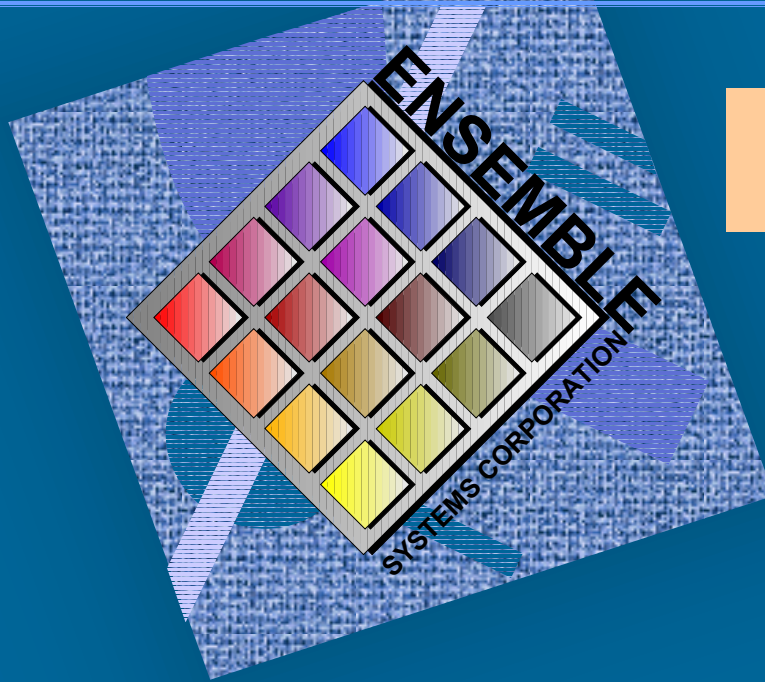


Legacy Systems in the new Economy



Overview

- The problem
- Legacy Characteristics
- How did it happen?
- eBusiness Microscope & Y2K
- Legacy Valuation



The Problem

- New economy new kids on the block.
- Bricks and mortar companies have awakened
- This is not a small problem. According to the Gartner Group:
 - there are some 10,000 mainframe computers worldwide housing some 200 Billion lines of COBOL code
 - It cost \$6 to \$23 per line of COBOL to manually rewrite a system, and
 - 85% of existing applications will migrate to client/server architecture.

Use the Legacy

- B&M companies have a large portion of their net worth in software systems that represent their business.
- They would like to use this with minimum cost.
 - eBusiness thrust
 - Addition of CRM, ERP, etc.
 - Mergers and Acquisition
 - 1st generation web mess

Legacy Characteristics

- 15 to 30 years old
- Outdated data storage technology that may be proprietary.
- Lots of COBOL
- Architect-ed for a different purpose!
- Monolithic — Business Logic/Data Design/Presentation Design integrated into a seamless pile
- Obscure and complex logic—the dread spaghetti code
- Excellent documentation—yeah right
- And the wonderful—incorrect code comment
- Changes in business logic has to be performed at the source code level
- There's been several generations of staff
- Original staff long since gone—many are even dead
- Staff afraid to change anything
 - The poke it and stand back method
 - Small change huge regression test
- Staff asking the boss and he don't know either

How did we get here?

- The code started out with problems
 - Before Modern software design
 - Data Modeling
 - Structured Design
 - Poor or no change control
 - Poor testing—testability not designed into the code



How did we get here?

- The second reason: Software Entropy
 - Measure of disorder \equiv lack of information \equiv lack of system knowledge
 - Measurement of the deterioration of a system
 - 2nd law of thermodynamics says the only way to stop the deterioration is through work.
 - The causes
 - Maintenance and enhancement increases entropy
 - Poor management attitude: Software is a cost not an investment—under funded
 - Change for change sake
 - Just fix it, next quarter must look good
 - Software makes it easy to create complexity
 - Loss of staff increases entropy
 - Lay off the “50 percenter”
 - Retention of legacies in the system?
 - We’ll just keep it for a while—yeah right!
 - Increased Entropy is bad, but it’s inevitable



eBusiness Microscope

- Shows the architectural problems clearly.
- Shows the entropy problems clearly.
- Customers reaching into your information
- Security problems
- Externalization of Functional Definition



Y2K-Remember Y2K

- Strengthened and lengthen the life of many legacy systems
 - Lots of code was cleaned up
 - Lots was learned about existing systems
 - Education of management of IT assets
 - Wow are we this dependent on our s/w systems?
- Uncovered lots of problems
 - Poor project management
 - Poor testing
 - Poor configuration management
- Lots of tools were developed
 - System Evaluation
 - Rule mining.
 - Database logic
 - Business logic
 - Presentation logic
 - SEEC, Relativity -- half the costs?



Legacy Valuation

- How do we value legacy?
- Required to make a rational build/buy/convert decision