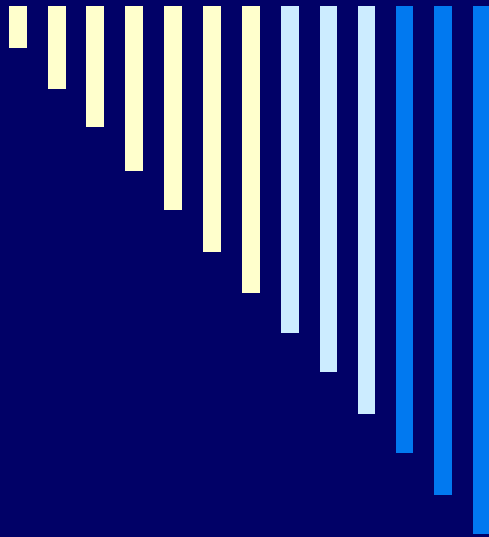


Application Software Maintenance- A Queuing Theory approach for continuous improvement

Presented by
Dr.P.Balasubramanian,
C.E.O.,Theme Work Analytics

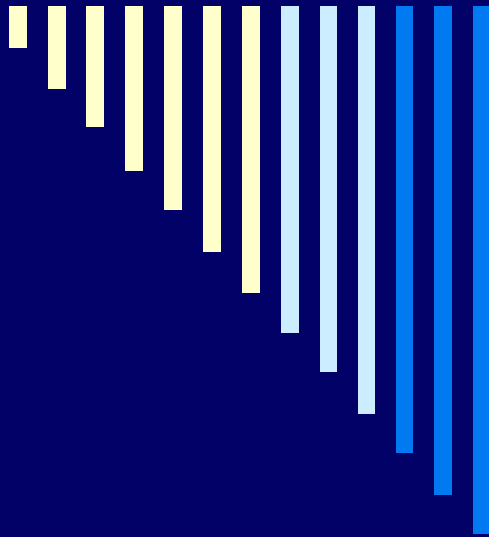
Feb 3,2006

Application Software Maintenance



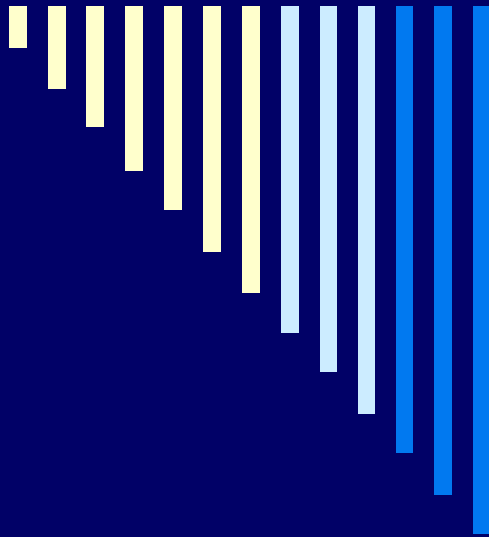
- 70 % of the I.T.software budget in any large company is spent on Application Maintenance
 - They are ERP, Financial Accounting, Costing, SCM ,CRM , Compliance Reporting or Share holder servicing systems .
 - They could be either custom developed , bought out packages or hybrid
 - Even when packages are implemented, they tend to be customized over a period of time
 - An average system will have close to a million lines of COBOL code or equivalent
 - These systems need to be maintained on an ongoing basis.
-

Application Software Maintenance



- Maintenance activity can be tedious since the system documentation lags its development ; does not contribute to technology skills augmentation for staff
 - Such applications also carry the business risk of interruption when key staff leave the company
 - Yet they are mission critical systems. They need to be maintained.
 - Firms in India, Ireland and Philippines have found a service market here and have grown rapidly
-

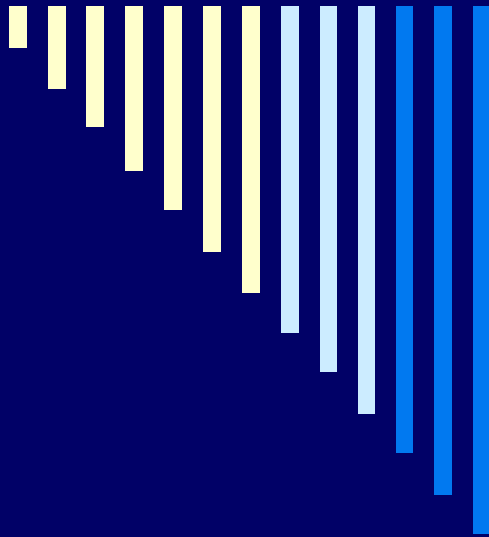
Application Software Maintenance



The Maintenance need arises due to

- Bugs undiscovered in initial implementation
 - Lax data verification procedures and subsystems
 - Changed data conditions
 - Changing business needs
 - Changes in environment; legislation etc
 - Bugs injected during the Maintenance process
-

Application Software Maintenance

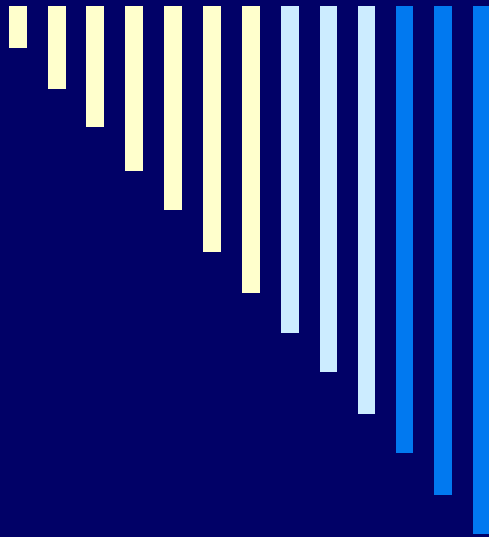


The Maintenance effort required is a function of

- Bugs density
- Total lines of code
- Frequency of use of the system etc
- Required service level

Hence a critical challenge facing firms is to manage the code integrity and the resources expended to maintain the system. This has to be done in the context of the task being mundane with low levels of worker motivation.

Application Software Maintenance

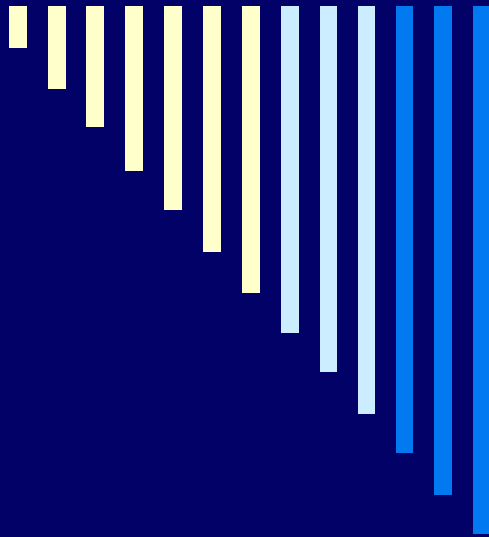


The Maintenance initiatives are categorized as

1. Minor bugs
2. Major bugs (minor or major determined by function criticality and effort required to rectify)
3. Enhancement initiatives (minor or major)

The maintenance team size is usually constant. Hence the challenge facing the team manager is to decide on allocation of resources to different maintenance tasks. When an enhancement initiative is time critical and requires large effort level , the work is either outsourced or team size is increased. Most of the time the Maintenance team works with a back log.

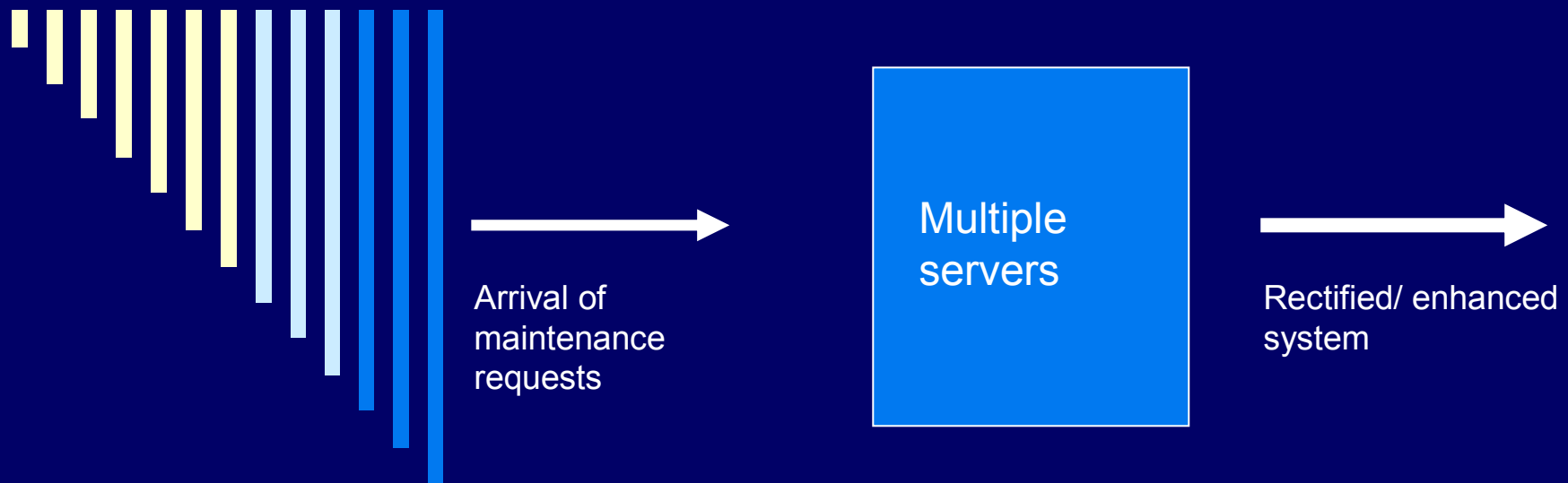
Application Software Maintenance



In 1995 we were faced with the task of submitting a competitive proposal to Nynex corporation to take over the maintenance of number of key applications.

Queuing Theory based approach was evolved at that time.

Application Software Maintenance

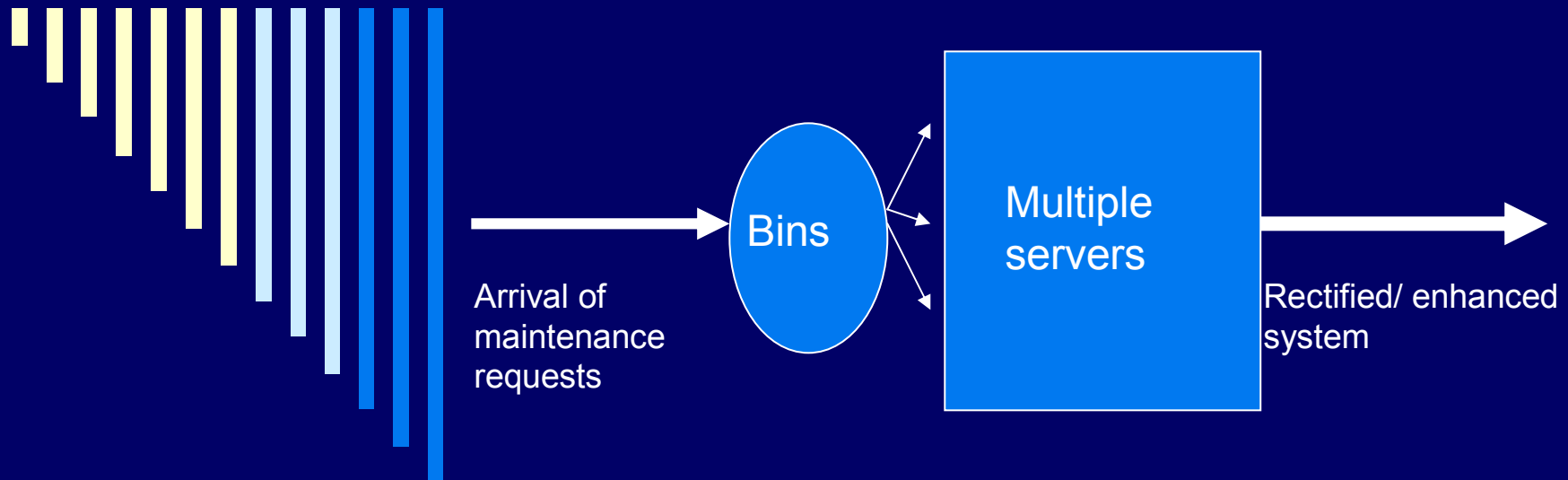


Ideally, the service capacity should be higher than the service demand to avoid build up of queues and extensive wait time in queues.

It should facilitate first in first out principle.

In reality they are not assured.

Application Software Maintenance



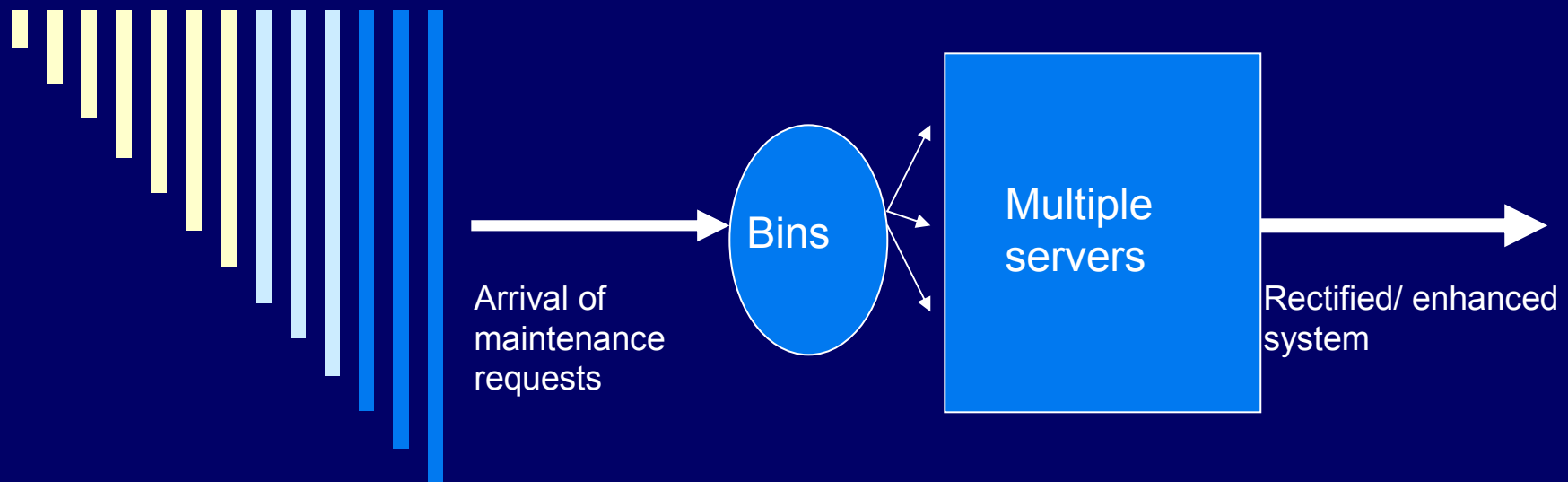
Servers tend to specialize in parts of the application.

Arrival rate at times can be excessive. Demand > capacity

Randomness of arrivals leads to queue build up

Item rectification priority changes dynamically.

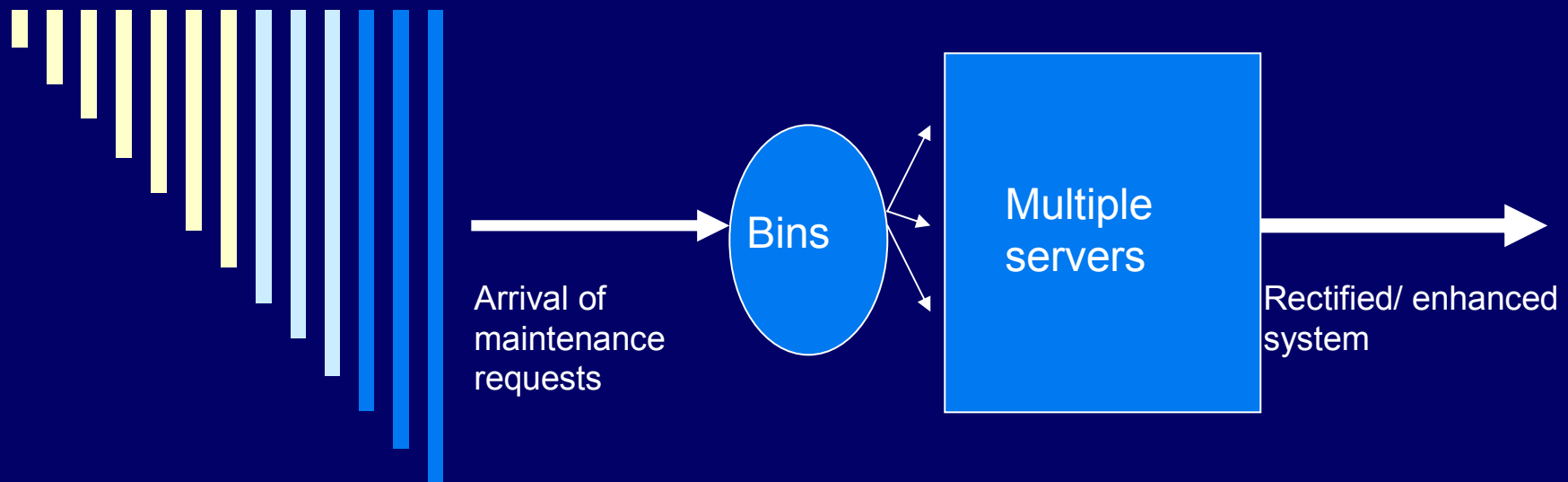
Application Software Maintenance



We determined that key to our success in winning the bids is to improve server productivity.

The productivity improvements should result in decrease in number of servers needed from time to time.

Application Software Maintenance

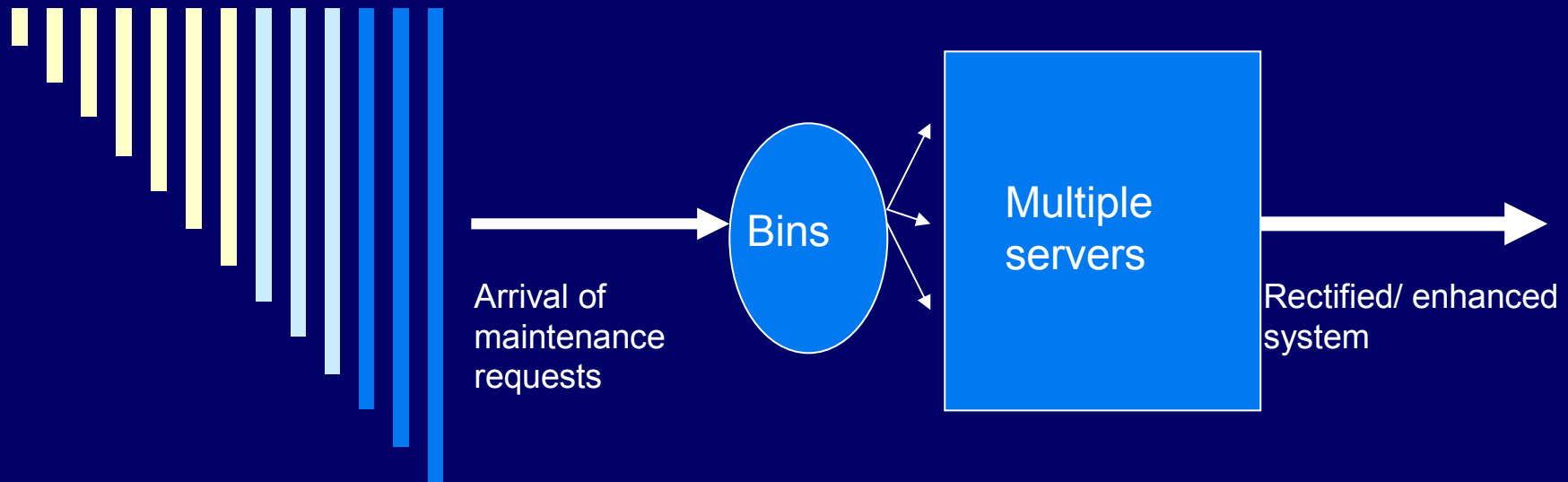


We separated the efforts needed for bug fixing from enhancements.

We could reduce the team size from quarter to quarter if

- (a) The customer agreed to freeze the enhancement requests for a period of time or to treat them as a separate project
 - (b) We can improve server productivity continuously
-

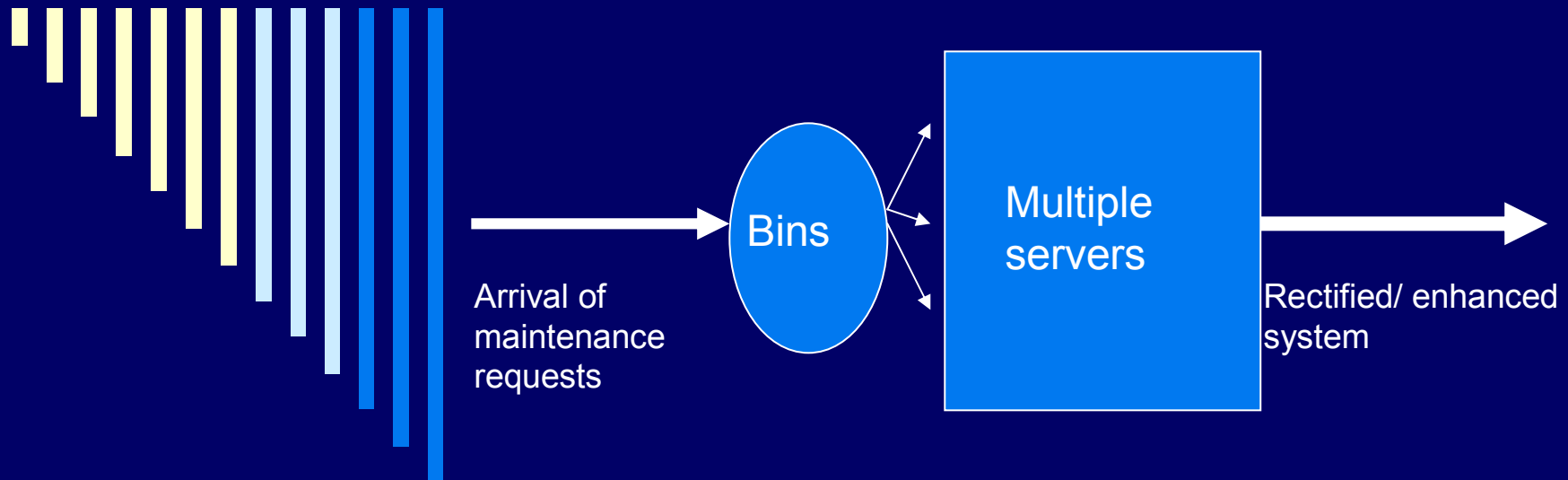
Application Software Maintenance



Server productivity can be improved by

- Building multiple skills in each server
 - Bringing the documentation upto-date
 - Creating a shared knowledge data base for system based learning
 - Creating standard procedures for error analysis, impact analysis, rectification and testing
-

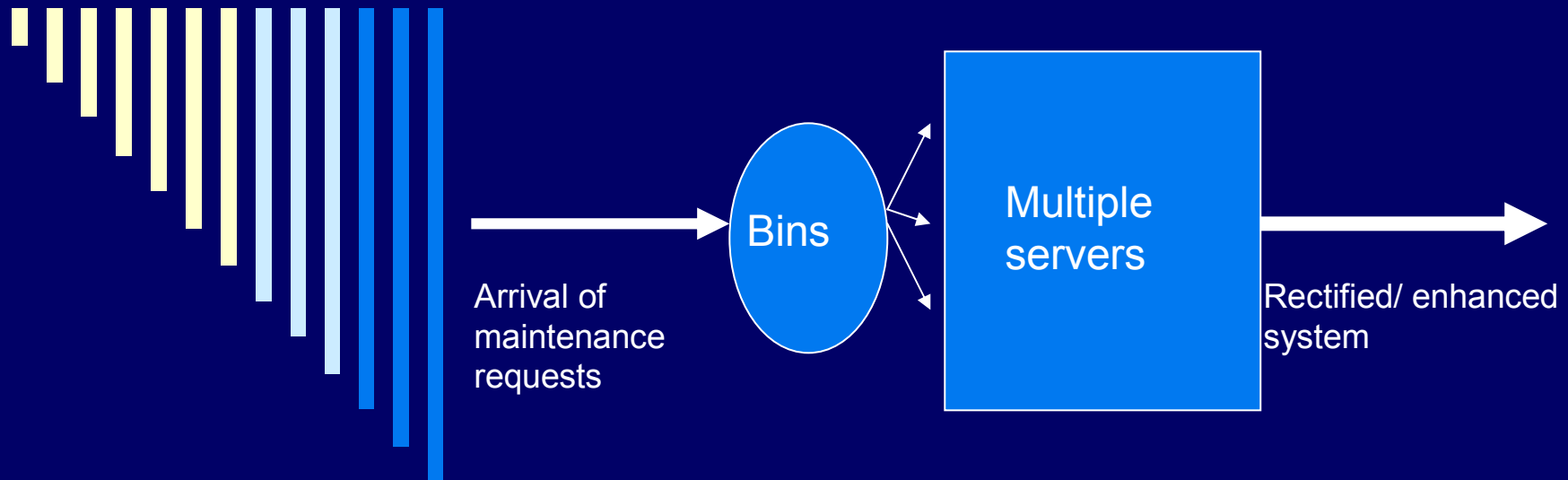
Application Software Maintenance



Server productivity can be improved by

- Use of domain experts as trainers and project resources
 - Extensive use of metrics to measure and report productivity levels and service levels
 - Restructuring the code periodically to improve its maintainability
-

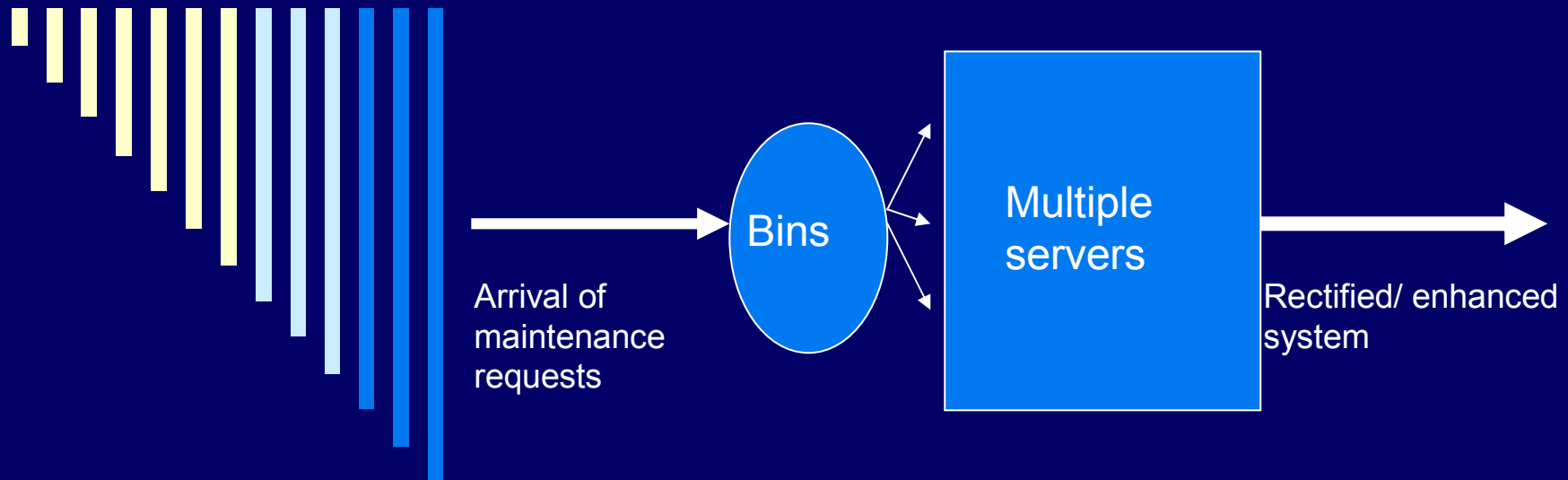
Application Software Maintenance



When this approach was articulated in our proposal and backed by commitment to reduce resources needed for maintenance from quarter to quarter

- Nynex was impressed sufficiently to hand over 3 major systems for a period of three years.
 - Successful execution of these projects led to our winning many assignments from other customers with similar needs. The project data were freely shared with customers .
-

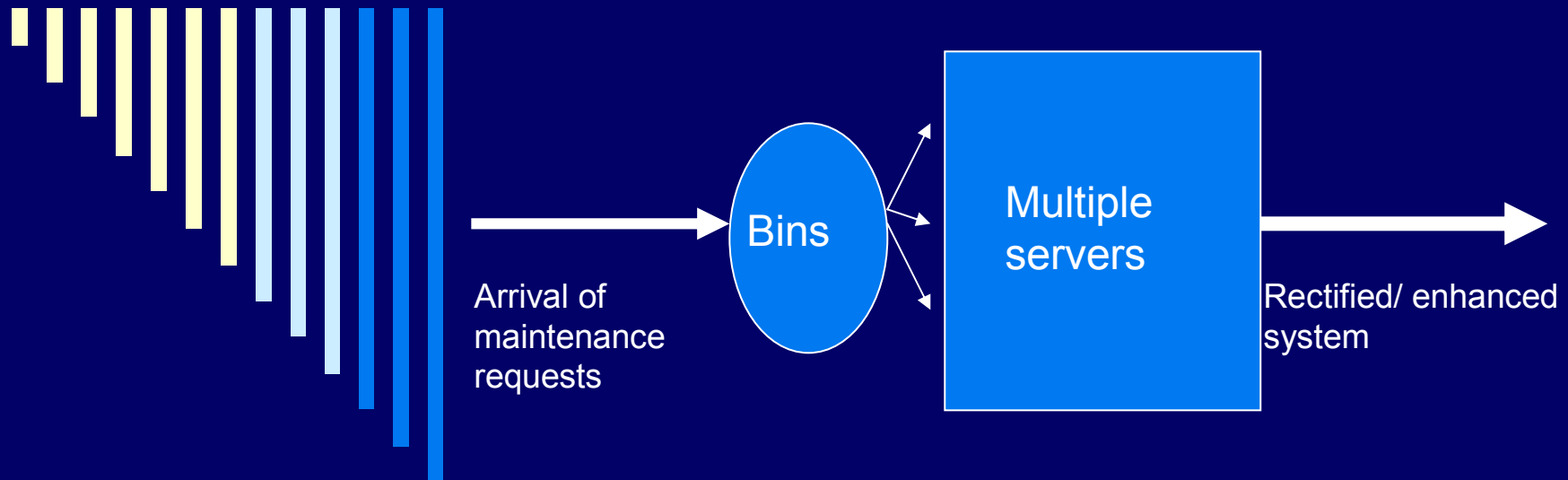
Application Software Maintenance



Post Y2K the pressure on use of IT dollars for development projects was high. Customers became very demanding in aggressive reduction of resources used for maintenance and redeployment in development projects .

This challenge was converted into an opportunity by the following approach.

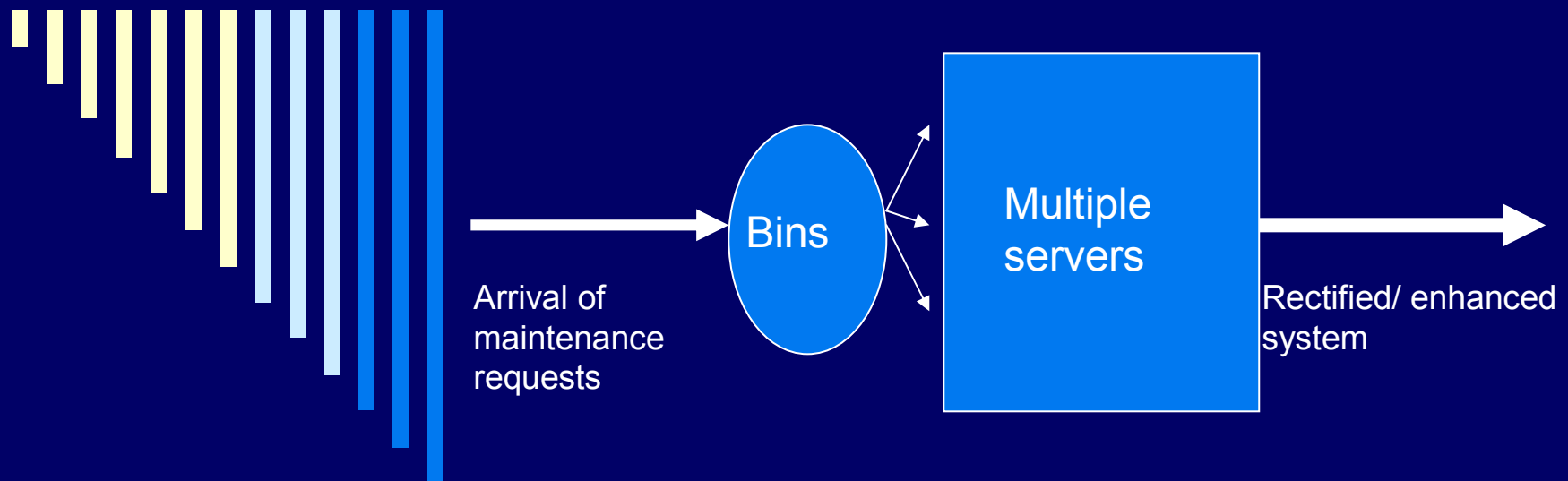
Application Software Maintenance



We sensed that a dramatic solution would be to keep reducing the arrival rate continuously.

A causal analysis of the arrival rate was carried out of previously executed projects. It revealed that (a) slow surfacing of hidden bugs over a period of time and (b) bugs injected during the maintenance process were the prime causes.

Application Software Maintenance

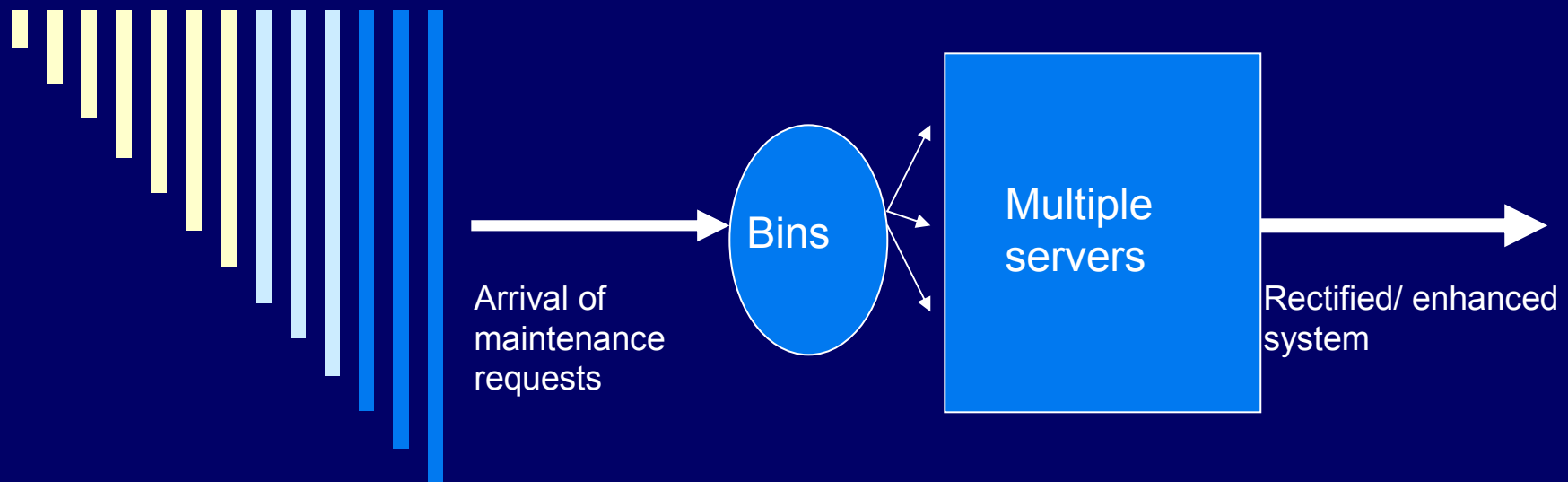


It called for a radically different approach to testing the software.

Till then testing was confined to

- Successful removal of the reported bug
-

Application Software Maintenance

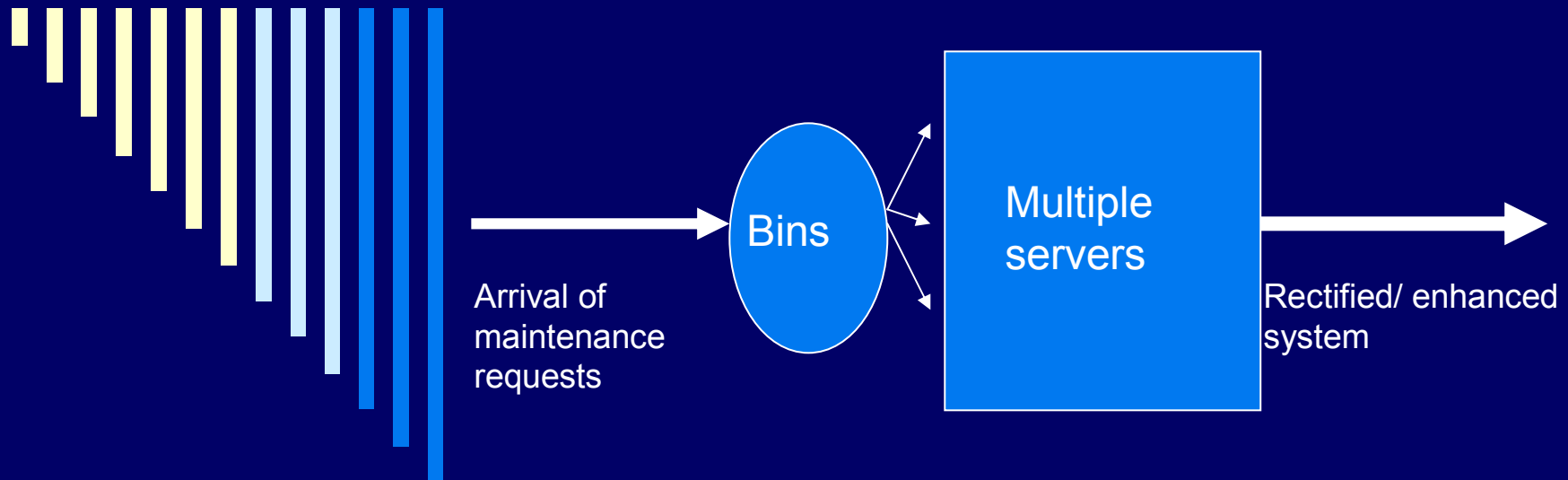


It called for a radically different approach to testing the software.

In future , we proposed

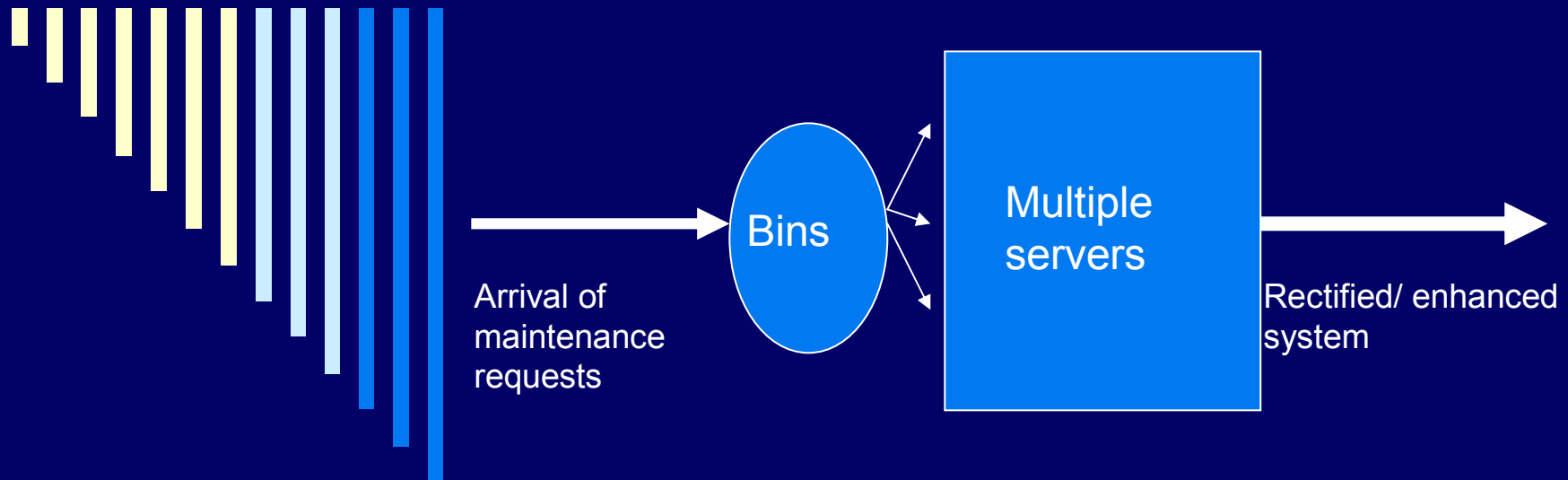
- Regression testing to ensure that no other part of the system was affected due to this change
- Comprehensive testing of the system in the first quarter (as though customer was contractually obligated to deliver an error free system to us)

Application Software Maintenance



Further the data edit programs and modules were to be reviewed and tightened to avoid incorrect data getting through far into the system and then causing system halts.

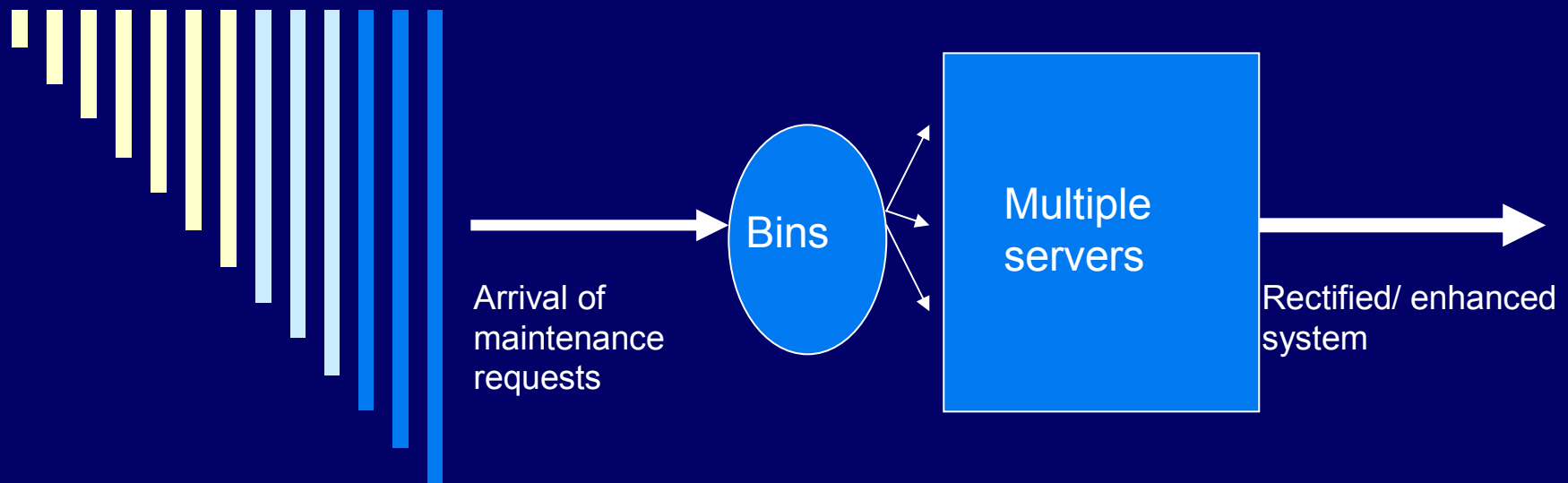
Application Software Maintenance



These led to early capture of hidden bugs and minimizing of freshly injected bugs. The benefits realized were substantial. Maintenance team size , in some instances went down from 12 to 2 over a two year period.

The happy customer was more than willing to give the development assignment to the team members released from maintenance duty.

Application Software Maintenance



Thank You.

balasubp@gmail.com
