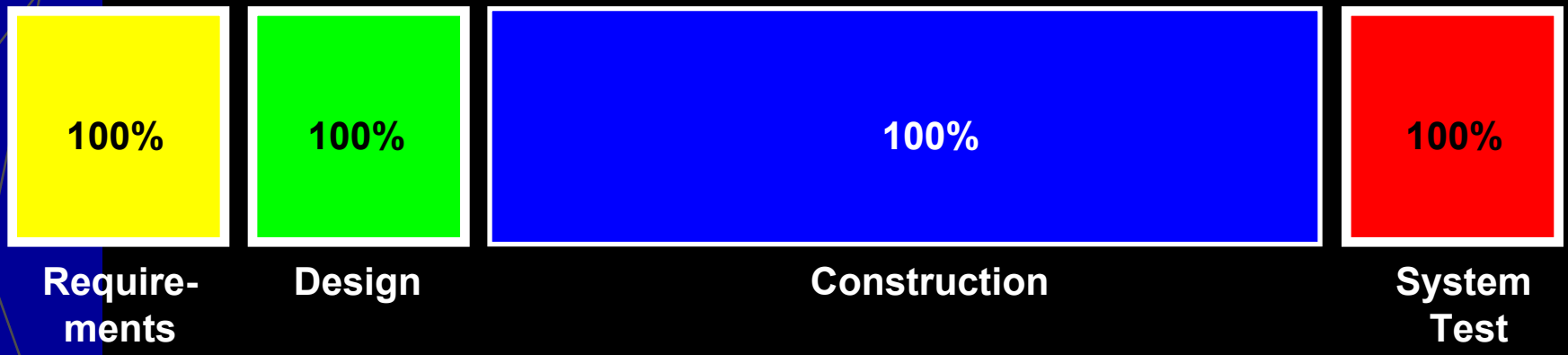


“Activity” ≠ “Phase”

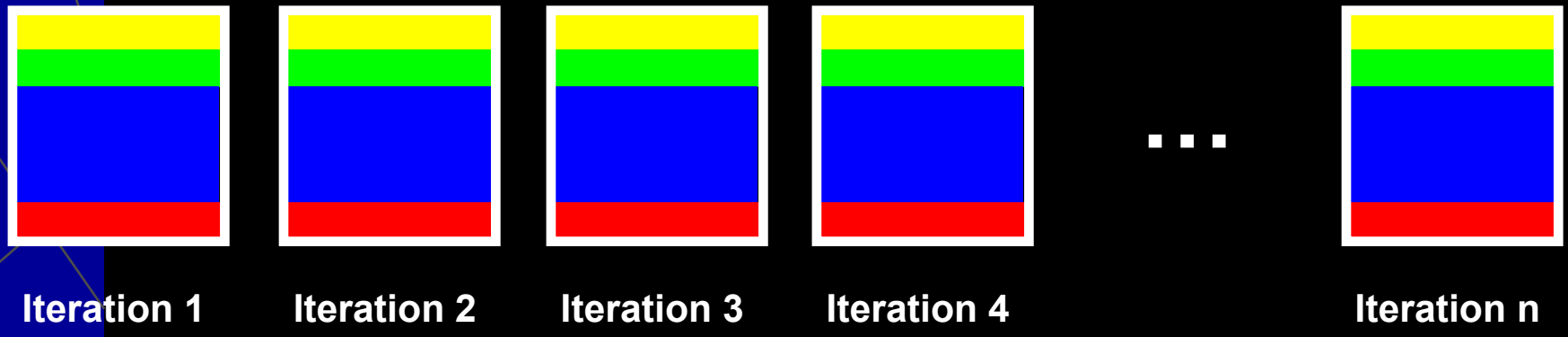
- Software development consists of well-known activities:
  - Requirements
  - Design
  - Construction
  - Test
- These activities may or may not be performed in phases

The Two Extremes: 100% Sequential vs. 100% Iterative

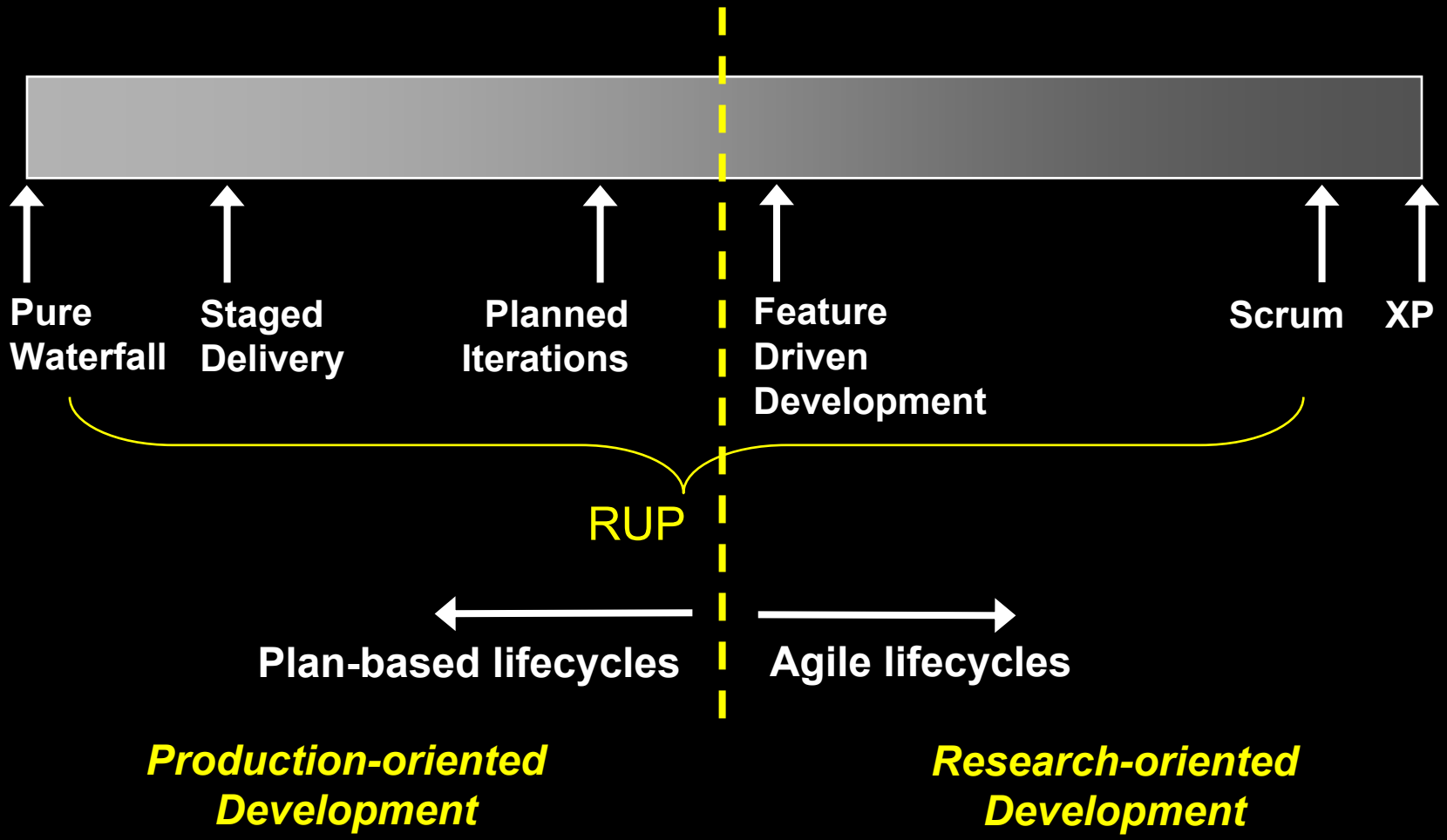
100% Sequential (e.g., Waterfall Development)



100% Iterative (e.g., Extreme Programming)



A Spectrum of Software Processes



Selecting a Lifecycle

- Does the lifecycle meet the project's need to:
  - Control cost and schedule?
  - Provide progress visibility?
  - Handle requirements or design uncertainty?
  - Produce reliable, maintainable products?
  - Deliver business value early (i.e., in stages)?
  - Be simple to understand and easy to use?
  - …

References

- [Fagan96] Michael Fagan, “Foreword” in Wheeler, Brykczynski, Meeson, *Software Inspection - An Industry Best Practice*, IEEE Computer Society Press, 1996
- [Jones99] Capers Jones, *“Software Quality in 1999: What Works and What Doesn’t”*, Software Quality Research Inc., 1999
- [McConnell98] Steve McConnell, *Software Project Survival Guide*, Microsoft Press, 1998
- [Mogyorodi03] Gary Mogyoridi, *“What is Requirements-Based Testing”*, Crosstalk, March 2003
- [O’Neill97] Don O’Neill, *“Setting Up a Software Inspection Program”*, CrossTalk, February, 1997
- [Russell91] Glen Russell, *“Experience with Inspection in Ultralarge-Scale Developments”*, IEEE Software, Vol 8, No 1, January, 1991
- [Tockey05] Steve Tockey, *Return on Software: Maximizing the Return on your Software Investment*, Addison Wesley, 2005
- [Wagner06] Stefan Wagner, *“A literature survey of the quality economics of defect-detection techniques”*, Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering ISESE '06, 2006

Contact Information

Construx

Delivering Software Project Success

- **Training**
- **Consulting/Coaching**
- **Software Tools**

sales@construx.com

construx.com

(425) 636-0100