

OS/2 Assessment -Phase II

**Edwin R. Lassetre
Darryl Rubin**

**IBM Corporation
Microsoft Corporation**

Mar/90

IBM Confidential

Plaintiff's Exhibit
9020
Comes V. Microsoft

EXHIBIT
712
cm

IBM 04 0000073395

Agenda	1
Objectives	2
Methodology	3
Categorizing Code by Value	4
Software Quality Criteria	5
Architecture quality	6
Design Quality	7
Implementation Quality	8
Review Data	9
Selected Code Review Results	10
Conclusions - Highlights	11
Conclusions - Lowlights	12
IBM vs Microsoft Code	13
IBM vs Microsoft Code	14
Code Redundancy	15
Code Value Classification	16
Effort vs Contribution in "value sector"	17
Reduced Contribution	18
Recommendations	19

◆ Objectives

◆ Methodolgy

◆ Data

◆ Conclusions

◆ Recommendations

- ◆ Review OS/2 at code level
- ◆ Assess IBM vs. Microsoft code quality
- ◆ Compare relative contribution to product value

- ◆ Quality criteria

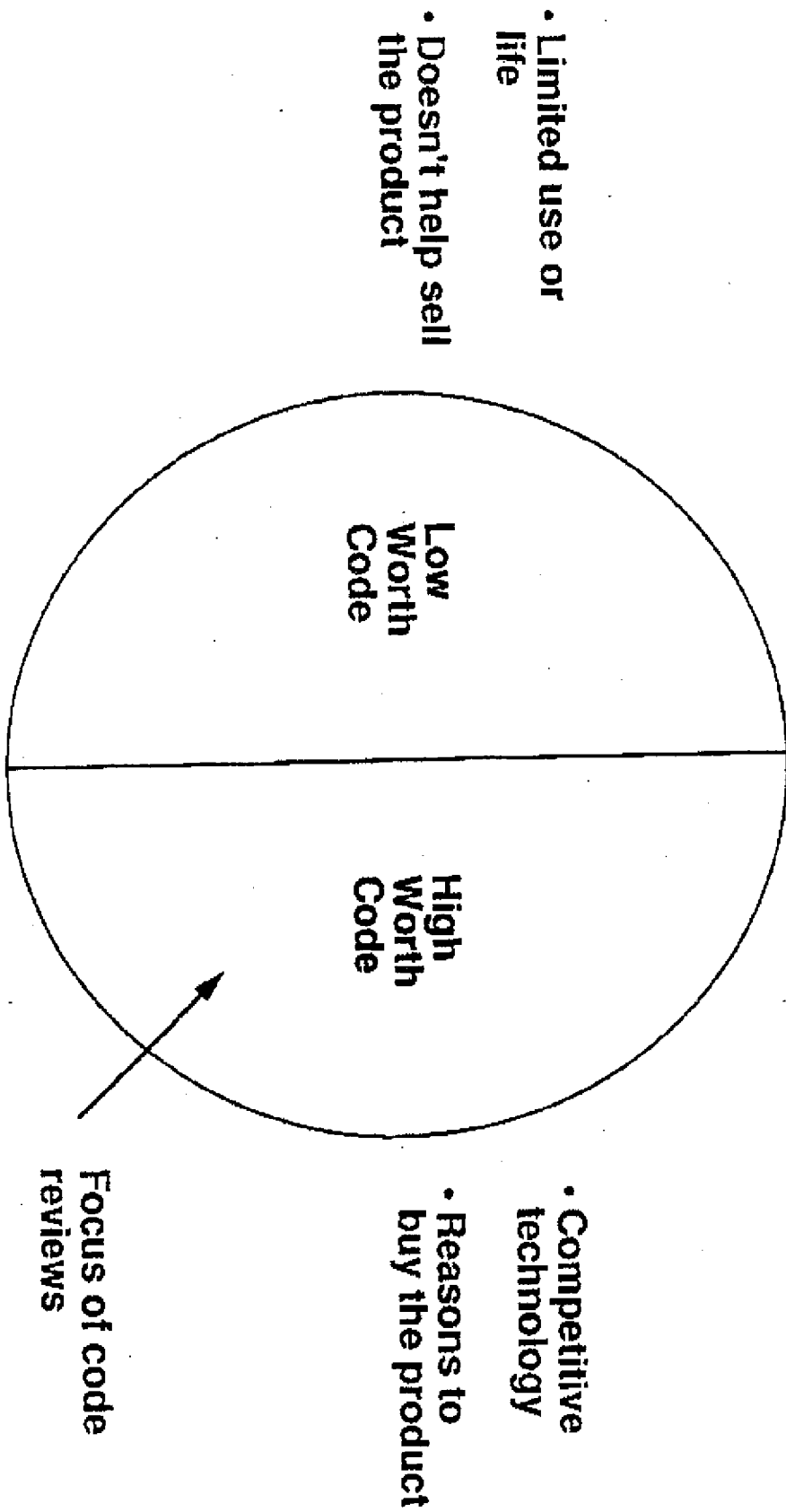
- ◆ Bottom up code review

- ◆ Code sampling

"Important" components (depth)

IDSS taxonomy (breadth)

Categorizing Code by Value



IBM 04 0000073400

◆ Architecture

◆ Design

◆ Implementation

Meet product goals efficiently in current and subsequent releases

- ◆ Choice of technology

- ◆ Trade offs

- ◆ State of the art

- ◆ Extensibility

- ◆ Scalability

◆ **Factorization**

Unique vs common modules

Appropriate layering / primitives

◆ **Interface design**

Performance

Appropriate abstraction / hiding

◆ **Shared data structures / algorithms**

Speed vs space

Limits / scalability

- ◆ Logical organization

 - State flags / special cases

 - Sequencing

 - Procedures

- ◆ Error handling

- ◆ Memory allocation strategy

- ◆ Data driven versus hard coded logic

- ◆ Synchronization/use of global variables

- ◆ Efficiency

 - Amount of code for function

- ◆ Modules in product: 4000+
- ◆ Modules reviewed: 125+

Selected Code Review Results

OS/2

Code Reviewed	Efficiency	Logic	Algorithms/ Data Structures
Microsoft Code			
Virtual Memory	OK	OK	OK
Semaphores	OK	OK	OK
IFS	OK	OK	OK
GRE BitBit	Great	OK+	Great
GRE Drawline	OK	OK+	OK
Dialog Editor	OK	OK+	OK
Epson driver	OK	OK	Slick
HPFS	OK	OK	Great
IBM Code			
Physical Allocation	Fat Slow	Tangled	Inappropriate
Scheduler	Slow	OK	Inappropriate
PMFILE	Fat Slow	Monolithic	Inappropriate
Control Panel	Fat	Disorganized	OK
Help Manager	Fat	Disorganized	Inappropriate
Spooler	Fat Slow	Disorganized	Inappropriate
4019 Driver	Fat Slow	Hard to follow	Inappropriate
SHAPI	Fat Slow	OK	Inappropriate
Swapper	Fat	OK	Inappropriate
GPI	Fat Slow	OK	Fumbled abstractions

- ◆ Built-in graphics

though problems with follow through

- ◆ Thread/Process model

- ◆ HPFS

- ◆ Exposure of graphics engine for use by device drivers

- ◆ IFS architecture

◆ **Lack of needed architecture**

Redundant services

Poor integration - the seams show

◆ **Low technology in key areas**

◆ **Fat code**

◆ **Lack of extensibility**

◆ **Many low value components**

- ◆ More LOCs per function (up to 2-3X)

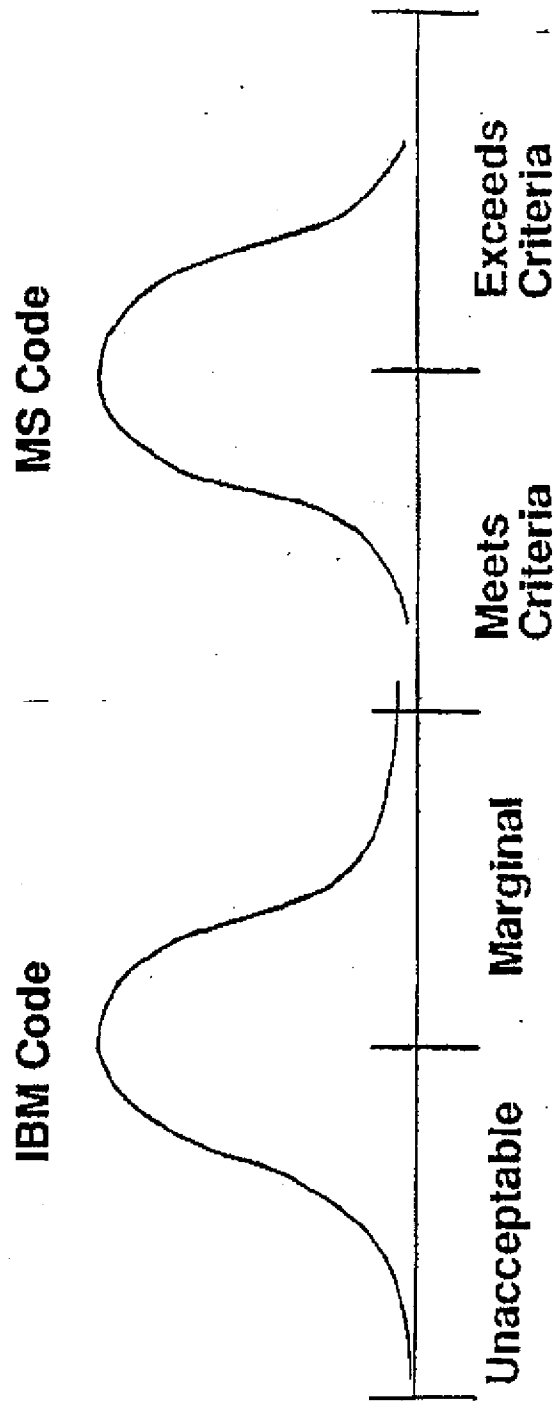
- ◆ Less direct/poorer structure

- ◆ More redundancy

- ◆ Less efficient in time critical areas

- ◆ Harder to read/maintain

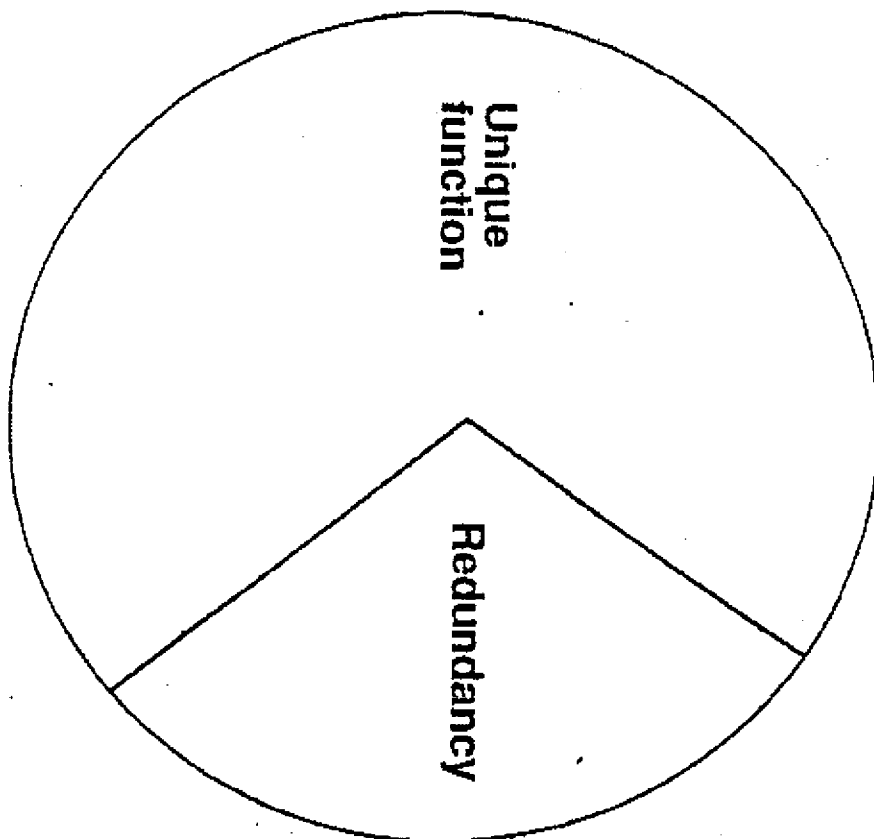
IBM vs. MS Code



IBM 75 397

IBM 04 0000073410

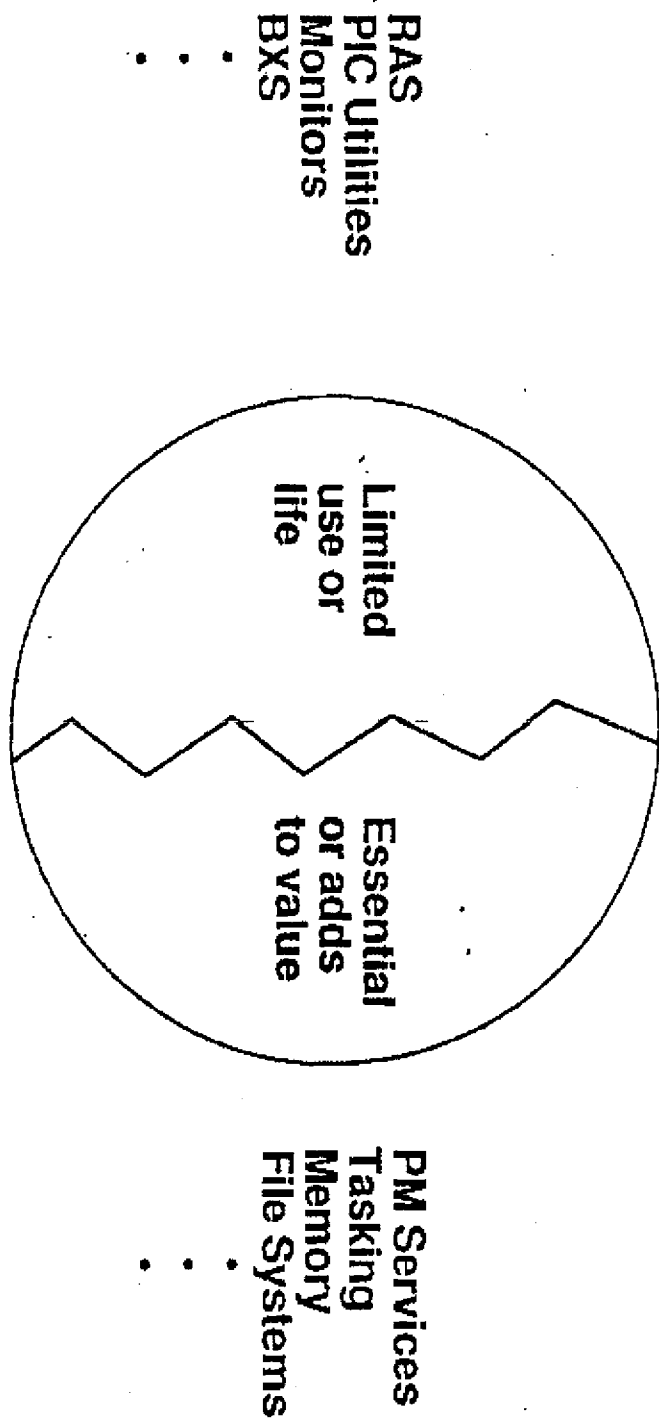
Code Redundancy



- PM Drivers (blitter, ...)
- PM UI (tidy, ...)
- Thunk
- STR*
-
-
-

IBM 04 0000073411

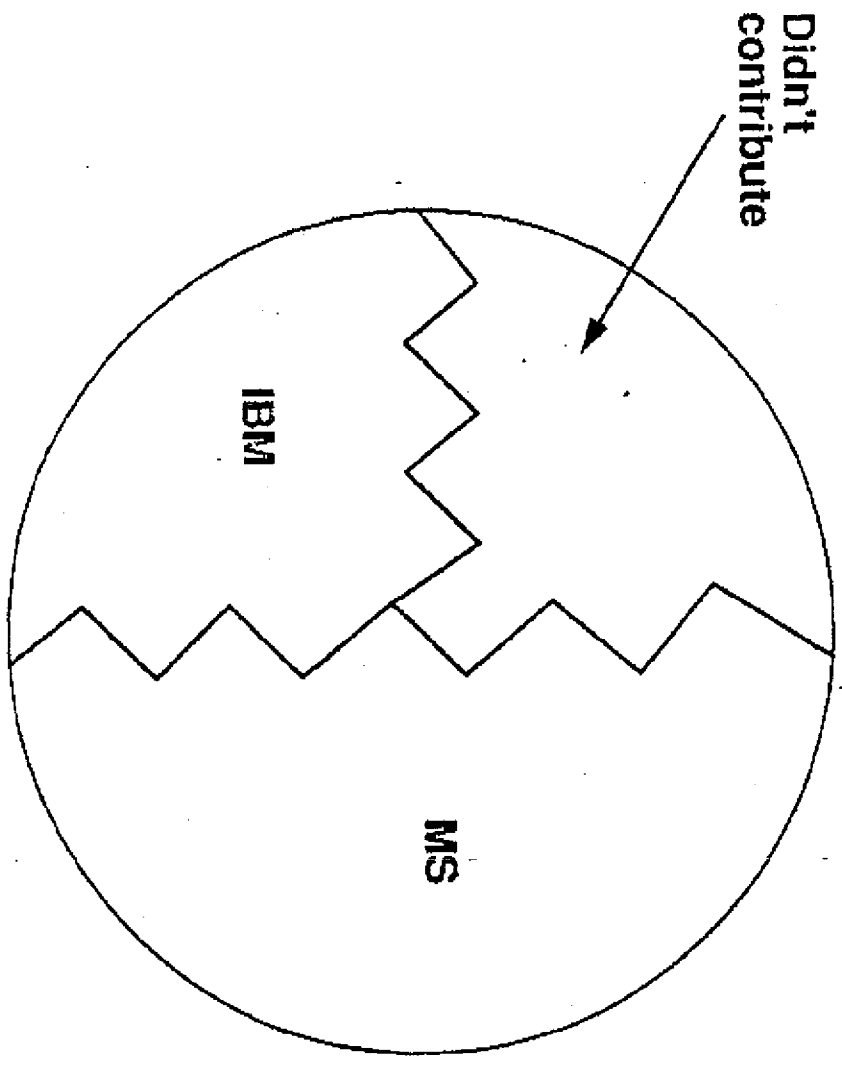
Code Value Classification



Half or less than half is of value

IBM 04 0000073412

Effort vs. Contribution in "Value" Sector



IBM 04 0000073413

◆ Fat code

(1:6 < Fat:Meat < 4:1)

◆ Replacement needed

(current contribution close to zero)

◆ Rework needed

(partial contribution)

Unable to correlate LOC and contribution

- ◆ System design effort to spring clean the system

- ◆ Renewed focus on code workmanship and attention to detail

- ◆ One IBM development site

- ◆ Streamlined MS/IBM coordination