No.1 Software Quality

Hironori Washizaki

Twitter: @Hiro_Washi washizaki@waseda.jp

http://www.washi.cs.waseda.ac.jp/

Course guidance

Software Measurement and Estimation A Practical Approach

演習で学ぶ
ソフトウエア
メトリクスの基礎

【ソフトウエアの測定と見積もりの正しい作法】

Linda M.Laird and M.Carol Brennan
リンダ・M・ライルド＋M・キャロル・ブレナン 著
Makoto Nonaka and Hironori Washizaki
野中 誠＋鷲崎弘宜 訳

169題の演習を通じて
ソフトウエアの
欠陥・工数・規模が
"見える"！

「数字による管理」を確立し、
進化させるための実践的ガイド

日経BP社 定価(本体2800円＋税)

- 資料は CourseN@vi
- Course materials on CourseN@vi
- 2/3 以上出席してください。
- 2/3 attendance required
- Text book
  - [Linda09] リンダ・M・ライルド, M・キャロル・ブレナン著, 野中誠, 鷲崎弘宜訳 , "演習で学ぶソフトウエアメトリクスの基礎 ソフトウエアの測定と見積もりの正しい作法", 日経BP社 , 2009.
  - Linda M. Laird and M. Carol Brennan, ``Software Measurement and Estimation: A Practical Approach," John Wiley & Sons, 2006.
- Other books
  - [Pressman05] Roger S. Pressman 著, 西康晴, 榊原彰, 内藤裕史監訳, "実践ソフトウェアエンジニアリング", 日科技連出版社, 2005 ("Software Engineering: A Practitioner's Approach")
  - [SWEBOK] 松本 吉弘 (翻訳) , "ソフトウェアエンジニアリング基礎知識体系—SWEBOK2004", オーム社 , 2005.
  - [SQuBOK] SQuBOK策定部会 編, "ソフトウェア品質知識体系ガイド - SQuBOK Guide", オーム社, 2007

Work: Software Quality

- What is software quality?
- Any story on good or poor software quality?

Difficulty in defining software quality

- Depending on contexts
- Depending on requirements
- Different aspects
- Different targets
- Software properties: flexible, complex, change-prone, invisible

品質とは What is Quality

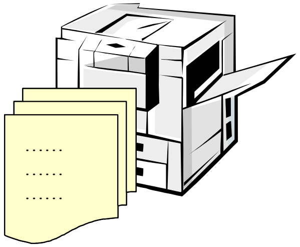
- （一般の）品質: あるものの特性または属性 [American Heritage Dictionary]
- Quality in general: characteristic or attribute of entity

- （ソフトウェアの）品質: ソフトウェアの使用時に必要性を満たす能力を決定する属性全体 [ISO9126-1]
- Software quality: capability of software product to satisfy stated and implied needs when used under specified conditions

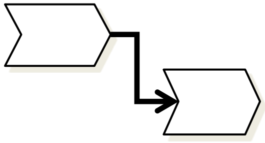
Software quality: target

- Service and work

- Product



- Process



- Resource

Software quality: characteristic

- Category of software quality attributes that bears on software quality
  - Inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

- Functionality
  - The capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.
- Reliability
  - The capability of the software to maintain the level of performance of the system when used under specified conditions
- Efficiency
  - The capability of the software to provide the required performance relative to the amount of resources used, under stated conditions.
- Usability
  - The capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.
- Maintainability
  - The capability of the software to be modified.
- Portability
  - The capability of software to be transferred from one environment to another.

Quality model: interpretation and classification of characteristics

- Any software products
  - ISO/IEC 9126-1:2001
  - ISO/IEC 25010:2011
  - FURPS+
- Requirements specifications
  - IEEE Std 830-1998

ISO/IEC 9126-1:2001

利用時の品質 Quality in use			
有効性 Effectiveness	生産性 Productivity	安全性 Safety	満足性 Satisfaction

内部品質・外部品質 Internal and external quality					
機能性 Functionality	信頼性 Reliability	使用性 Usability	効率性 Efficiency	保守性 Maintainability	移植性 Portability
Suitability	Maturity	Understandability	Time behavior	Analyzability	Adaptability
Accurateness	Fault tolerance	Learnability	Resource behavior	Changeability	Installability
Interoperability	Recoverability	Operability	Compliance	Stability	Co−existence
Security	Compliance	Attractiveness		Testability	Replaceability
Compliance		Compliance		Compliance	Compliance

ISO/IEC 9126-1:2001，Software engineering−Product quality−Part 1: Quality model

ISO/IEC 25010:2011 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models

利用時の品質 Quality in use				
有効性 Effectiveness	**効率性** **Efficiency**	**リスク回避性** **Freedom from risk**	満足性 Satisfaction	**利用状況網羅性** **Context coverage**

内部品質・外部品質 Internal and external quality							
機能適合性 **Functional suitability**	**互換性** **Compatibility**	**セキュリティ** **Security**	信頼性 Reliability	使用性 Usability	性能効率性 Performance efficiency	保守性 Maintainability	移植性 Portability
F. completeness	Co-existence	**Confidentiality**	Maturity	**Appropriateness recognisability**	Time-behavior	Analysability	Adaptability
F. correctness	Interoperability	**Integrity**	Fault tolerance	Learnability	Resource utilization	**Modifiability**	Installability
F. appropriateness		**Non-repudiation**	Recoverability	Operability	**Capacity**	**Reusability**	Replaceability
		Accountability	**Availability**	**User error protection**		Testability	
		Authenticity		**UI aesthetics**		**Modularity**	
				Accessibility			

FUPRS+

Functionality	Usability	Reliability	Performance	Supportability	+: other constraints
機能群	人的要因	障害の頻度と深刻度	速度	テスト容易性	Language
能力	美しさ	回復可能性	効率	拡張性	Cost, deliverly
セキュリティ	・・・	・・・	・・・	順応性	
				保守容易性	
				・・・	

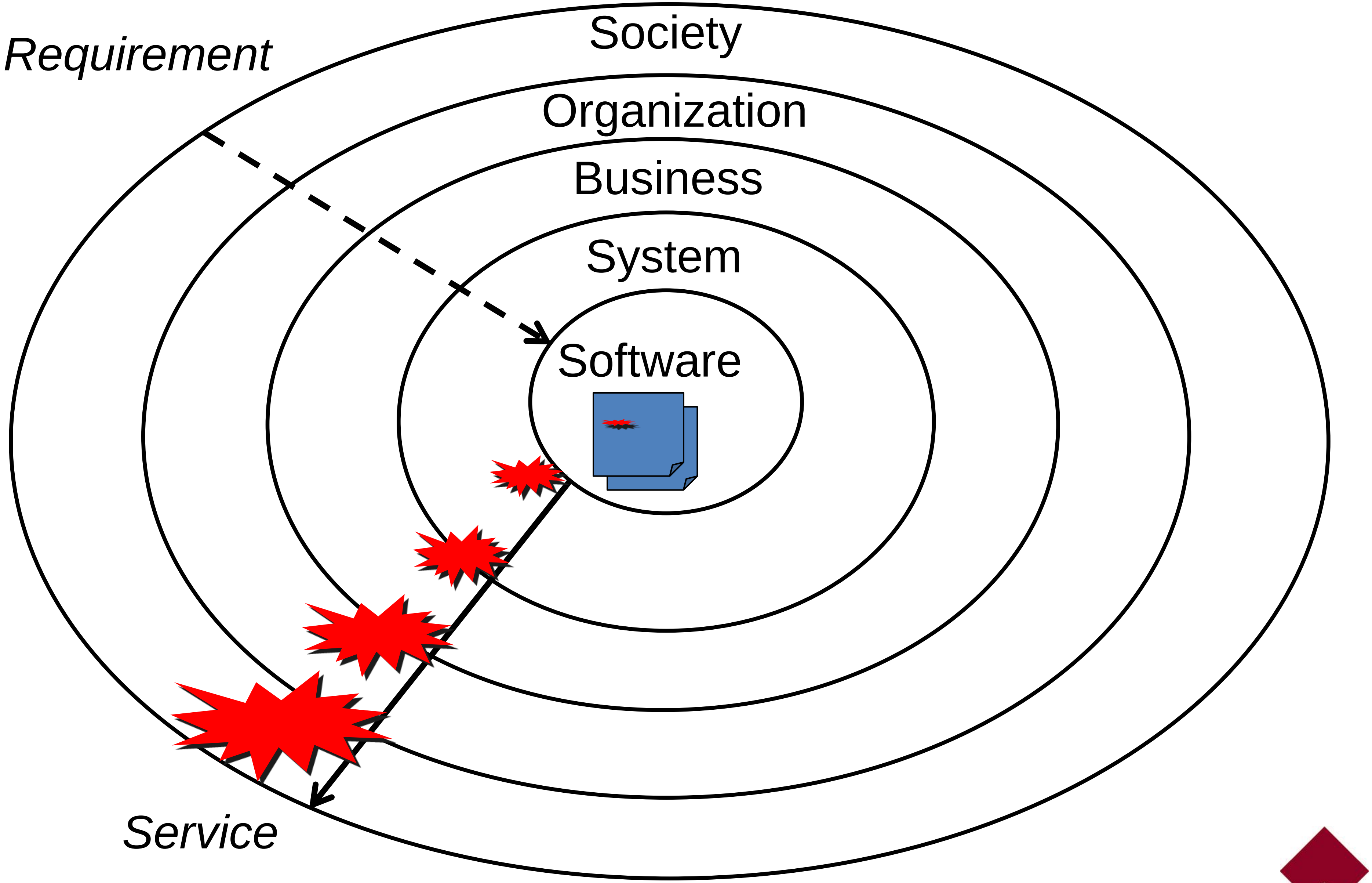
Robert Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice-Hall, 1992

IEEE Std830-1998

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

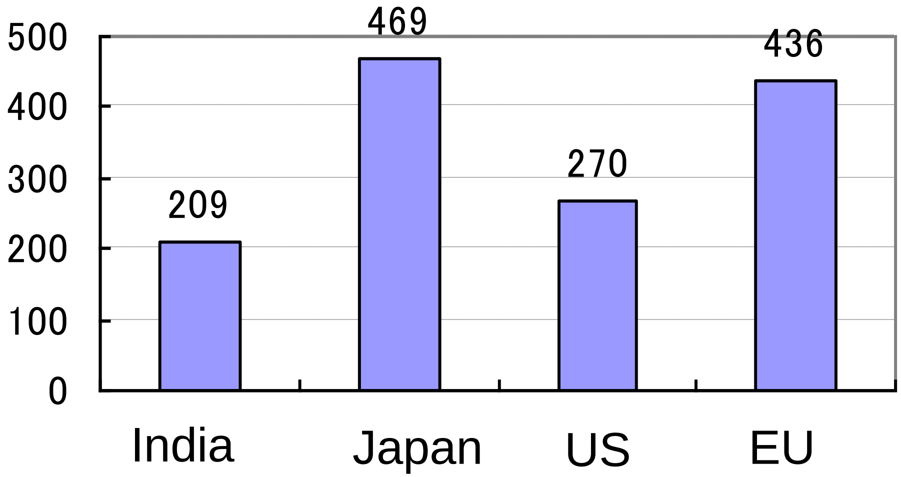
View from Software



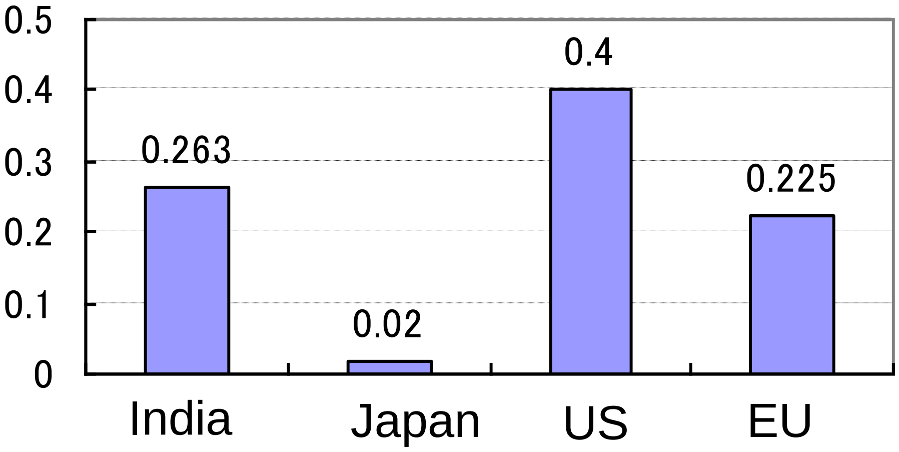
Kikuma Ring [Sakakibra’09]

Software Defects

- Windows 2000 contained 63,000 “known” defects when released
- Japanese software developments: high productivity and quality?
- Success rates in QCD: 91% vs. 27%



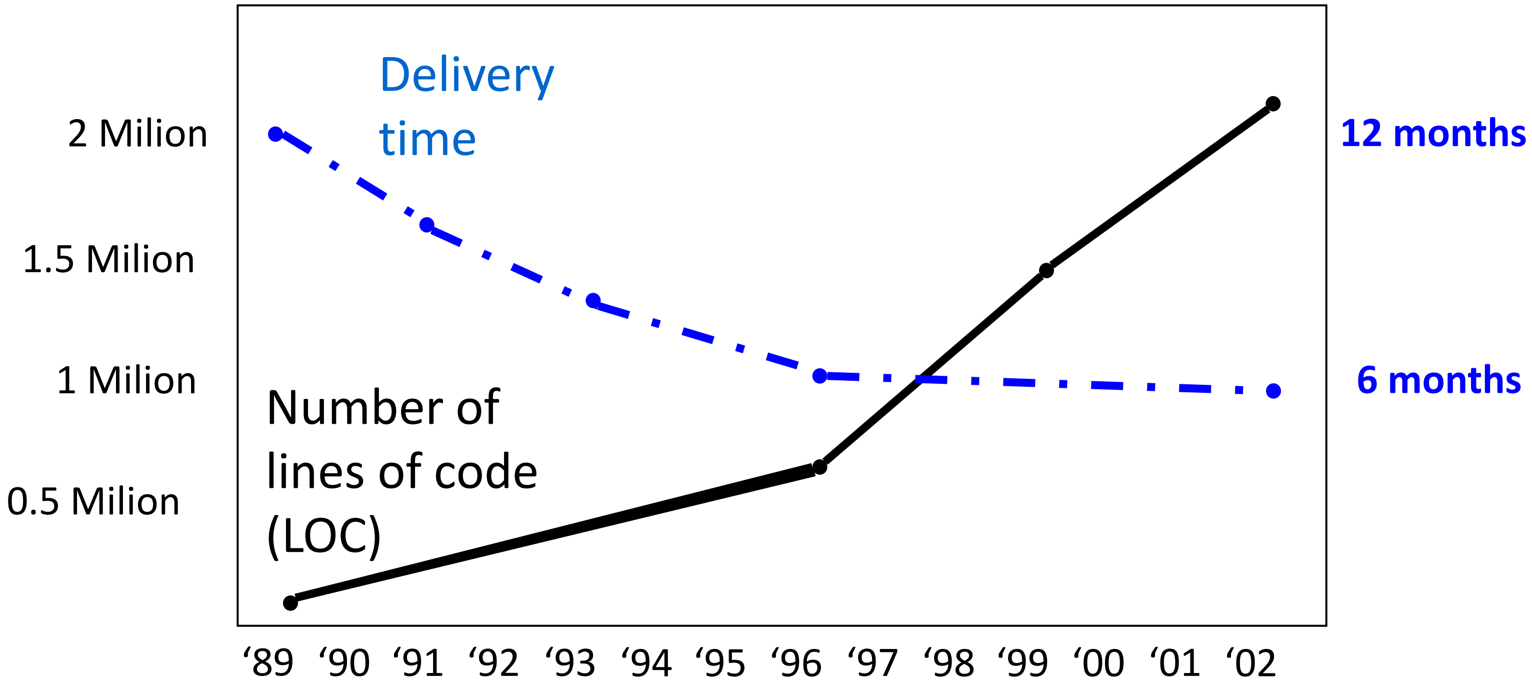
Productivity: LOC / Man-month



Defect density: Defects within one year after release / KSLOC

M. Cusumano, “Software Development Worldwide: The State of the Practice”, IEEE Software, 2003.

Software and its Size

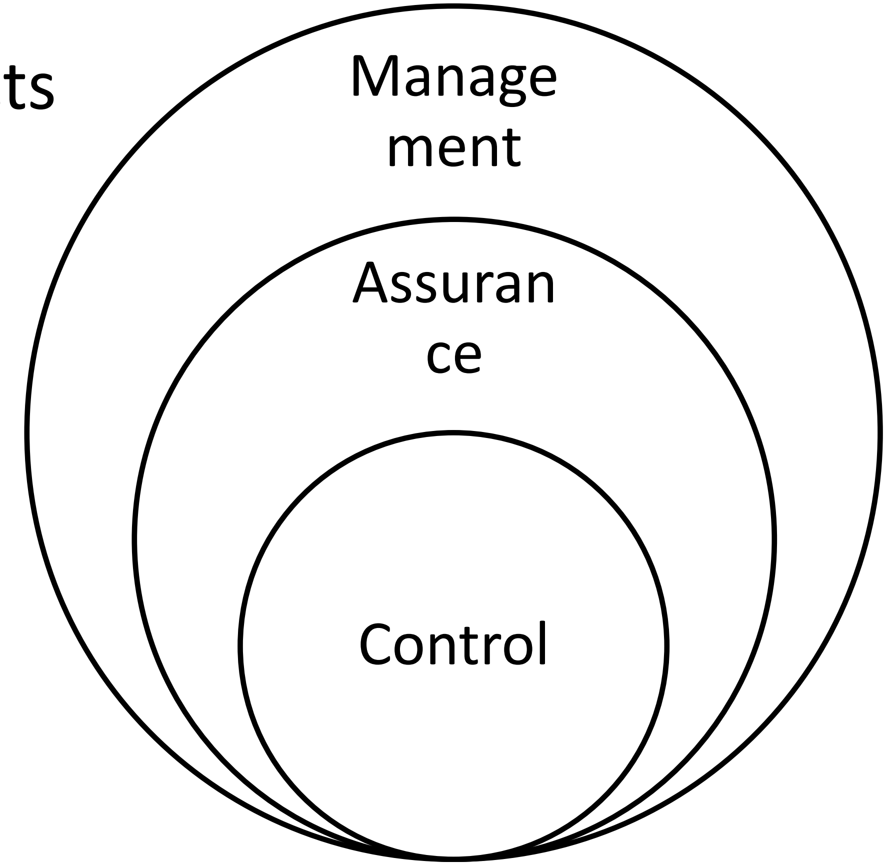


Feature-phone (NOT smart-phone): size becomes three times by two years

M. Hirayama, Quality Improvement in Embedded System Development, ET2002 TB-6, 2002

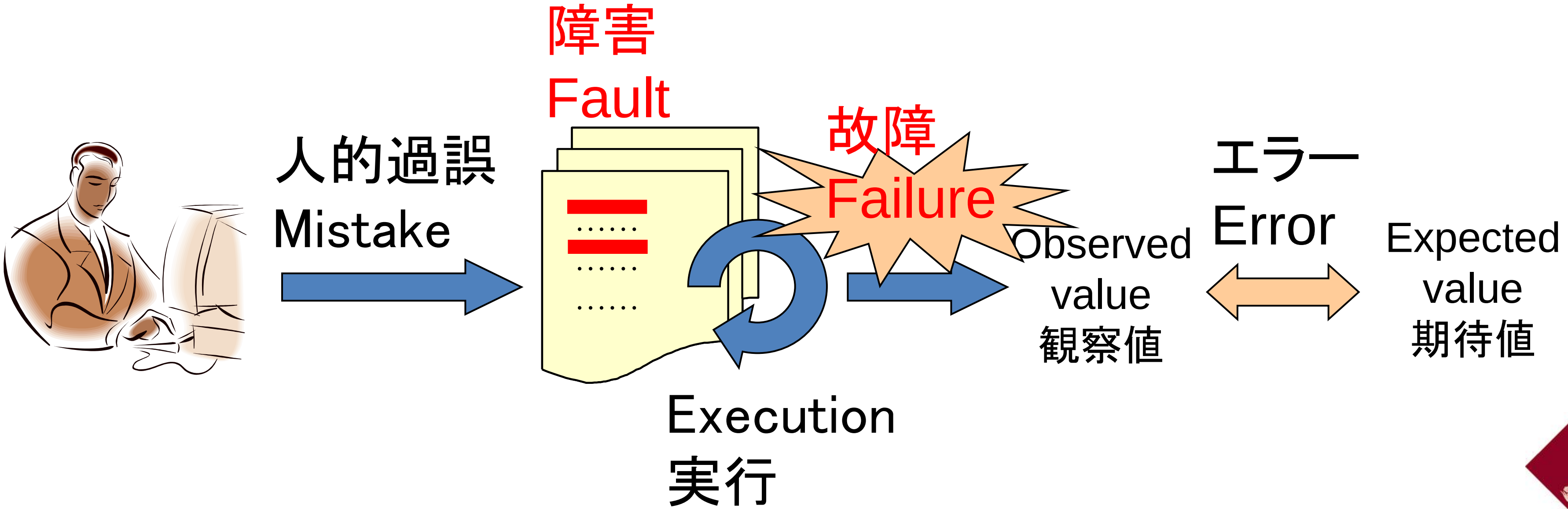
What we need

- Quality control
  - Activities making products satisfying requirements
- Quality assurance
  - Control + auditing and reporting
- Quality management
  - Organizational activities



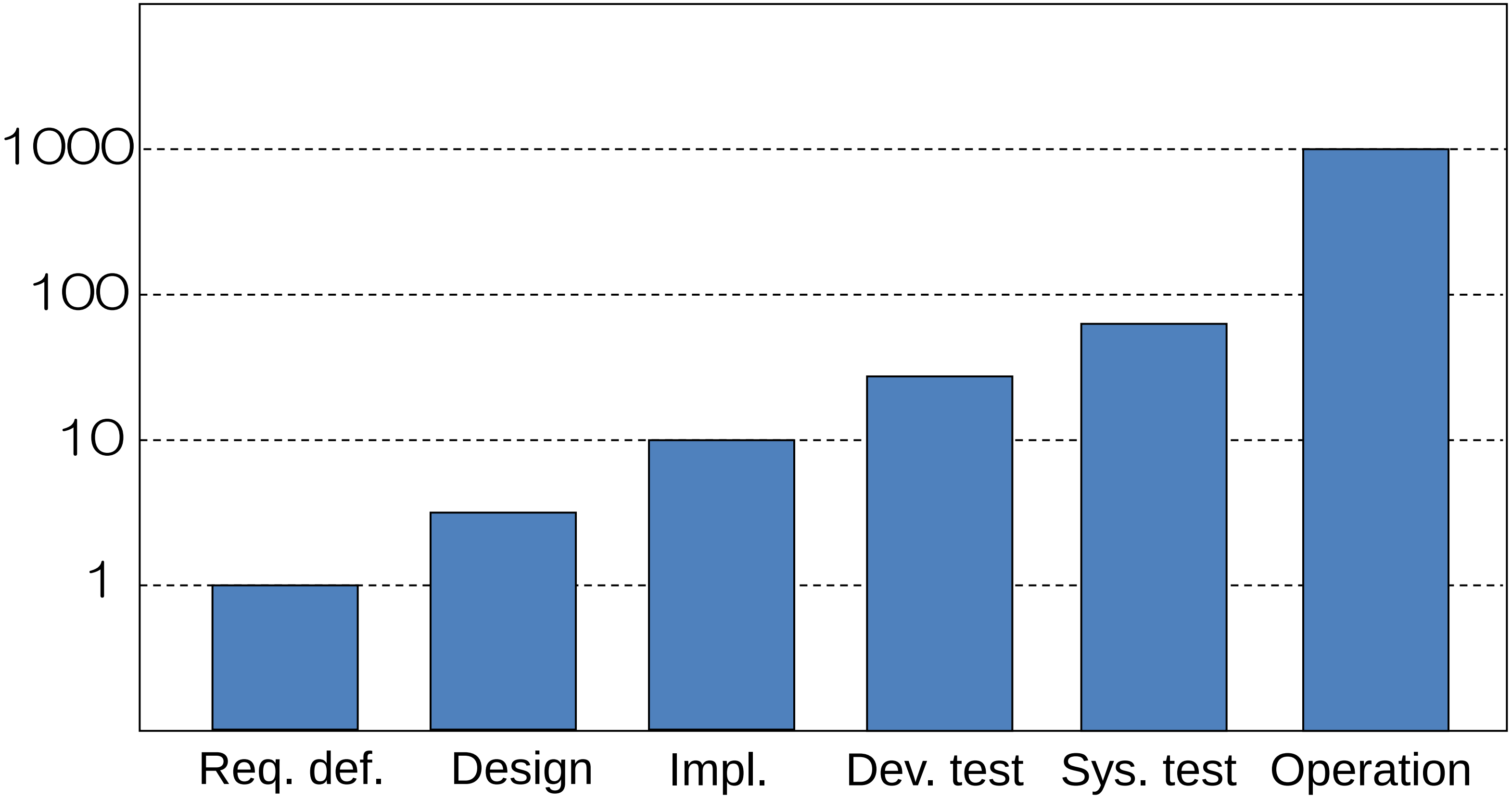
What to do

- Quality requirements definition
- Review
- Testing
- Formal verification
- Measurement
- Root cause analysis and improvement

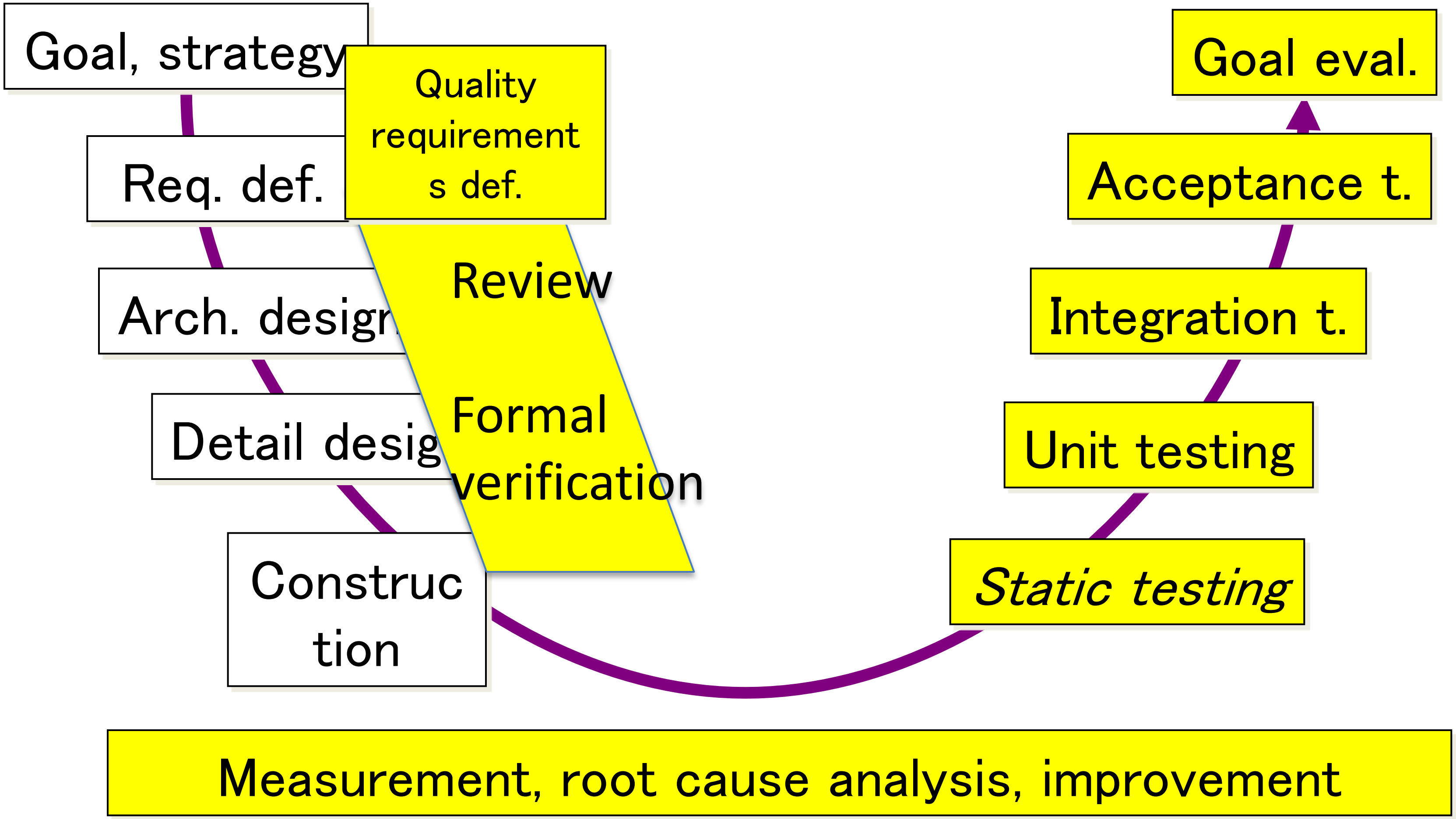


When to do

- Cost for removing defects exponentially increases in entire process [Pressman05]



Lifecycle process and quality assurance



V&V

- 検証（Verification）
  - 正しくソフトウェアを作れているのか
  - Are we building the product right?
- 妥当性確認（Validation）
  - 正しいソフトウェアを作れているのか
  - Are we building the right product?

Summary

- Diversity and difficulty in defining software quality
- Major quality models
- Current status of software quality
- Software quality assurance technologies
- Lifecycle process and V&V