INTRODUCTION TO SOFTWARE ENGINEERING

Dr.Lê Hùng Tiến

VLU

Introduction To Software Engineering

Carnegie Mellon University
The Practical Software Engineering Series

The Software Measurements & Metrics

Course Objective

□ Upon completion of this course, students will have the ability to:

- Understand the discipline and principles of Software Engineering.
- Understand the evolution of software in industry and the global competitive trends.
- Understand software process, product and services.
- Understand software modeling & techniques.
- Demonstrate an appreciation for the breadth of software engineering.

Lecture Learning Objectives

- Upon completion of this lecture, students will be able to:
  - Understand software measurements and metrics.
  - Understand data gathering and analysis.
- Outcomes:
  - Be able to distinguish measures, metrics and indicators.
  - Be able to collect and analyze data for projects.
  - Use metrics to make decisions.
  - Understand the use of metrics in business.

Measurements In Industry

- Measurement is fundamental to many disciplines:
  - Economics
  - Business
  - Medical
  - Aerospace
  - Engineering
    - Civil Engineering
    - Mechanical Engineering
    - Hardware Engineering
    - Software Engineering: A luxury?

- Software measurements are not very well understood or applied consistently in software projects.
- Many software project managers do not receive adequate training in application measurements of projects or using data to make decisions.

Why Software Measurements?

- Do you know:
  - How much does your software project cost?
  - How long will your software project take?
  - How reliable is your software product?
  - How many defects are in your software product?

- Have you ever asked why:
  - Software costs and schedule are not accurately estimated?
  - Software is shipped full of defects?
  - Software work is high stress and requires long hours?
  - Software development incurs large expenses?
  - Project status is difficult to get?

The 2004 Software Survey

- □ In 2004, the U.K Software Engineering Group conducted a survey of over 800 software project managers and Chief Information Officers asking:
  - How much is your company spending on software?
  - What percentage of your costs arise from unplanned rework?
  - What percentage of your projects are on time and on budget?
  - How confident are you that your current projects will perform according to their estimates?
  - How does the skills of your project team compare to industry averages?
    - □ Above? Same? Or Below?
  - How satisfied are customers of your software?
    - □ Very satisfied? Satisfied? Not Satisfied?

- □ Most can not answer these questions, they simply don’t know.

History Of Software Measurements - 1

The Early Years	Measure
1950 -1970	Line Of Code (LOC)
	Go-to Statements
	Error Counts
	Nesting
1970 – 1980	Complexity
	Structure
	Module Size
	Cohesion/Coupling

During these years, measurements and metrics were not formally used in projects unless required. Most data collection was not well defined or consistent, and it was management’s responsibility to define the process.

History Of Software Measurements - 2

The Design Years

1980 -1990

Measure

Function Points
Graph Theoretic
Complexity
Feature Points
Complexity
Process Maturity
Object/Class of Object

The “Show Me” Years

1990 -2005

Measure

Functional Quality
Technical Quality
Quality Attributes
Productivity
Business Values
Balance Scorecard

Measurements

- You cannot:
  - Improve your business without a baseline.
  - Establish a baseline without measurements.
  - Create measurements without understanding the process.
  - Control what you cannot measure.

- Measurement is the first step to:
  - Understanding the software process.
  - Controlling the software project.
  - Predicting the quality of the software product.
  - Improving the value of the software organization.
  - Improving business performance.
  - Maturing in the engineering of software disciplines.

Current Measurements

- Businesses need a balanced scorecard that will measure the drivers of shareholder value, as well as shareholder value itself. A good balanced scorecard should focus on:
  - Financial perspective: High level finance measures (shareholder value) are appropriate for both internal and external audiences.
  - Customer satisfaction: Overall performance of the organization in areas of greatest importance to the customer.
  - Internal process: The performance of the internal business processes which has the greatest impact on customer satisfaction.
  - Organization capability: The ability of the organization to continually innovate and improve, both internally and in the market place.

Example: A Fast Growing Software Company

- The critical success of this kind of company is its ability to innovate, sense changes in the market, and develop new products in a timely basis. This is reflected in its measurements:
  - Financial perspective: Increase market share as the result of continuous introduction of new products.
  - Customer satisfaction: Ability to provide earlier functionality in new products.
  - Internal process: Faster development cycle, quick response time and less employee turnover.
  - Organization capability: Innovation in the introduction of each new product.

Example: A Integrated Supply Company

- The critical success of this kind of company is its ability to quickly respond to customer orders, minimizing order fulfillment cycles. This is reflected in its measurements:
  - Financial perspective: Increase revenue, reduce cost.
  - Customer satisfaction: Ability to reduce cycle time on the supply side.
  - Internal process: Faster supply cycle, quick response time and inventory turn around.
  - Organization capability: Process improvement cycle time and quicker response time.

Class Discussion

- Assume that you work for a company (use your imagination) and you need to define business objectives.
- Process:
  - List all of your customers.
  - List all of the products and services you provide to each customer.
  - What products/services do you provide (giving real added value from a customer point of view) that differentiates you from your competitors?
  - What are your strengths and weaknesses?
  - Where is the greatest potential for improvement?
  - What issues can get in your way?

**Using all of the above information,
what are the few things you need to achieve
to be successful as measured by your customers?**

Definition

□ **Measure:** A standard or unit of measurement – The extent dimensions, capacity, etc. – of anything, especially as determined by a standard.

For example:

Number of defects

Source lines of code (SLOC)

□ **Metric**: A calculated or composite based on two or more measures; a quantified measure of the degree to which a system, component, or process possesses given attributes.

For example:

Number of defects per thousand source lines of code (KSLOC).

Metrics

- A combination of two or more measures make up a metric.

- A combination of two or more metrics gives the information meaning.

  - <u>For Example</u>: 10 defects/KSLOC – Is it Good? Bad?

Last project	Current project
15 defects/KSLOC	10 defects/KSLOC

Release 1	Release 2	Release 3
12 defects/KSLOC	8 defects/KSLOC	13 defects/KSLOC

- Comparing metrics from projects or baseline gives meaning to the metric.

Indicator

- **<u>Indicator</u>**: A measure or combination of measures (metric) that provides insight into a software issue or concept;
- For example:
  - Number of Defects/KSLOC = indicator of customer satisfaction
  - Number of Defects/KSLOC = indicator of product quality
- Software Engineers collect measures and develop metrics so that indicators will be obtained.
- Project indicators allow project managers to assess the status of a project, track potential risks, uncover problems, adjust workflow or tasks and evaluate overall quality.
- Process indicators allow management to review the defined process on what works and does not work for improvement measures.

Metrics

- Metrics are used to measure the progress toward meeting an objective.
- Metrics must be tied to business objectives and exist to measure progress that must include goals and a timeframe.
- Metrics must be clearly and consistently defined before used to avoid confusion or different interpretations.
- Metrics must focus attention on the few actions that will make a difference in a project, product, process, organization or business objectives.
- Management reviews must use metrics for decision making.
- Action items must be executed when progress doesn’t match expectations.

Metric Targets

- Processes: A collection of software-related activities.
  - For example: Requirements, Design, Code, Test, Release

- Products: Artifacts, deliverables or documents that result from a process activity.
  - For example: Design Document, Code, Test Cases

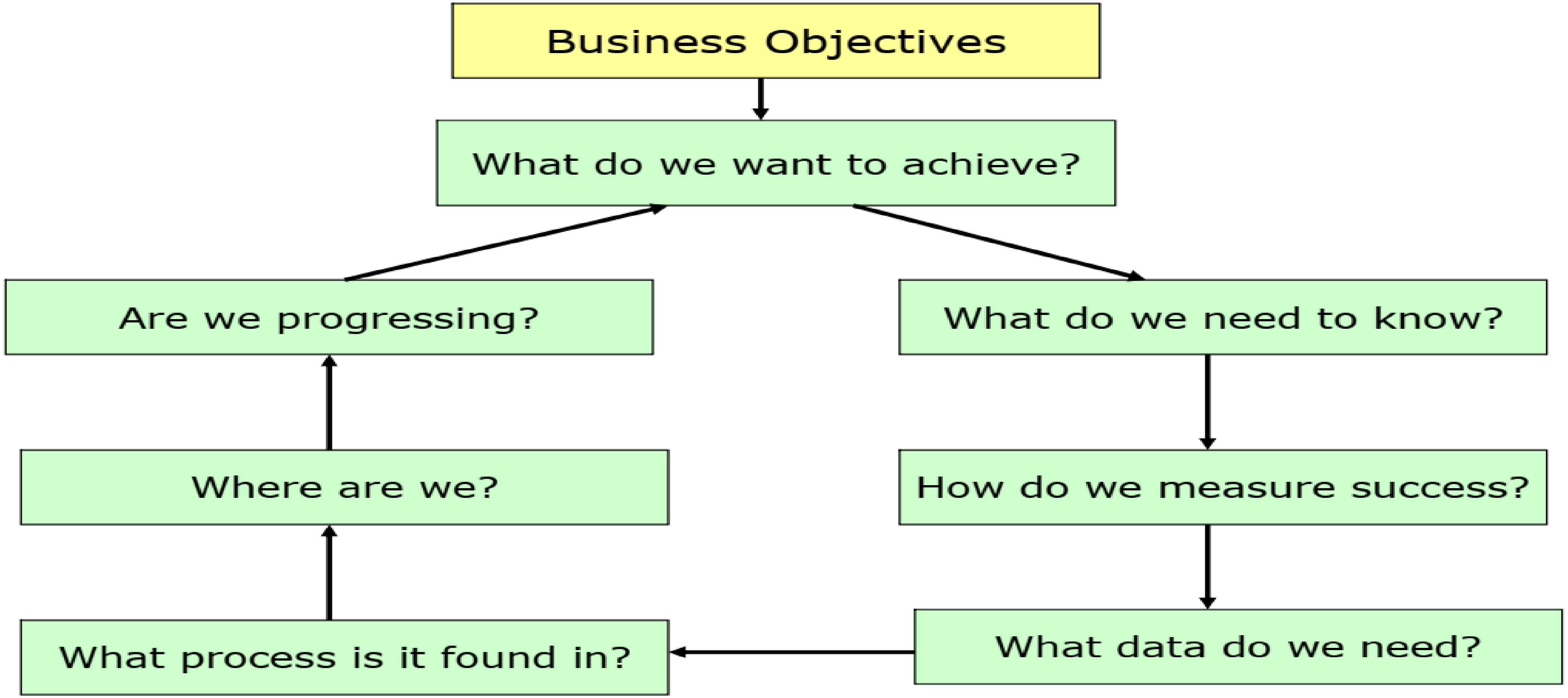
- Resources: Entities required by a process activity.
  - For example: People, Teams, Tools

Metrics

- To be useful, metrics should be:
  - Gathered for a specific purpose
  - Explicitly defined
  - Properly used
  - Properly managed
  - Carefully analyzed (confidential aspect)

- Managers must ask:
  - What data should be collected?
  - What data should be analyzed and used?
  - How do we collect and analyze data?
  - How do we get practitioners to adopt measurement?
  - How do we get managers to use data to make decisions?

The Defined Process For Metrics



Example – 1

- **Objective**

  Increase management by facts & data 50% by year end.
- **What do we want to achieve?**

  Increase management decisions based on measurements provided and record them in decision log book.
- **What do we need to know?**
  - How much has management decisions increased based on measurements?
  - How many measurements are provided to management ?
- **How do we measure success?**
  - Comparison of management decisions based on measurements to those prior to having measurements (Baseline = 0%).
  - Percentage of management using measurements for decision making.

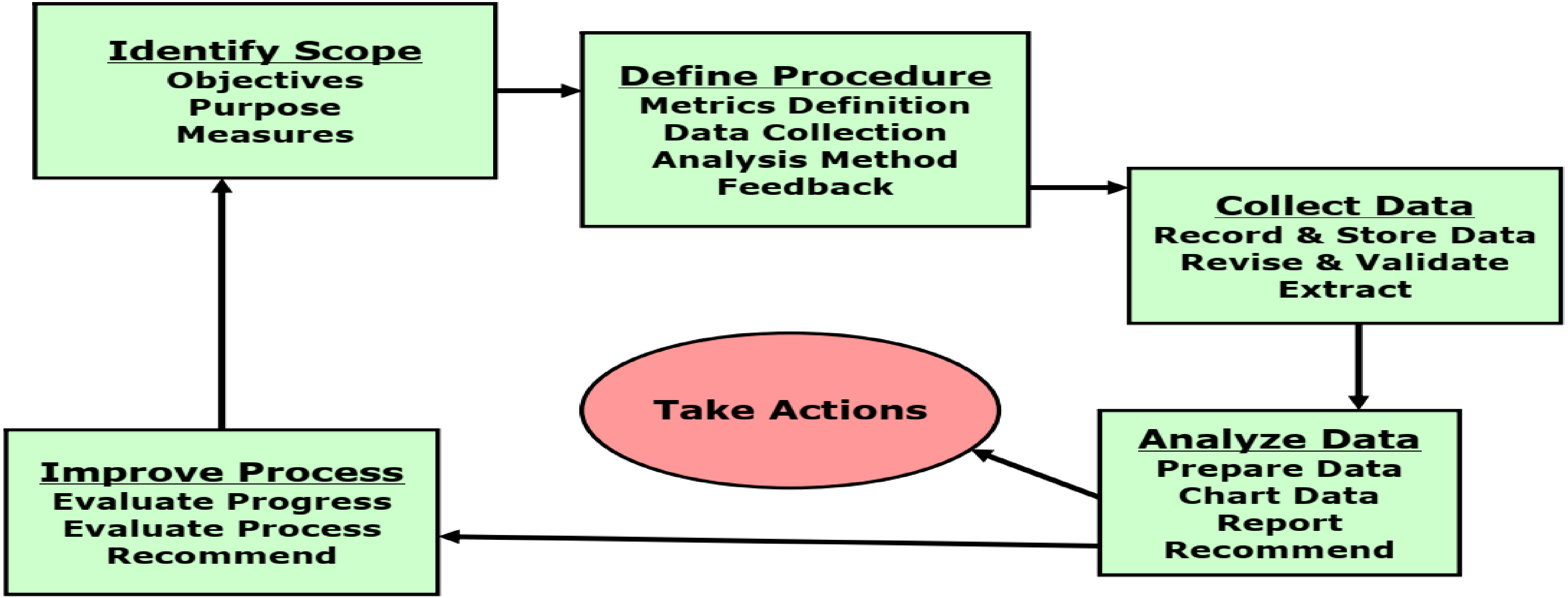
Example – 1

- **Objective**

  Increase management by facts & data 50% by year end.
- **What do we want to achieve?**

  Increase management decisions based on measurements provided and record them in decision log book.
- **What do we need to know?**
  - How much has management decisions increased based on measurements?
  - How many measurements are provided to management ?
- **How do we measure success?**
  - Comparison of management decisions based on measurements to those prior to having measurements (Baseline = 0%).
  - Percentage of management using measurements for decision making.

Practical Metrics Implementation



Identify Scope: Project

- Purpose
  - Management needs better insight into project's progress
- Objectives
  - Manage project according to plan
  - Reduce defects to the customer
- Measurements
  - Schedule estimates vs. actuals
  - Effort or Size (Normalizer)
  - Project's defect
    - Post-Released
    - Pre-Released
  - Cycle Time
  - Customer Satisfaction
  - Number of management decisions reported based on metrics

Data Collection

- Software measurement is only as good as the data collected and analyzed.
    - What Is Good Data?
        - Correctness
        - Accuracy
        - Precision
        - Consistency
    - Data Collection Process
        - Forms (manual)
        - Tools (automated)
        - Web based

Good Data

□ Correctness: Data collected according to the rules of the definition of the metric.
Example: Rule for counting code (LOC).

□ Accuracy: Data collected may not be the same as the actual value.
Example: Time collected using analog clock may not be the same as a digital clock.

□ Precision: Number of decimal places needed to express the data.
Example: Project time reported in hours, not minutes or seconds.

□ Consistency: Data should be collected the same way so it can be compared.
Example: Comparing size between projects using same rule for size counting.

Define Procedure: Project Data

□ Project managers will collect the following project data:

Sizing:	Volume of codes, artifacts, and paperwork
Estimating:	Efforts, costs, and schedules
Planning:	Schedules and milestones
Tracking:	Progress against plans & estimates
Quality:	Errors, Defects, other quality attributes

□ <u>Type of decision</u>

When to collect:	Monthly, weekly, during reviews
Who collects:	Technical lead - issues with bias, data access, cost, availability, motivation
How to collect:	Tools available, procedures and forms used in the project

Metric Example

□ Measurement Data:

Project Name	Size LOC	Effort Person/Months	Cost $(x1000)	Pre-released Defect	Post-released Defect
ABC	12,000	12	20	134	24
XYZ	40,000	36	80	456	83
EFH	32,000	32	76	387	76

□ Metrics:
- □ Defect/Size
- □ Cost/Size
- □ Effort/Size

Analyze Defect Types

1.	Documentation	- Comments, message
2.	Syntax	- Spelling, punctuation, typos, formats
3.	Build/Package	- Change management, version control
4.	Assignment	- Declaration, duplicate names, scope, limits
5.	Interface	- Procedure call or reference, I/O, user formats
6.	Checking	- Error message, inadequate checks
7.	Data	- Structure, content
8.	Function	- Logic, pointers, loop, recursion, computation
9.	System	- Configuration, timing, memory
10.	Environment	- Design, compile, test, other support system problems

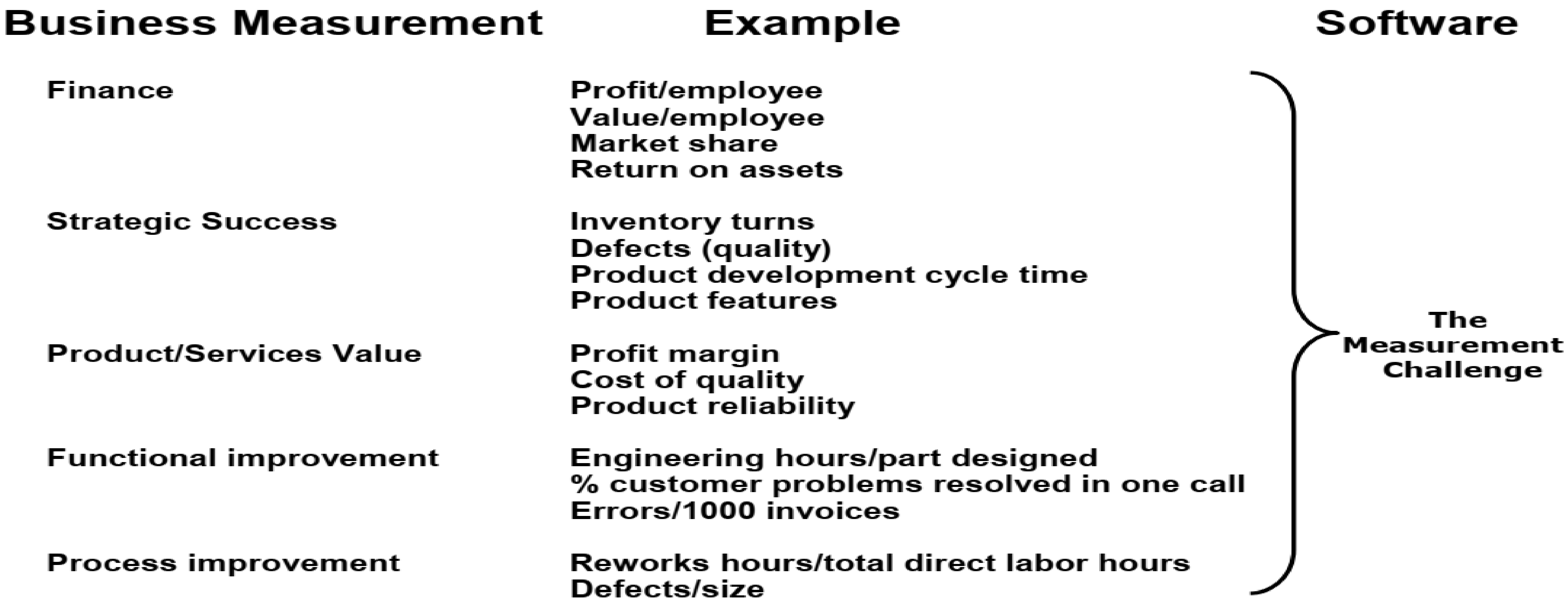
Metric Program

- Metric programs are expensive because of:
  - Training and collection tool development cost.
  - Staff resources to support data collection and analysis.
  - Need to justify investment in measurements & metrics program.
- Benefits:
  - Allows engineers and managers to focus on software quality & process.
  - Data helps people understand problems earlier in the life cycle and the severity of problems.
  - Metrics help organizations establish baselines and focus on actions with quantitative results.
  - Measurements provide data that lead to information and eventually knowledge about the organizational processes.
  - Project managers learn from mistakes from previous projects and take action to avoid them.
  - Continuously improving to reduce cost, increase quality, productivity and customer satisfaction.

The Business Of Software

- □ **Quality**: In the past, quality has been viewed as a differentiator. Now this is a given – most companies do measure quality.
- □ **Time to market**: Most companies today focus on faster time to market by compressing the time to design, build, deliver, and service new products. Business re-design, technology changes and process improvement have provided many companies with order of magnitude improvements in cycle time.
- □ **Services**: Value-added services are now included in products. The trend toward a service economy is placing a greater emphasis on service as a differentiator.
- □ Most measurement programs focus on the internal technical part of software engineering unrelated to business.
- □ A good software engineer understands the need to bridge the gap by linking measurement to the business performance.

Business-Oriented Measurements



Business Measurement	Example	Software
Finance	Profit/employee Value/employee Market share Return on assets	The Measurement Challenge
Strategic Success	Inventory turns Defects (quality) Product development cycle time Product features	
Product/Services Value	Profit margin Cost of quality Product reliability	
Functional improvement	Engineering hours/part designed % customer problems resolved in one call Errors/1000 invoices	
Process improvement	Reworks hours/total direct labor hours Defects/size	

Summary

- Measurement enables managers and software engineers to improve the process by assisting in planning, tracking, and control of the software project, as well as assessing the quality of the product.
- Product quality is one of the key measures and processes of a software project.
- Measurements, metrics, data collection, data analysis and the use of metrics in projects, organizations and business must be clearly defined in the Define Process.
- To be useful, software metrics should be simple and computable, persuasive, consistent and objective.
- In a competitive world, all products and services must be tied to the business goals.
- Organizations must explicitly define a quality program that utilizes metrics for quality and business advantage.