MBASE and CMMI II

Ray Madachy

CS 577b

April 7, 2003

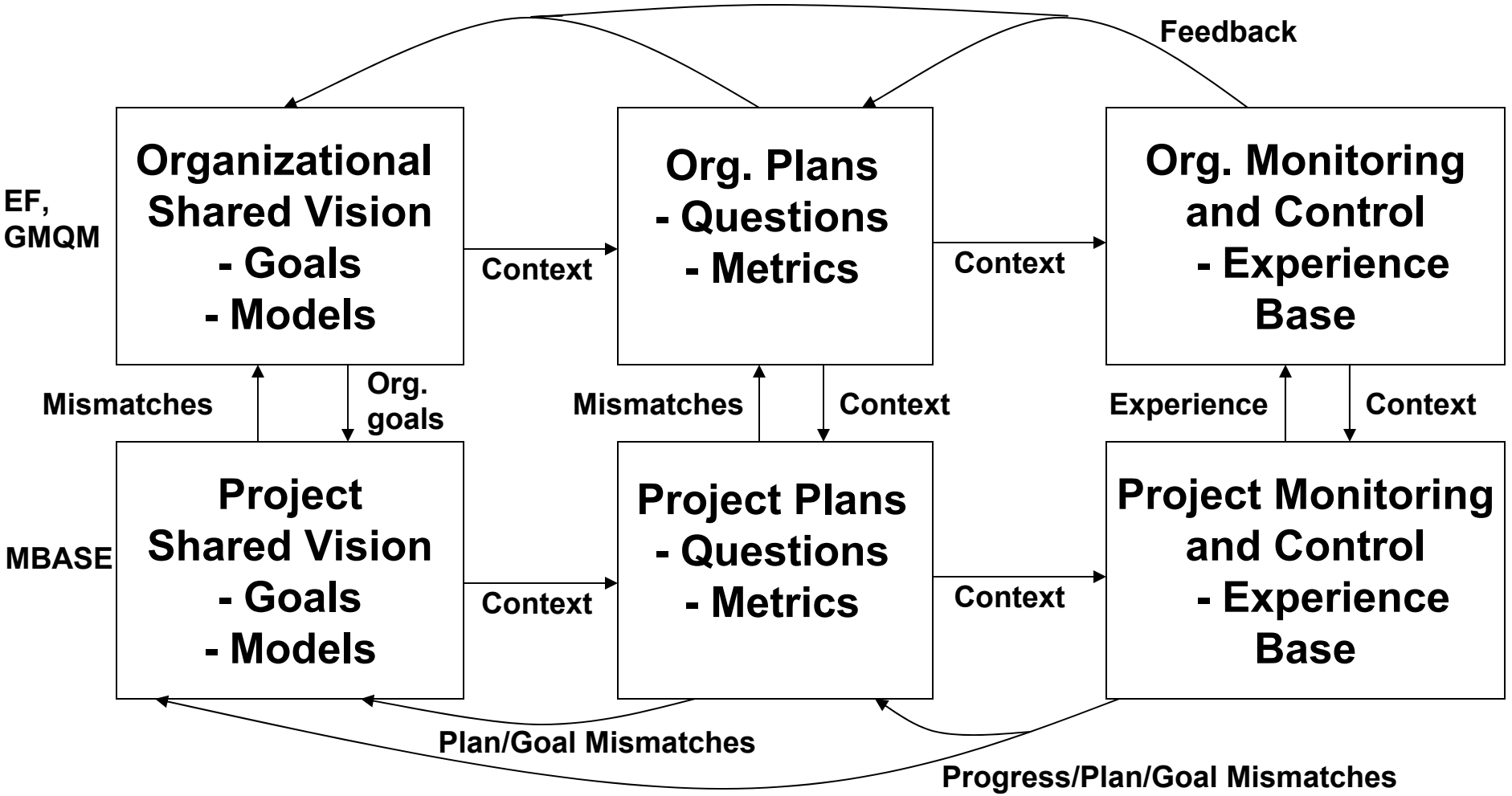
Outline

- **Review of MBASE and CMMI I**
- **Metrics, models and process analysis**
- **CMM Level 4-5 process areas**

MBASE and CMMI I Review

- CMM-I framework for software and systems process maturity
- MBASE project-level process implementation
- CeBASE experience factory organization-level process implementation

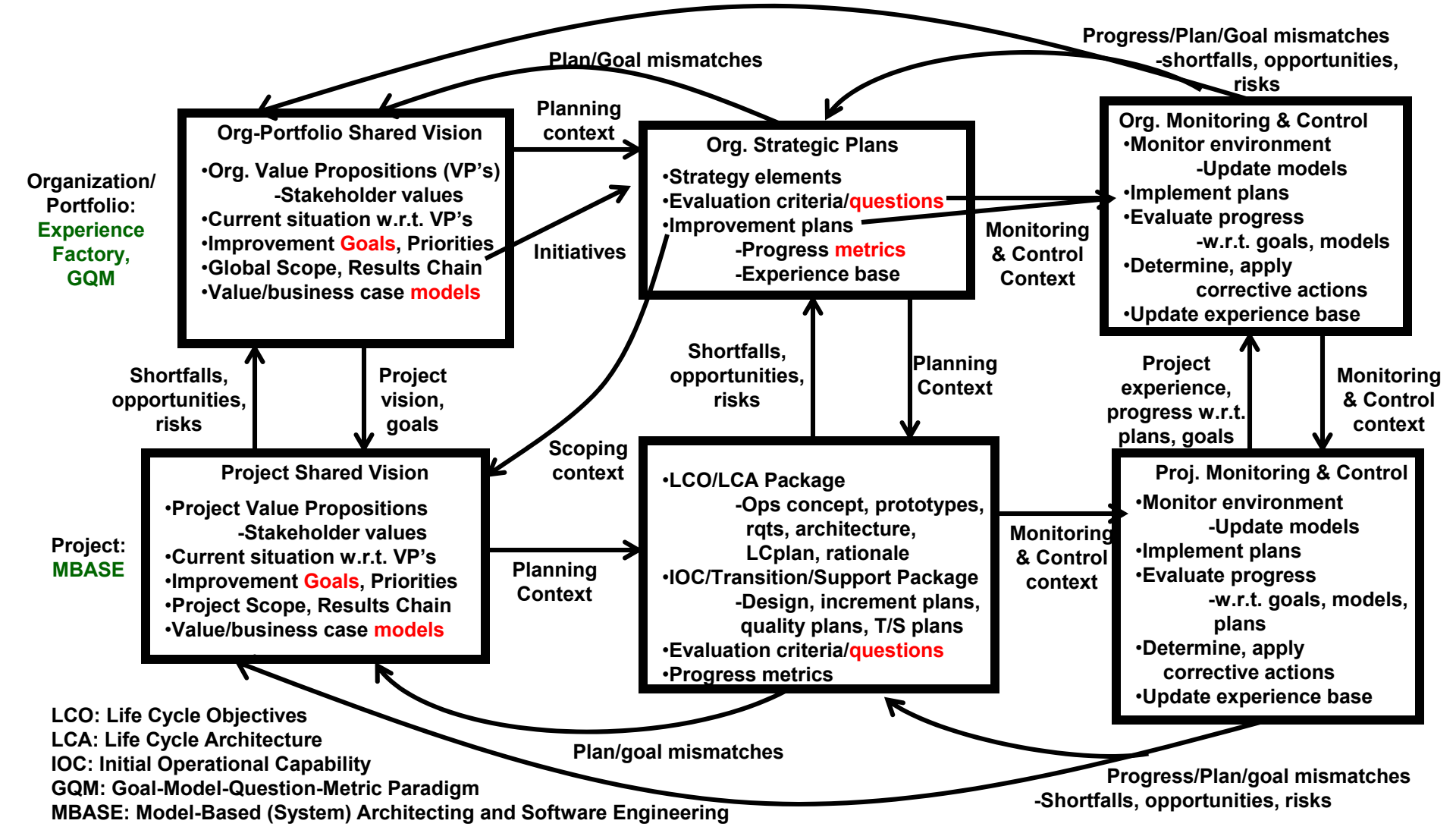
The CeBASE Method = MBASE + Experience Factory



University of Southern California
Center for Software Engineering

The CeBASE Method

-Applies to organization's and projects' people, processes, and products



LCO: Life Cycle Objectives
LCA: Life Cycle Architecture
IOC: Initial Operational Capability
GQM: Goal-Model-Question-Metric Paradigm
MBASE: Model-Based (System) Architecting and Software Engineering

Outline

- **Review of MBASE and CMMI I**
- **Metrics, models and process analysis**
- **CMM Level 4-5 process areas**

Metrics Organizational Guidelines

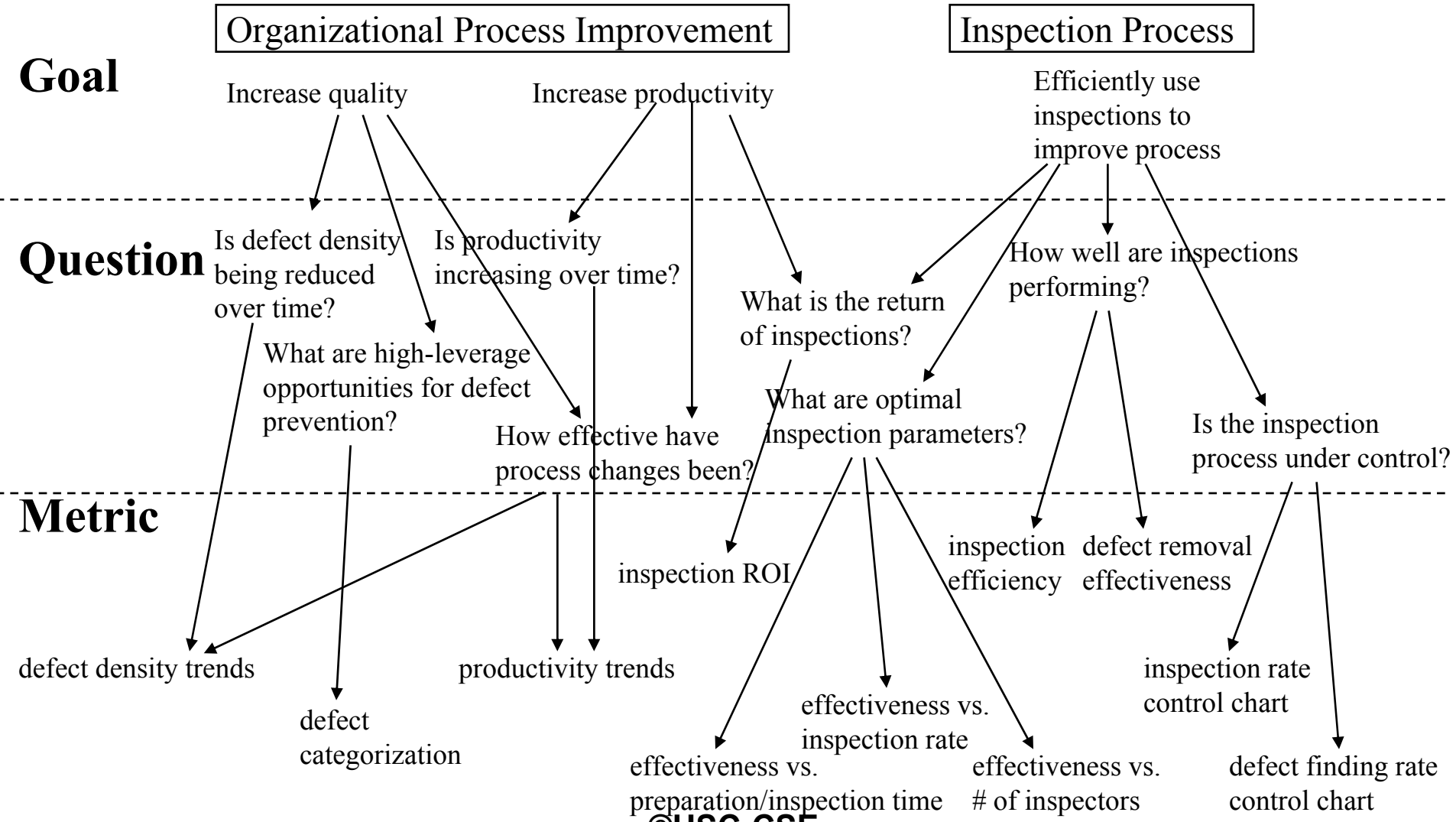
Functional Management	1. Don't allow anyone in your organization to use metrics to measure individuals. 2. Set clear goals and get your staff to help define metrics for success. 3. Understand the data that your people take pride in reporting; don't ever use it against them; don't ever even hint that you might. 4. Don't emphasize one metric to the exclusion of others. 5. Support your people when their reports are backed by data useful to the organization.
Project Management	6. Don't try to measure individuals. 7. Gain agreement with your team on the metrics that you will track, and define them in a project plan. 8. Provide regular feedback to the team about the data they help collect. 9. Know the strategic focus of your organization and emphasize metrics that support the strategy in your reports.
Project Team	10. Do your best to report accurate, timely data. 11. Help your managers to focus project data on improving your processes. 12. Don't use metrics data to brag about how good you are or you will encourage others to use other data to show the opposite.

from [Grady 92]

Goal-Question-Metric Paradigm

- Goal-Question-Metric (GQM) is a framework for developing a metrics program
- Steps:
  - generate a set of organizational goals
    - what do you want to improve?
  - derive a set of questions relating to the goals
    - answers provide visibility into meeting the goals
  - develop a set of metrics needed to answer the questions

Sample GQM Roadmap



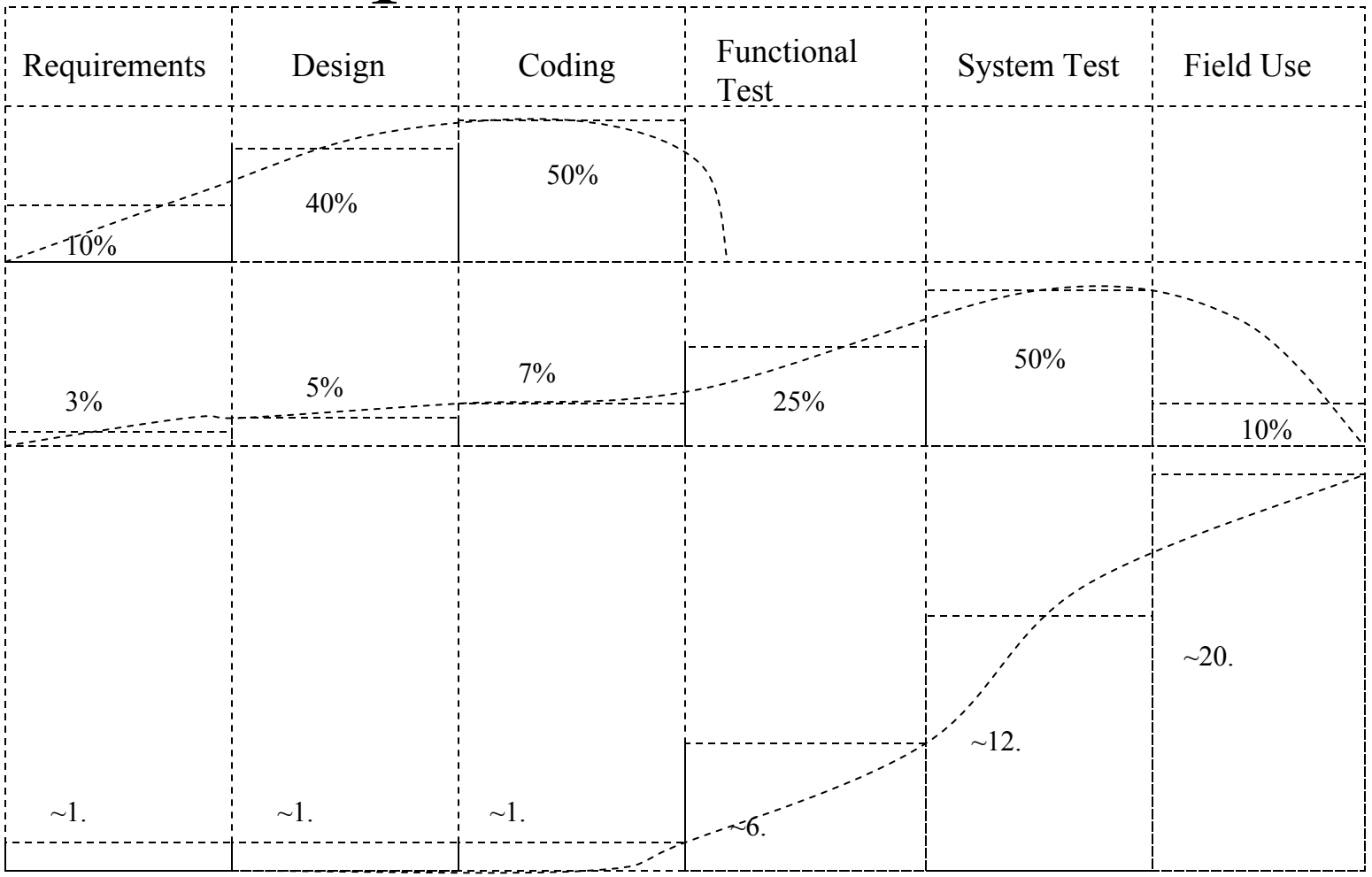
Use of Quantitative Models

- Necessary in order to develop process goals, plans (including indicator trends), and define appropriate limits.
- Encapsulate our understanding of development processes (and support organizational learning).
- Benchmark process improvement since models are calibrated to our process environment.
- Examples:
  - productivity - COCOMO, COCOTS, other cost models
  - defect density and defect rates - COQUALMO defect introduction and removal model, system dynamics process model
  - business value models
  - any process parameter - statistical variation properties of actuals, Delphi poll results

Defect Analysis

- Defect: any flaw in the specification, design, or implementation of a product.
- Facilitate process improvement through defect analysis
  - defect categorization to identify where work must be done and to predict future defects
  - causal analysis to prevent problems from reoccurring

Sample Fault Distribution



Process Maturity Fault Distributions

Process Maturity Level	Requirements	Design	Coding	Functional Test	System Test	Field Use	Phase
	10%	40%	50%				Fault Introduction Distribution
5	5%	20%	40%	20%	10%	<5%	
4	3%	12%	30%	30%	20%	5%	
3	0%	2%	20%	38%	32%	8%	Fault Detection Distribution
2	0%	0%	3%	30%	50%	17%	
1	0%	0%	2%	15%	50%	33%	
	1	1	1	6	12	20	Relative Fault Cost

Defect Flow Tracking

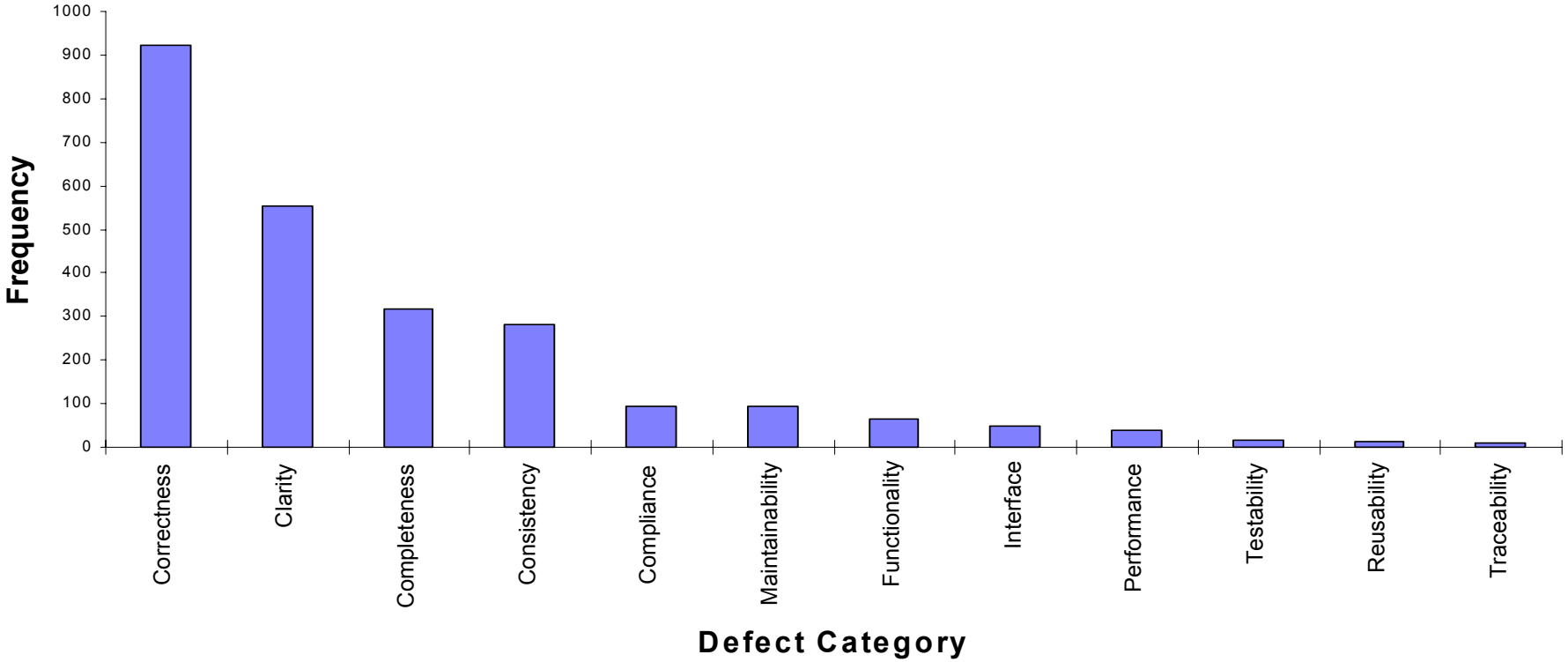
- A defect introduction and removal matrix can be generated and used as a basis for defect analysis and prevention.

	Percentage of Defects				
	Phase injected				
Phase detected	Requirements	Preliminary design	Detailed design	Code/unit test	Total
Requirements	37%				8%
Preliminary design	22%	38%			16%
Detailed design	15%	18%	34%		17%
Code/unit test	7%	24%	28%	43%	25%
Integration testing	7%	9%	14%	29%	14%
System testing	11%	12%	24%	29%	19%
Total	100%	100%	100%	100%	100%

Causal Analysis

- Data on defects is collected and categorized
- Trace each defect to its underlying cause
- Isolate the vital few causes
  - Pareto principle: 80% of defects are traceable to 20% of all possible causes
- Move to correct the problems that caused the defects

Causal Analysis Example



Statistical Process Control

- Look for causes of variation in process data
  - system or chance
    - normally can't change these
    - continuously active in the process
    - are part of the process
  - special or assignable
    - something can be done about it
    - not always active in the process
    - are "extraordinary" events

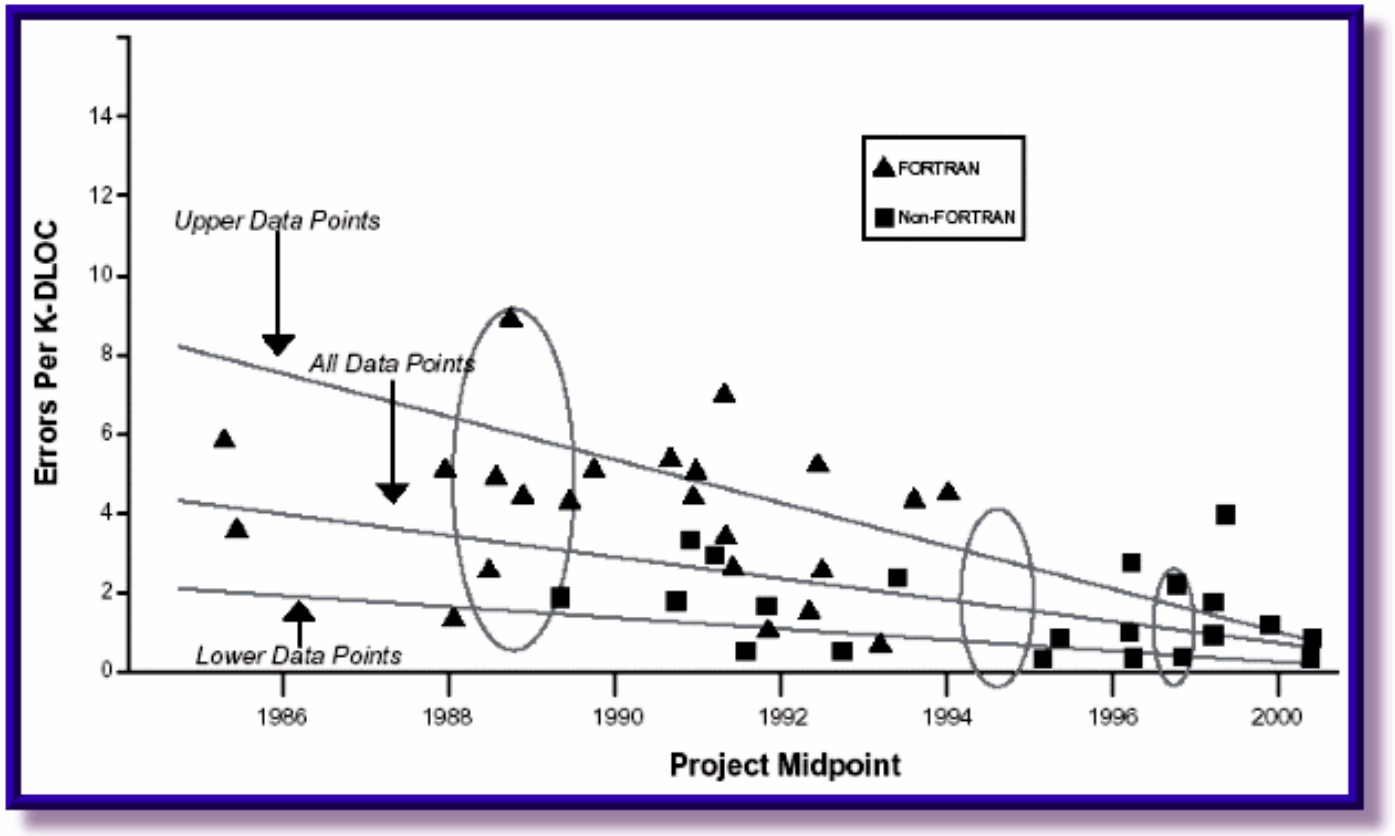
System Causes of Variation

- When variation is due only to system causes, the variations will tend to take on a normal distribution
- The process is by definition running under stable conditions
- If special causes are present, the curve will be distorted or lost its normal distribution
- Need to understand significance of variations in order to control them

Process Capability

- Process capability is concerned with the variation caused by all the sources within the system that can affect:
  - people, methods, machines, material, environment
- A capable process is under statistical control where assignable causes have been eliminated; it is a "stable" process

CeBASE Process Capability Example



Controlled Performance

- Given that the capability for a process is known, then
  - we can measure each new instance of the performed process against the capability definition
  - and when a special cause or a pattern is noted, understand what it means and correct it (if necessary) to bring future performance under control

Statistical Process Control Techniques

- scatter diagrams
- check sheets
- Pareto diagrams
- run charts
- boxplots
- histograms
- control and attribute charts
- cause and effect diagrams
- process capability indices and ratios

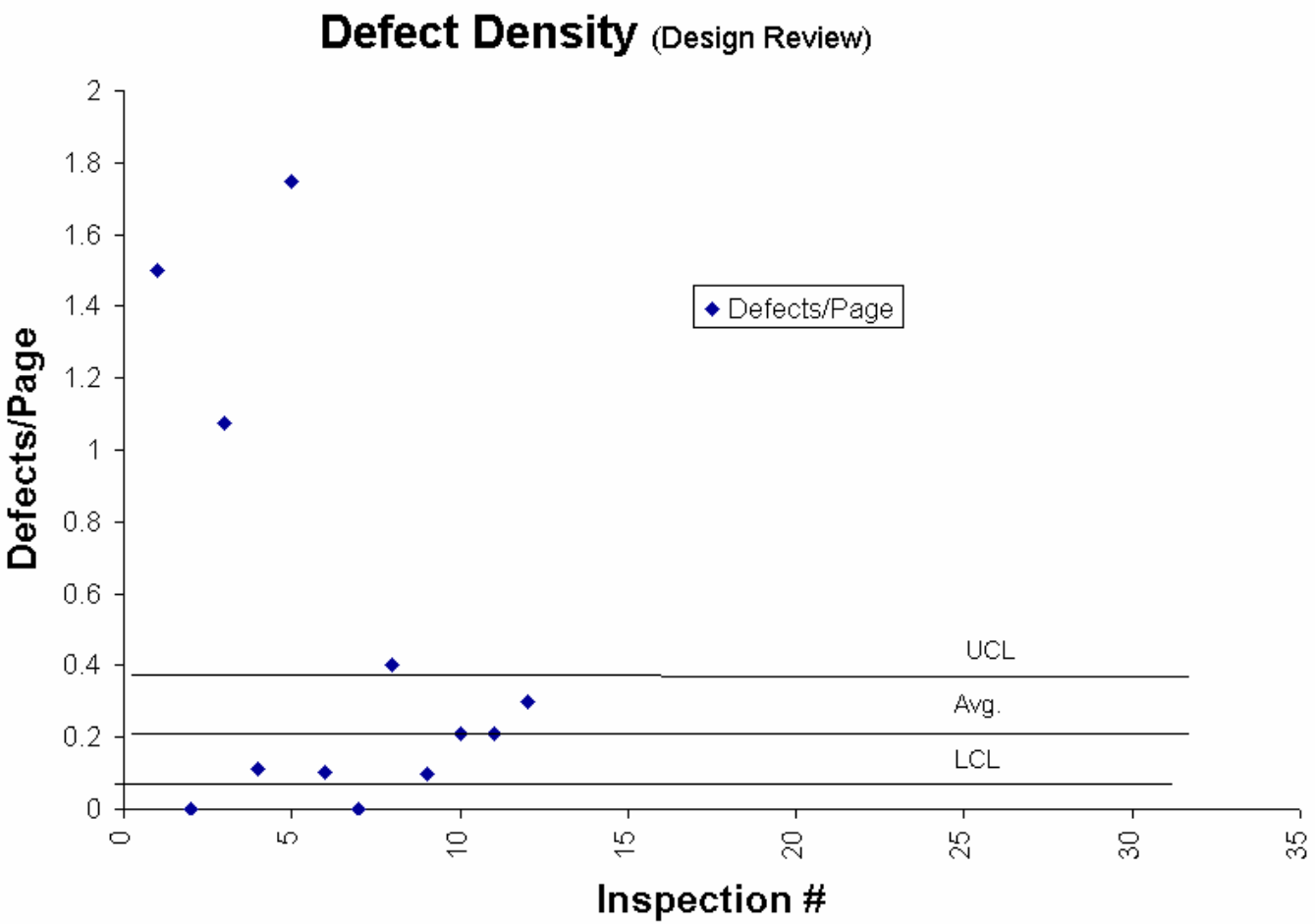
Frequency Histograms

- Take a snapshot of the process in time
- Can help keep track of variations
- Ask:
  - Do measures distribute into a bell curve?
  - What is the average?
  - Do we meet specifications?
- They do not provide information about patterns over time

Control Chart

- Enables visibility into:
  - when the process is running satisfactorily
  - when something is different than expected and may need correction
  - patterns over time
- Measurements to learn capability must be collected over reasonable time period to include variations from all sources, e.g. 30 days
- Capability may change over long-term, so control charts should be revised periodically

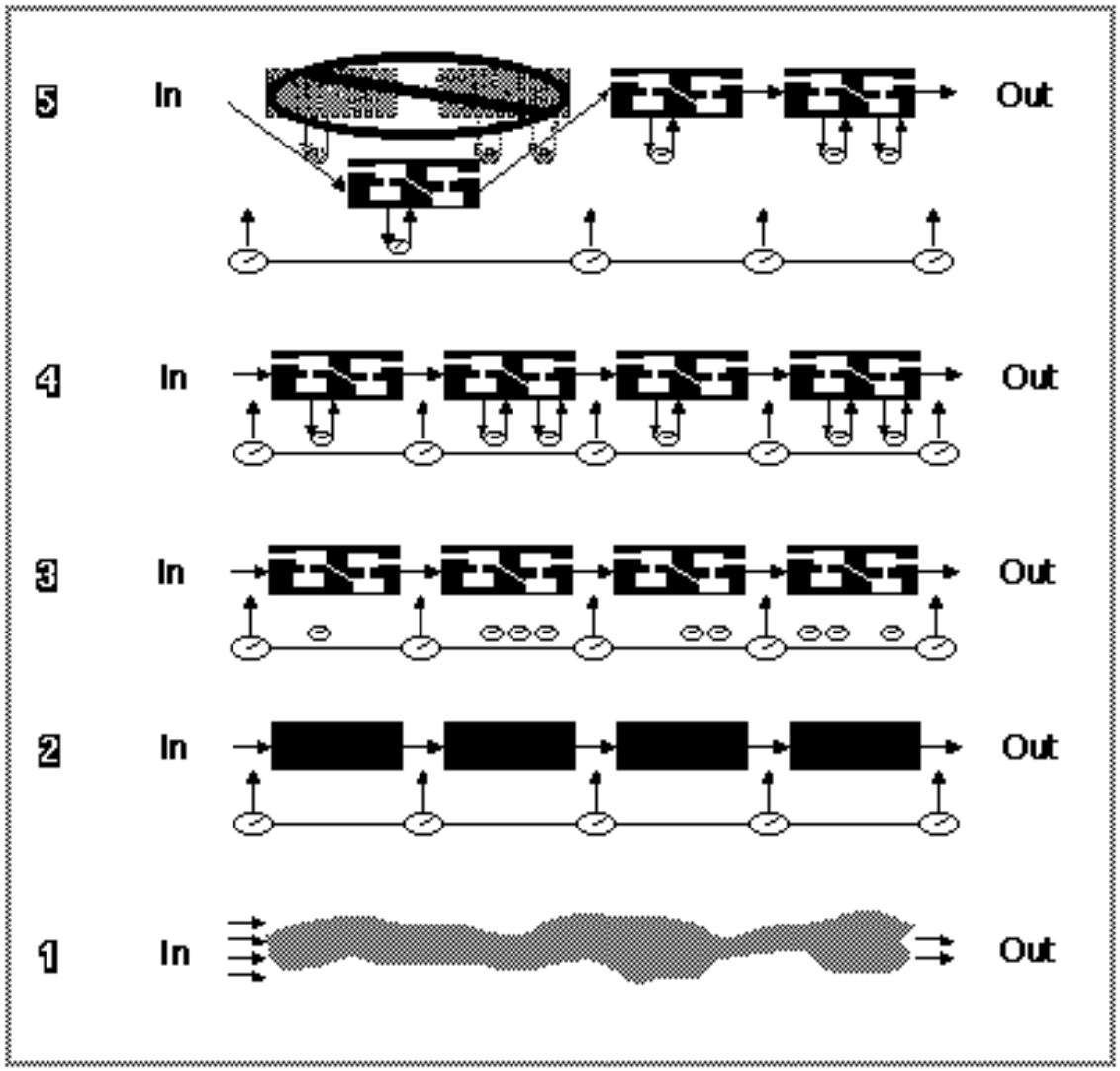
Sample Control Chart



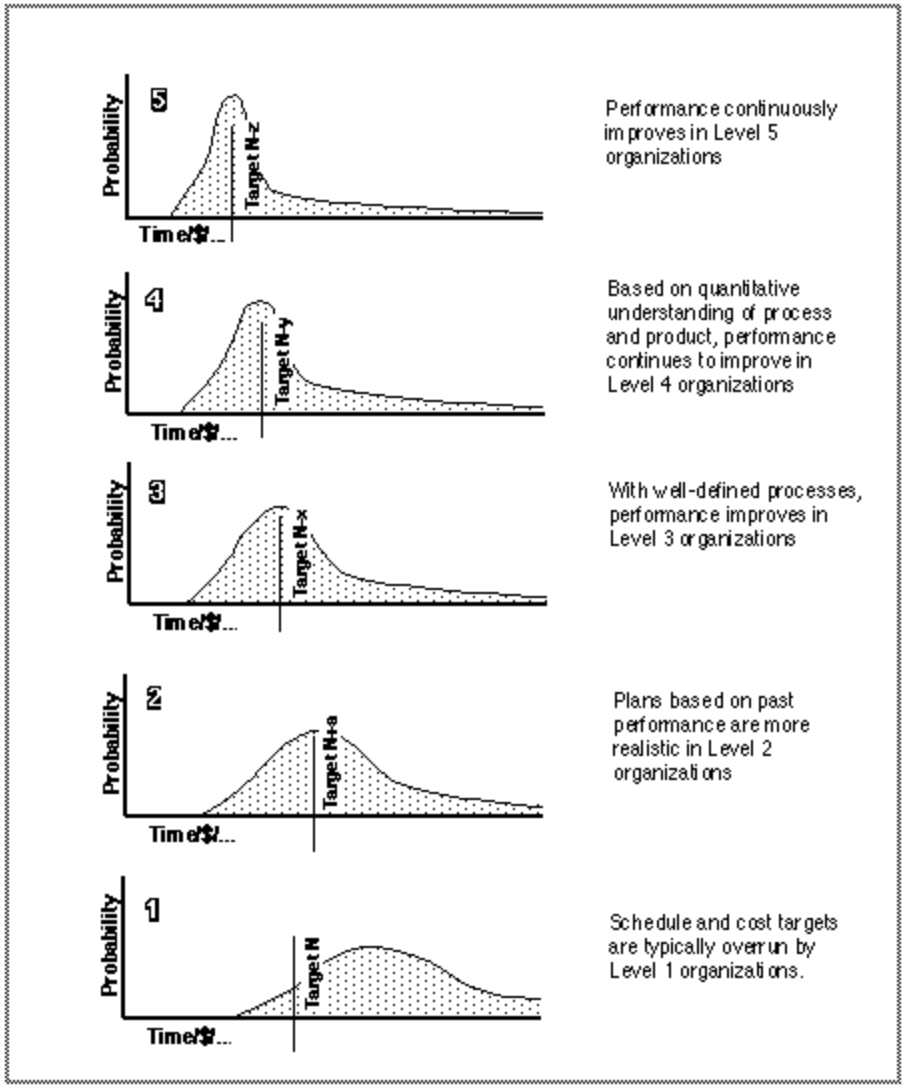
Outline

- **Review of MBASE and CMMI I**
- **Metrics, models and process analysis**
- **CMM Level 4-5 process areas**

Process Visibility per Maturity Level



Process Capability by Maturity Level



CMM Level 4 Introduction

- In CMM Level 4, software process and product quality are quantitatively understood and controlled via collection of detailed measurements.
- In QPM, projects use quantitative techniques to take process measurements, analyze their software process, identify special causes of variations in the performance of the process, control the performance of the process within well-defined limits, and report their results.
- In SQM, projects take measurement and analyze product quality, identify special causes of variations, control the quality within well-defined limits, and report their results.

QPM Goals

- The quantitative process management activities are planned.
- The process performance of the project's defined software process is controlled quantitatively.
- The process capability of the organization's standard software process is known in quantitative terms.

SQM Goals

- The project's software quality management activities are planned.
- Measurable goals for software product quality and their priorities are defined.
- Actual progress toward achieving the quality goals for the software products is quantified and managed.

Sample Metrics

- QPM is controlling the process performance; keeping it stable within acceptable limits
  - productivity
  - quality
  - cycle time
  - defect removal effectiveness
- SQM is identifying product quality goals and quantitatively controlling the quality
  - defect density
  - computer resource utilization
  - product functionality
  - reliability/maintainability
  - all traceable to customer requirements

Quantitative Process Management

The purpose of QPM is to control the process performance of the software project quantitatively. Software process performance represents the actual results achieved following a software process.

ENTRY

1. Policy for measuring and quantitatively controlling performance of project's defined process (C1)
2. Policy for analyzing the process capability of the organization's standard software process (OSSP) (C2)
3. Group exists for coordinating QPM (Ab1)
4. Adequate resources/funding (Ab2)
5. Support for collecting, recording, and analyzing data (Ab3)
6. Training for individuals implementing or supporting QPM activities (Ab4)
7. Training for participants (Ab5)
8. Procedures for Ac1, Ac4, Ac5, & Ac7

TASK

1. Develop Project's QPM plan (Ac1)
2. Perform the QPM plan (Ac2)
3. Determine a strategy for data collection and quantitative analysis based on project's process (Ac3)
4. Collect measurement data to control project's process (Ac4)
5. Analyze the project's process to bring under quantitative control (Ac5)
6. Prepare and distribute reports of QPM results (Ac6)
7. Establish and maintain the process capability baseline for the OSSP (Ac7)

VERIFICATION

1. Reviews with senior management (V1)
2. Reviews with project manager (V2)
3. Reviews/audits by SQA (V3)
4. Measurement of status of QPM activities (M1)

EXIT

1. QPM activities are planned (G1)
2. Project's process performance is controlled quantitatively (G2)
3. Process capability of OSSP is known quantitatively (G3)

Software Quality Management

The purpose of SQM is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals

ENTRY

1. Policy for managing project's defined quality (C1)
2. Adequate resources/funding (Ab1)
3. Training for individuals implementing or supporting SQM activities (Ab2)
4. Training for participants (Ab3)
5. Procedures for Ac1

TASK

1. Develop Project's SQM plan (Ac1)
2. Perform the activities in the SQM plan (Ac2)
3. Project's quality goals are defined, monitored, and revised throughout the life cycle (Ac3)
4. Project's actual quality is measured, analyzed, and compared to the goals on an event driven basis (Ac4)
5. Quality goals are allocated appropriately to sub contractors (Ac5)

VERIFICATION

1. Reviews with senior management (V1)
2. Reviews with project manager (V2)
3. Reviews/audits by SQA (V3)
4. Measurement of status of SQM activities (M1)

EXIT

1. SQM activities are planned (G1)
2. Measurable quality goals and their priorities are defined (G2)
3. Progress towards quality goals is quantified and managed (G3)

QPM Metrics Set

- Projects use standard metric indicators in conjunction with established indicator limits for their quantitative analysis.
- Process goals (productivities, defect densities, review effectiveness) are calculated from combining indicator data.
- Examples:
  - estimated in-process productivity = estimated size * progress in % complete / staffing level
  - defect density = defects found / estimated size
  - inspection effectiveness = inspection defects / inspection effort.

Sample Process Goals

- Numeric values for goals are based on historical data and models
- The following productivity and quality measurements are derived from the progress, effort and defect indicator data
- Goal: control productivity
  - SLOC/person-month in inception
  - SLOC/person-month in elaboration
  - SLOC/person-month in construction
- Goal: control quality
  - Defects/KSLOC introduced in inception
  - Defects/KSLOC introduced in elaboration
  - Defects/KSLOC introduced in construction

Sample Product Quality Goals

- Defect density (defects/KSLOC)
- Critical computer resource utilization (%)
- Additional goals based on customer or organization requirements/priorities
  - functionality
    - response time
    - throughput
    - database size
  - reliability/maintainability (mtbf/mttr)
  - size (KSLOCs)

Level 5 Introduction

- In CMM Level 5, the entire organization is focused on continuous process improvement. The organization has the means to identify weaknesses and strengthen the process proactively, with the goal of preventing the occurrence of defects. Data on the effectiveness of the software process is used to perform **cost benefit analyses** of new technologies and proposed changes to the organization's software process.
- Innovations that exploit the best software engineering practices are identified and transferred throughout the organization.
- Software project teams in Level 5 organizations analyze defects to determine their causes. Software processes are evaluated to prevent known types of defects from recurring, and lessons learned are disseminated to other projects.
- Level 5 organizations are continuously striving to improve the range of their process capability, thereby improving the process performance of their projects.
- Improvement occurs both by incremental advancements in the existing process and by innovations using new technologies and methods.

Level 5 KPAs

- The purpose of **Defect Prevention (DP)** is to identify the causes of defects and prevent them from recurring.
- The purpose of **Technology Change Management (TCM)** is to identify new technologies (i.e., tools, methods, and processes) and track them into the organization in an orderly manner.
- The purpose of **Process Change Management (PCM)** is to continually improve the software processes used in the organization with the intent of improving software quality, increasing productivity, and decreasing cycle time for product development.

Defect Prevention

- The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring.

- Defect Prevention involves analyzing defects that were encountered in the past and taking specific actions to prevent the occurrence of those types of defects in the future. The defects may have been identified on other projects as well as in earlier stages or tasks of the current project. Defect prevention activities are also one mechanism for spreading lessons learned between projects.

- Trends are analyzed to track the types of defects that have been encountered and to identify defects that are likely to recur. Based on an understanding of the project's defined software process and how it is implemented (as described in the Integrated Software Management and Software Product Engineering key process areas), the root causes of the defects and the implications of the defects for future activities are determined.

- Both the project and the organization take specific actions to prevent recurrence of the defects.

Defect Prevention

ENTRY

1. Policy for organization to perform DP activities (C1)
2. Policy for projects to perform DP activities (C2)
3. Organization-level team exists to coordinate DP activities (Ab1)
4. Project level team exists to coordinate DP activities (Ab2)
5. Adequate resources/funding (Ab3)
6. Training for members of the S/W engineering group and related groups (Ab4)
7. Procedures for Ac1, Ac3, Ac6, & Ac7

TASK

1. Develop Project's DP plan (Ac1)
2. Team has kick-off meeting to prepare for DP activities (Ac2)
3. Conduct causal analysis meetings (Ac3)
4. Conduct coordination meetings to review the implementation of action proposals from the causal analysis meetings (Ac4)
5. Document and track DP data (Ac5)
6. Revise the organization's standard process resulting from DP actions (Ac6)
7. Revise the project's defined process resulting from DP actions (Ac7)
8. Provide feedback to developers on the status and results of DP actions (Ac8)

VERIFICATION

1. Reviews with senior management (V1)
2. Reviews with project manager (V2)
3. Reviews/audits by SQA (V3)
4. Measurement of status of DP activities (M1)

EXIT

1. DP activities are planned (G1)
2. Common causes of defects are sought out and identified (G2)
3. Common causes of defects are prioritized and systematically eliminated(G3)

Technology Change Management

- The purpose of Technology Change Management is to identify new technologies (i.e., tools, methods, and processes) and track them into the organization in an orderly manner.

- Technology Change Management involves identifying, selecting, and evaluating new technologies, and incorporating effective technologies into the organization. The objective is to improve software quality, increase productivity, and decrease the cycle time for product development.

- The organization establishes a group (such as a software engineering process group or a technology support group) that works with the software projects to introduce and evaluate new technologies and manage changes to existing technologies. Particular emphasis is placed on technology changes that are likely to improve the capability of the organization's standard software process (as described in the Organization Process Definition key process area).

- By maintaining an awareness of software-related technology innovations and systematically evaluating and experimenting with them, the organization selects appropriate technologies to improve the quality of its software and the productivity of its software activities. Pilot efforts are performed to assess new and unproven technologies before they are incorporated into normal practice. With appropriate sponsorship of the organization's management, the selected technologies are incorporated into the organization's standard software process and current projects, as appropriate.

Technology Change Management

ENTRY

1. Policy for improving the organization's technology capability (C1)
2. Senior management sponsors the TCM activities (C2)
3. Senior management oversees the TCM activities (C3)
4. A group exists who is responsible for TCM activities (Ab1)
5. Adequate resources/funding (Ab2)
6. Support for collecting and analyzing data to evaluate technology change (Ab3)
7. Appropriate data are available to support evaluation and selection of technology change (Ab4)
8. Training for individuals responsible for TCM activities (Ab5)
9. Procedures for Ac1, Ac5, Ac7, & Ac8

TASK

1. Develop organization's TCM plan (Ac1)
2. Identify areas of technology change (Ac2)
3. Inform managers and technical staff of new technologies (Ac3)
4. Systematically analyze and identify areas in the organization's standard process that need or could benefit from technology change (Ac4)
5. Select and acquire new technologies for the organization and the projects (Ac5)
6. Conduct pilot efforts for improving technology before introducing into normal practice (Ac6)
7. Incorporate new technologies into organization's standard process (Ac7)
8. Incorporate new technologies into project's defined processes (Ac8)

VERIFICATION

1. Reviews with senior management (V1)
2. Reviews/audits by SQA (V2)
3. Measurement of status of TCM activities (M1)

EXIT

1. Incorporation of technology changes are planned (G1)
2. New technologies are evaluated to determine their effect on quality and productivity (G2)
3. Appropriate new technologies are transferred into normal practice across the organization (G3)

Process Change Management

ENTRY

1. Policy for implementing software process improvement (C1)
2. Senior management sponsors process improvement activities (C2)
3. Adequate resources/funding (Ab1)
5. Training for managers in software process improvement (Ab2)
6. Training for software engineering staff and related groups in software process improvement (Ab3)
7. Training for senior management in software process improvement (Ab4)
8. Procedures for Ac3, Ac5, & Ac8

TASK

1. Establish a software process improvement (SPI) program (Ac1)
2. Coordinate the SPI activities (e.g., software engineering process group) (Ac2)
3. Develop organization's SPI plan (Ac3)
4. Perform the organization's SPI plan (Ac4)
5. Establish and implement the process for handling SPI proposals (Ac5)
6. Establish teams to develop SPIs in assigned process areas (Ac6)
7. Conduct pilot efforts appropriately before installing into normal practice (Ac7)
8. Implement new processes (Ac8)
9. Maintain records of SPI activities (Ac9)
10. Provide feedback to managers and staff on SPI activities (Ac10)

VERIFICATION

1. Reviews with senior management (V1)
2. Reviews/audits by SQA (V2)
3. Measurement of status of PCM activities (M1)

EXIT

1. Continuous process improvement is planned (G1)
2. Participation in the organization's software process improvement activities is organization wide (G2)
3. The organization's standard process and the projects' defined processes are improved continuously (G3)