

# **Establishing an Organizational Project Performance Baseline**

**David Garmus  
David Consulting Group  
1770 E. Lancaster Avenue, Suite 15  
Paoli, PA 19301**

# Presentation Topics

---

- Use Measurement to Improve Performance
- Baseline your Performance

# Measure, Measurement, Metric

---

- Measure
  - As a noun, is defined as a number that assigns relative value
    - Examples: volume, height, function points, work effort
  - As a verb, means to ascertain or appraise by comparing to a standard
- Measurement
  - Is defined as assigning relative value
  - Measures gained from this activity are combined to form metrics
- Metric
  - A combination of two or more measures or attributes
  - Examples include:
    - defect density (defects/function point)
    - productivity rate (hours/function points)

# Measurement in Relation to Other Processes

---

- Three primary uses of measurement programs:
  - Process improvement
  - Improved project management
  - Quality improvement
- Why Measure?
  - We measure to understand and improve our processes

# Measurement Frameworks

---

- Measurement frameworks help
  - Assist in planning
  - Turn measures into useful information that can be used to guide decisions
- Goal-Question-Metric (GQM)
  - Methodology for tying measurements to goals using 3 steps:
    - Step 1: Identify organizational goals
    - Step 2: Formulate a series of questions for each goal
    - Step 3: Analyze the questions to determine if a quantitative means exists to answer them

# Common Measures

---

- Common Measures
  - Size\*
  - Defects\*
  - Effort\*
  - Duration\*
  - Cost
  - Customer satisfaction

\* referred to as the “core measures” by the Software Engineering Institute (SEI)

# The Need for Measurement

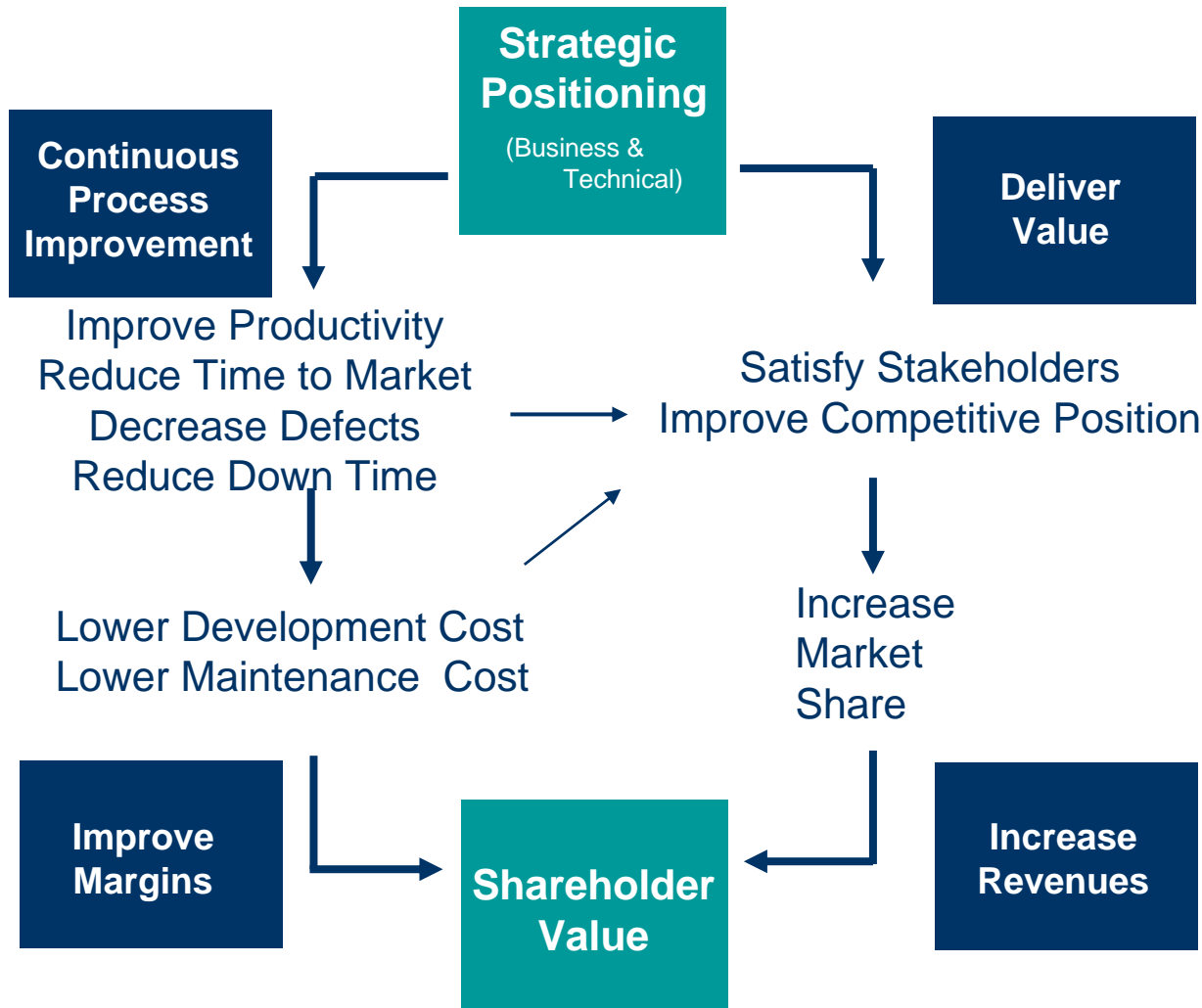
---

**FACT:** It is important to improve your ability to define, design, develop, deploy and maintain cost effective, high quality software solutions.

**ACTION:** Improvement is a continuous process which requires organizations to introduce innovation and change.

**SOLUTION:** Effectively managing change requires an ability to measure the impact of change.

# Measurement Matched To Stakeholder Needs





# Basing Metrics On Goals Of The Process Being Measured

---

## Contribution

### Business Related Measurement

Delivery Cost  
Time To Market  
Customer Satisfaction

Measure the impact of IT on the business

### Process Related Measurement

Effectiveness  
Integration  
Compliance

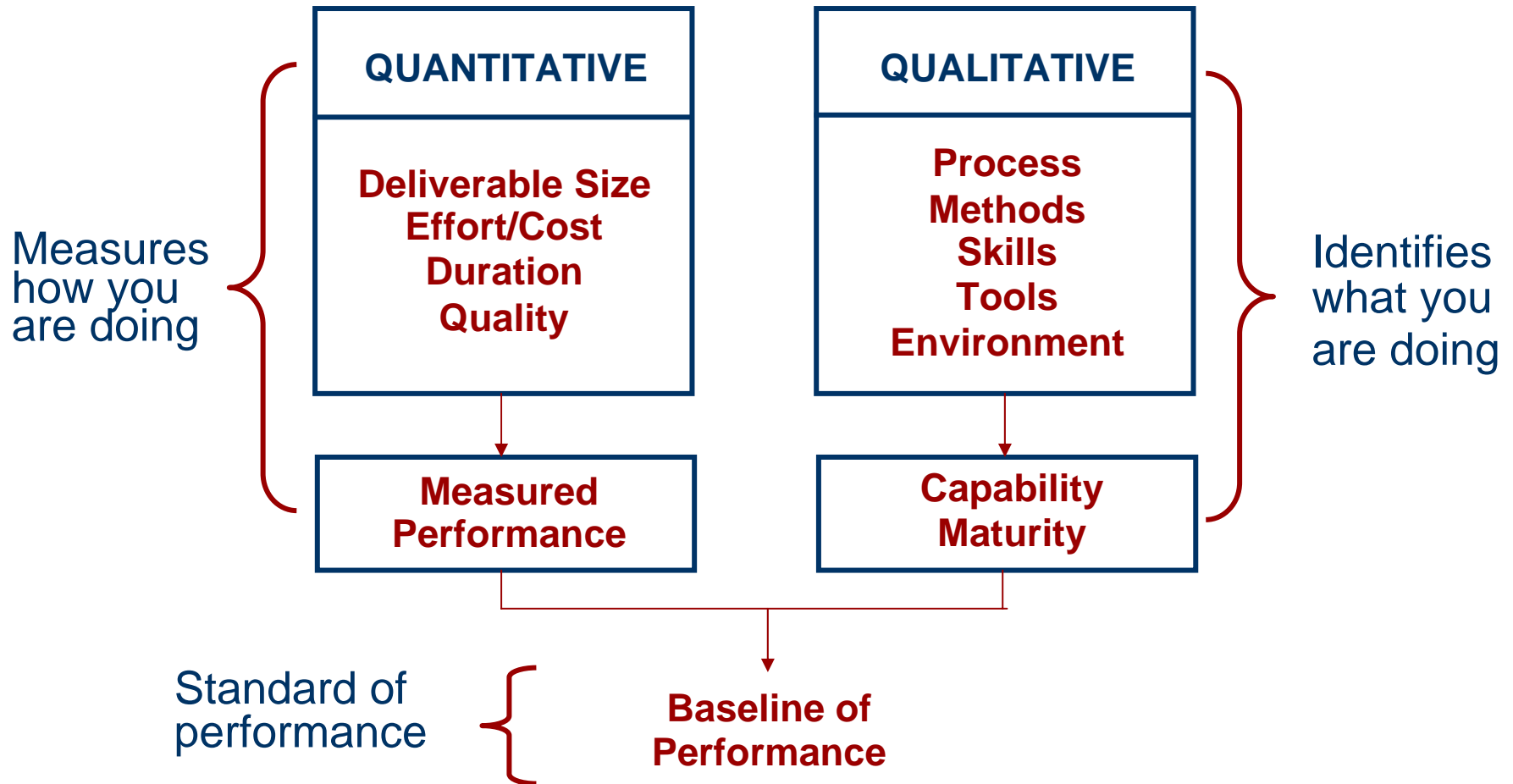
Identify trends and monitor progress in delivery & support

### Project Related Measurement

Project Tracking  
Estimating  
Change Management

Effectively utilize measurement in a pro-active format

# Using Measurement For Delivery & Support



# Presentation Topics

---

- Use Measurement to Improve Performance
- Baseline your Performance

# What is Your Level of Performance?

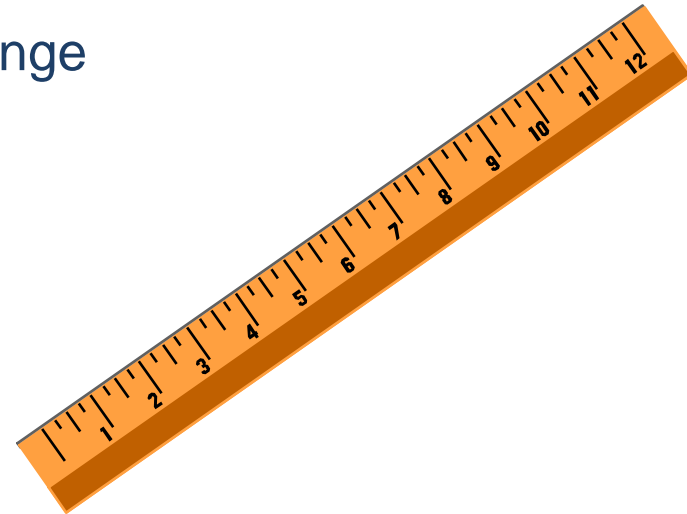
---

- Improvements resulting from current and future initiatives must be measured - ROI
- The basis for measuring improvements may include:
  - Industry data from commercial sources
  - International Software Benchmarking Standards Group (ISBSG) (<http://www.isbsg.org>)
  - Software Engineering Institute (SEI)
  - Organizational baseline data
- It is necessary for the organization to put a “stake in the ground” relative to current performance level in order to improve development practices

# Purpose Of Baselineing

---

- A Continuous, Methodical Process for Comparing Performance Levels Among Projects, Divisions or Organizations
- Monitoring Improvements and Change
- Representation and Realization of Best Practices



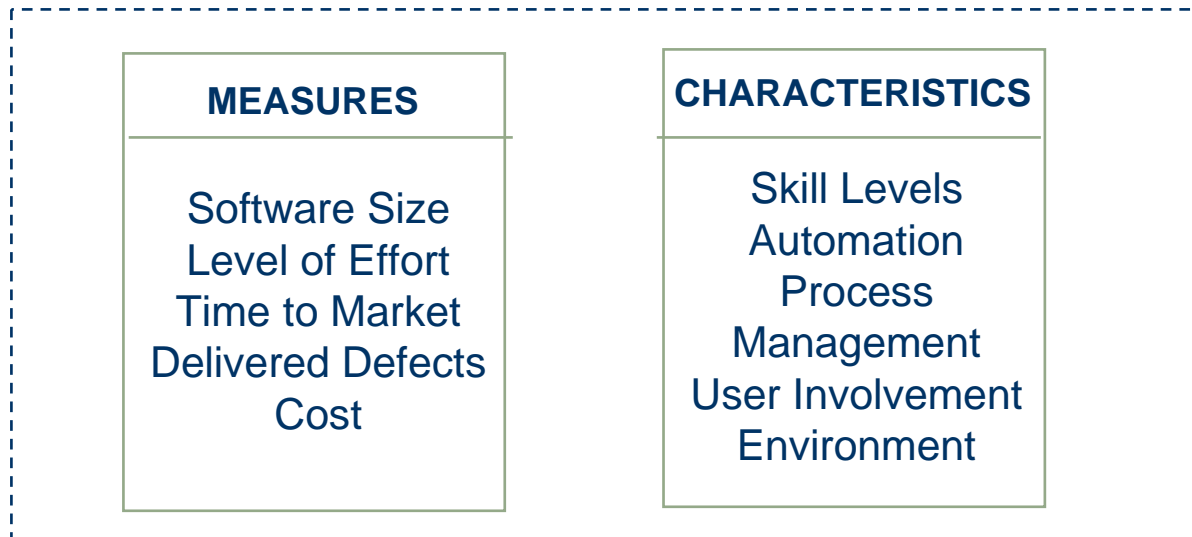
# Collecting & Reporting

---

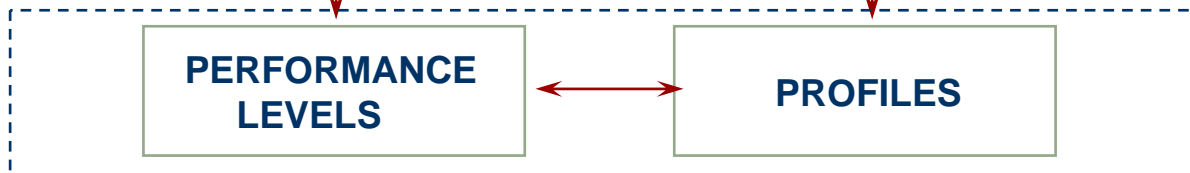
- Identify data set (typically project or portfolio oriented)
- Collect baseline data
  - Project measures (e.g., size, effort, cost, duration, defects)
  - Project attributes (e.g., skill levels, tools, process, etc.)
- Analyze data
  - Performance comparisons (identification of process strengths and weaknesses)
  - Industry averages and best practices
  - Performance modeling (identify high impact areas)
- Report results

# Quantitative & Qualitative Assessments

## Research



## Analysis



## Results

- Correlate Performance Levels to Characteristics
- Substantiate Impact of Characteristics
- Identify Best Practices



# Quantitative Data

Project	Start Date	End Date	FP	Effort Months	Schedule Months	Cost	Delivered Defects
<b>Project abc</b>	8/18/03	1/9/04	122	24.08	4.75	\$375,600	17
<b>Project xyz</b>	3/15/03	12/10/03	111	8.63	8.75	\$134,640	1
<b>Project 123</b>	5/27/02	5/9/03	83	25.77	11.50	\$401,958	3
<b>Project 890</b>	8/15/03	10/19/03	52	5.50	2.25	\$85,800	0

Project data is collected for:

Start and end date – duration

Size – expressed in function points

Effort – labor

Cost – predominately labor

Defects – pre and post implementation



# Application and Project Attributes

---

- Gather the core attributes for the measured applications
- Establish a standard set of attributes for the organization
  - May include well defined and objective characteristics (e.g., development language) or subjective and loosely defined characteristics (e.g., relative project complexity)
- These attributes can be classified into a variety of categories, such as:
  - General Project/Application Characteristics
  - Resources
  - Project and Process Management
  - Technology and Environmental Factors(These characteristics are discussed on the next slides)

# Defining Application and Project Attributes

---

- General Project/Application Characteristics
  - Project type (new development, enhancement, maintenance software release or upgrade, acquisition, conversion, replacement, support)
  - Project characteristics (risk assessment, in-house developed versus purchased and customized, outsourced)
  - Application type (real-time, data warehouse, financial, telecomm)
  - Architecture (main frame, client/server, web based, PC)
  - Degree of innovation (whether the application has been done previously by your organization, within your industry or at all)
  - Relative project complexity (low, medium, high)
  - System performance requirements (speed, security)
  - Project team (developer & user - person count, skill levels, etc.)
  - Organization (number of people, sites, countries, languages, etc.)

# Defining Application and Project Attributes

---

- Resources
  - Technical experience (e.g., level of project team experience with tools, language, methodology)
  - Business experience in functional area (e.g., banking, inventory control, weapons, taxes, telecommunication, etc.)
  - Users (e.g., degree of involvement, experience level with application subject area, familiarity with the system development process, prior experience with a specific hardware or software platform)
  - Support (e.g., support staff service levels)
  - Software developer (e.g., in-house, outsourced or a combination)
  - Training (e.g., included or excluded in project, conducted by IT or users)

# Defining Application and Project Attributes

---

- Project and Process Management
  - Methodology (e.g., Structured Analysis/Design, Prototype, Information Engineering, Object-Oriented, Agile)
  - Project management approach
  - Modeling techniques
  - Standards used (e.g., IEEE, ISO, CMMI)
  - Percentage of reuse (code, design, test cases, etc.)
  - Release strategy (e.g., phased, "x" releases per year, prototype, evolutionary, rolled out)
  - Project structure (e.g., project team throughout versus analysts handing off to programmers)

# Defining Application and Project Attributes

---

- Technology and Environmental Factors
  - Database management systems (flat, indexed, hierarchical, network, relational)
  - Number of database management systems
  - Development platform (new, good response time, access)
  - Physical environment (space, temperature, lighting, noise level)
  - Testing and debugging tools
  - Automated testing tools
  - Code analysis tools
  - Configuration management tools
  - Development language(s) (COBOL, C++, Java, .Net, HTML, etc.)
  - Operating system(s)
  - Communications/Networking requirements (none, LAN, WAN, Web)
  - Organization structure

# Identify Key Performance Factors

Management	Definition	Design
<ul style="list-style-type: none"> <li>• Team Dynamics</li> <li>• Morale</li> <li>• Project Tracking</li> <li>• Project Planning</li> <li>• Automation</li> <li>• Management Skills</li> </ul>	<ul style="list-style-type: none"> <li>• Clearly Stated Requirements</li> <li>• Formal Process</li> <li>• Customer Involvement</li> <li>• Experience Levels</li> <li>• Business Impact</li> </ul>	<ul style="list-style-type: none"> <li>• Formal Process</li> <li>• Rigorous Reviews</li> <li>• Design Reuse</li> <li>• Customer Involvement</li> <li>• Experienced Development Staff</li> <li>• Automation</li> </ul>
Build	Test	Environment
<ul style="list-style-type: none"> <li>• Code Reviews</li> <li>• Source Code Tracking</li> <li>• Code Reuse</li> <li>• Data Administration</li> <li>• Experienced Staff</li> <li>• Automation</li> </ul>	<ul style="list-style-type: none"> <li>• Formal Testing Methods</li> <li>• Test Plans</li> <li>• Staff Testing Experience</li> <li>• Effective Test Tools</li> <li>• Customer Involvement</li> </ul>	<ul style="list-style-type: none"> <li>• New Technology</li> <li>• Automated Process</li> <li>• Adequate Training</li> <li>• Organizational Dynamics</li> <li>• Certification</li> </ul>

# Software Practices Profile

Profile scores reflect the 'goodness' of the development practices for a given project. Six categories are evaluated and scored. The higher the score the higher the probability of a success.

<u>Project</u>	<u>Profile Score</u>	<u>Mgmt</u>	<u>Def</u>	<u>Des</u>	<u>Build</u>	<u>Test</u>	<u>Env</u>
Project 1	75.9	84.09	76.92	77.27	65.38	81.25	65.38
Project 2	60.3	68.18	64.10	56.82	61.54	59.38	50.00
Project 3	60.0	61.36	33.33	81.82	57.69	65.63	65.38
Project 4	54.5	61.36	64.10	45.45	61.54	50.00	42.31
Project 5	39.4	68.18	20.51	50.00	53.85	25.00	46.15
Project 6	35.3	38.64	20.51	15.91	69.23	43.75	38.46
Project 7	31.4	36.36	30.77	4.55	42.31	43.75	46.16

# Strengths & Opportunities (An Example)

---

## Definition

### Strengths

- Requirements are clearly stated and stable
- Development and customers are experienced in applications

### Opportunities for Improvement

- More formal requirements gathering process on larger projects
- More consistent use of prototyping on larger projects
- A formal review process

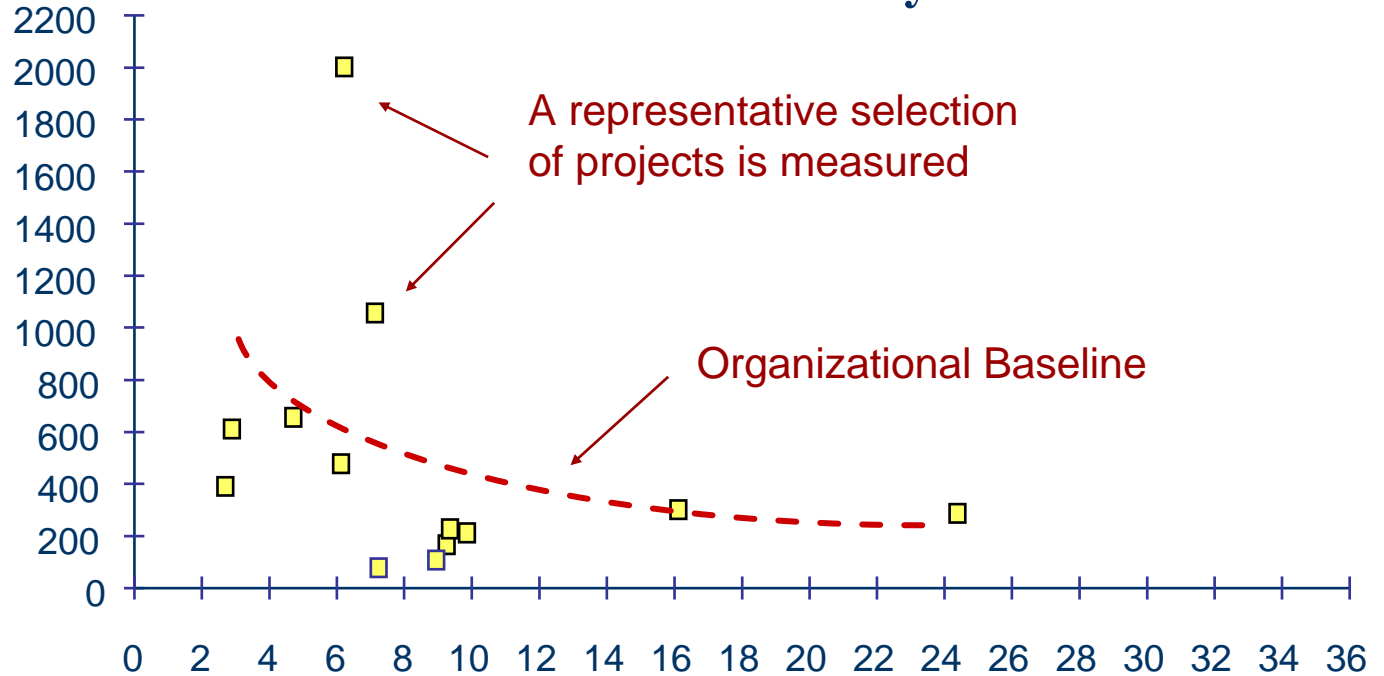


# Establish a Baseline

Size is expressed in terms of functionality delivered to the user

**Software Size**

## Performance Productivity

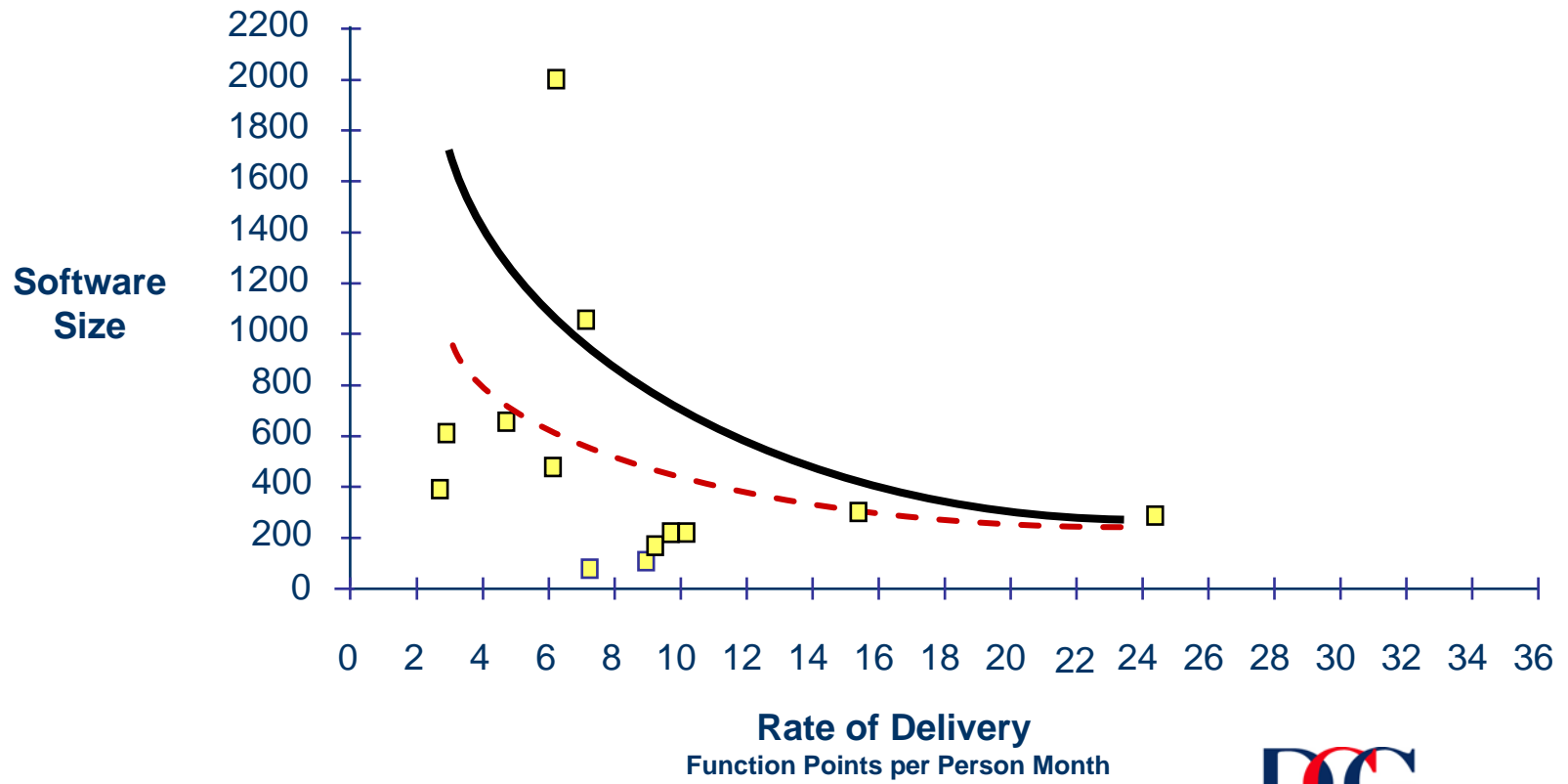


Rate of delivery is a measure of productivity

**Rate of Delivery**  
Function Points per Person Month

# Compare to Industry Benchmarks

## Industry Baseline Performance



# Function Points Per Person Month (130 hours) for Adaptive Maintenance (Enhancements)

---

## Average of Recent Enhancement Projects Across Different Platforms

Client Server	17
Main Frame	13
Web	25
e-business Web	15
Vendor Packages	18
Data Warehouse	9

# Question/Answer Session

---

- Any questions?

[dg@davidconsultinggroup.com](mailto:dg@davidconsultinggroup.com)  
[www.davidconsultinggroup.com](http://www.davidconsultinggroup.com)