

Software Metrics

- **Software engineering ?**
  - Science or engineering without measurements?
  - Control software process without metrics?
- **What/how do we measure?**

---

---

---

---

---

---

---

---

Role of Measurement

“The purpose of measurement in an industrial organization is to provide information that improves decision making in time to affect the outcome of the process (i.e., business success).”

David N. Card  
Software Productivity Solutions, Inc.  
dnc@sps.com

---

---

---

---

---

---

---

---

DeMarco’s Law

- **Costs (and errors) migrate**
  - From something that is measured
  - To something that is not
- **Thus, all costs and errors must be captured**
  - At least at a high level

---

---

---

---

---

---

---

---

Prerequisites

- **Process must exist first**
  - Documented
  - Practiced
- **Can't measure activities that are not defined**

---

---

---

---

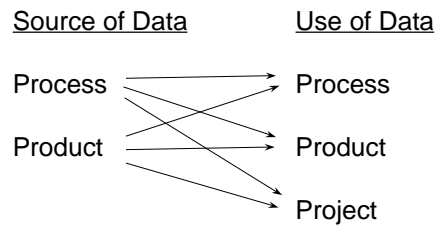
---

---

---

---

Classes of Measures




---

---

---

---

---

---

---

---

Direct Measures

- **Cost & effort**
- **Product size**
- **Product performance**
- **Defect rate**

---

---

---

---

---

---

---

---

Indirect Measures

- **Functionality**
- **Quality**
- **Complexity**
- **Reliability**
- **Maintainability**

---

---

---

---

---

---

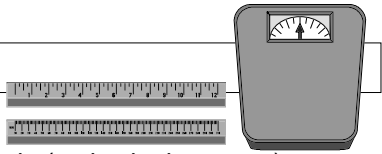
---

---

---

---

Metrics



- **Simple**
  - Lines of code (varies by language)
- **More complex**
  - Function points
    - Language independent?
- **Simple may be good enough**
  - In a specific, stable environment

---

---

---

---

---

---

---

---

---

---

Product "Size"

- **Literal size**
  - 1000's of lines of code (KSLOC)
    - Language dependency
    - Is more always better?
- **Function**
  - More abstract "capabilities"

---

---

---

---

---

---

---

---

---

---

Function Points

Parameter	Weight			Total
User inputs	3	4	6	
User outputs	4	5	7	
User inquiries	3	4	6	
Files	7	10	15	
Interfaces	5	7	10	

Complexity adjustment factor

IFPUG - International Function Point Users' Group

---

---

---

---

---

---

---

---

---

---

---

---

Feature Points T. Capers Jones, SPR

- **Similar to function points**
  - FP optimized for "business" applications
- **Single weight value**
- **Adds "algorithms" category**
  - For "engineering" applications

---

---

---

---

---

---

---

---

---

---

---

---

3-D Function Points Scott A Whitmire

- **Data dimension [business]**
  - Uses FP for this measurement
- **Function dimension [engineering]**
  - Steps + semantics = transformations
- **Control dimension [real-time]**
  - States and transitions

---

---

---

---

---

---

---

---

---

---

---

---

Baseline Data

- **Collect historical information**
- **Two purposes:**
  - Basis for future estimates
  - Reference level to judge process improvement

---

---

---

---

---

---

---

---

Estimation

- **Problem size**
  - Function points, etc.
- **Productivity**
  - Historical data (e.g., FP/month)
- **Overall workload**
  - Boehm, COCOMO

---

---

---

---

---

---

---

---

COCOMO

Constructive cost model

$$E = a(KLOC)^b$$

$$D = c(E)^d$$

E=effort (staff-months)  
 D=development time (months)  
 a,b,c,d=organization constants  
 (e.g., 3,1.12,2.5,0.35)

---

---

---

---

---

---

---

---

Tracking Indicators

- Requirements
- Size
- Effort
- Schedule
- Quality
- Rework

---

---

---

---

---

---

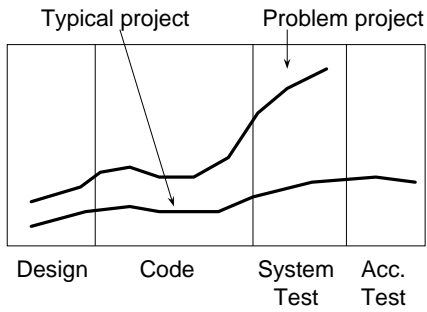
---

---

---

---

Tracking Size Estimates




---

---

---

---

---

---

---

---

---

---

SPC Terminology

- **Special cause**
  - “Temporary” performance beyond process control limits
  - Related to “stability”
- **Common cause**
  - “Persistent” change
  - Related to “capability”

---

---

---

---

---

---

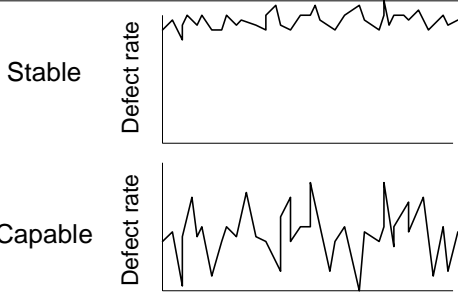
---

---

---

---

Process Performance



---

---

---

---

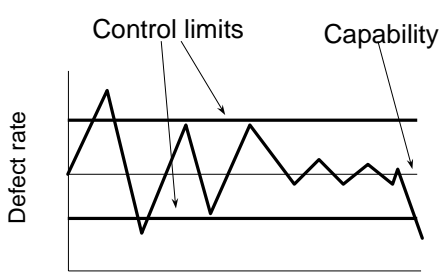
---

---

---

---

Process Maturity



---

---

---

---

---

---

---

---