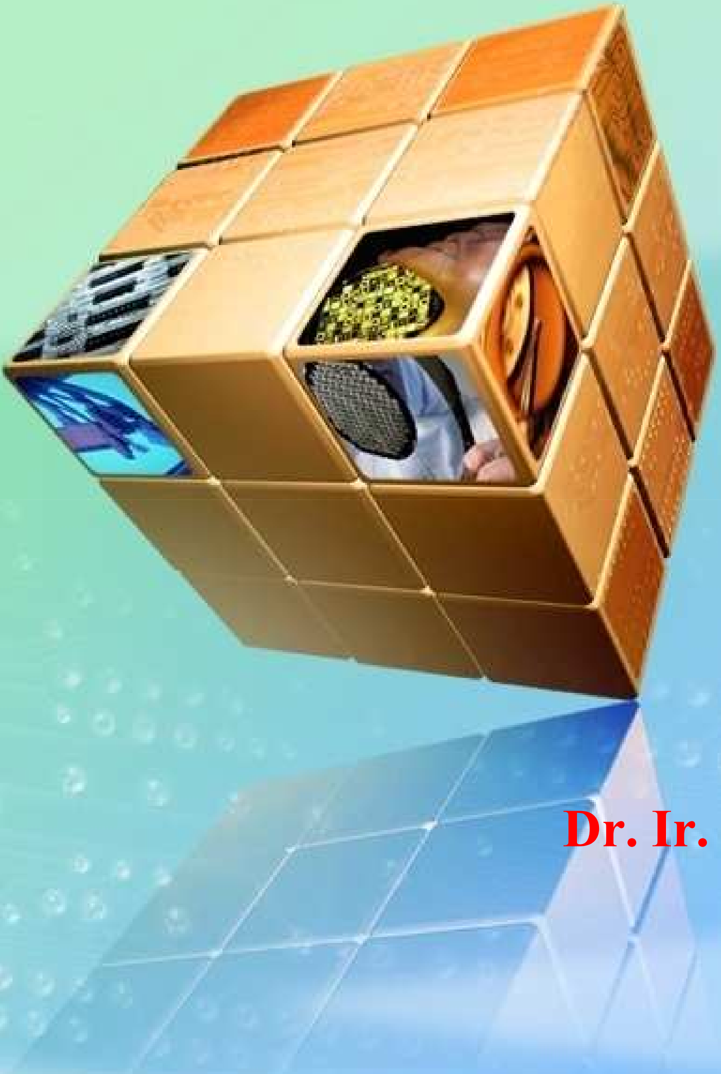
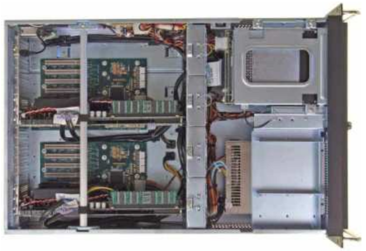


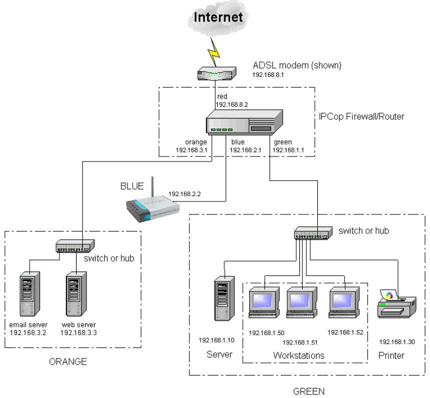
Bab I: Kinerja dan Evaluasi

Dr. Ir. Yeffry Handoko Putra, M.T

Contoh-contoh Sistem yang dianalisa dalam Teknik komputer



Sistem komputer



Sistem Jaringan

Sistem Pengukuran dan Monitoring

Sistem Pengontrolan Alat dengan komputer

Bagaimana belajar Analisa Kinerja Sistem

- Pelajari dasar matematika statistik
- Baca artikel secara sistematis
- Peroleh pengalaman dengan bereksperimen

Terminologi Kinerja

- **Metriks** :
  variabel ukur kinerja (metric)

- **Tujuan Kinerja (Goal)** :
  untuk mencapai suatu fungsi tujuan yang dipersyaratkan.(maksimal, minimal atau optimal)

  Fungsi tujuan bisa berupa fungsi biaya (cost function), fungsi kehandalan (reliabilitas function).

- **Kriteria** :
  Output hasil pencapaian /pemenuhan dari usaha pengelolaan kinerja

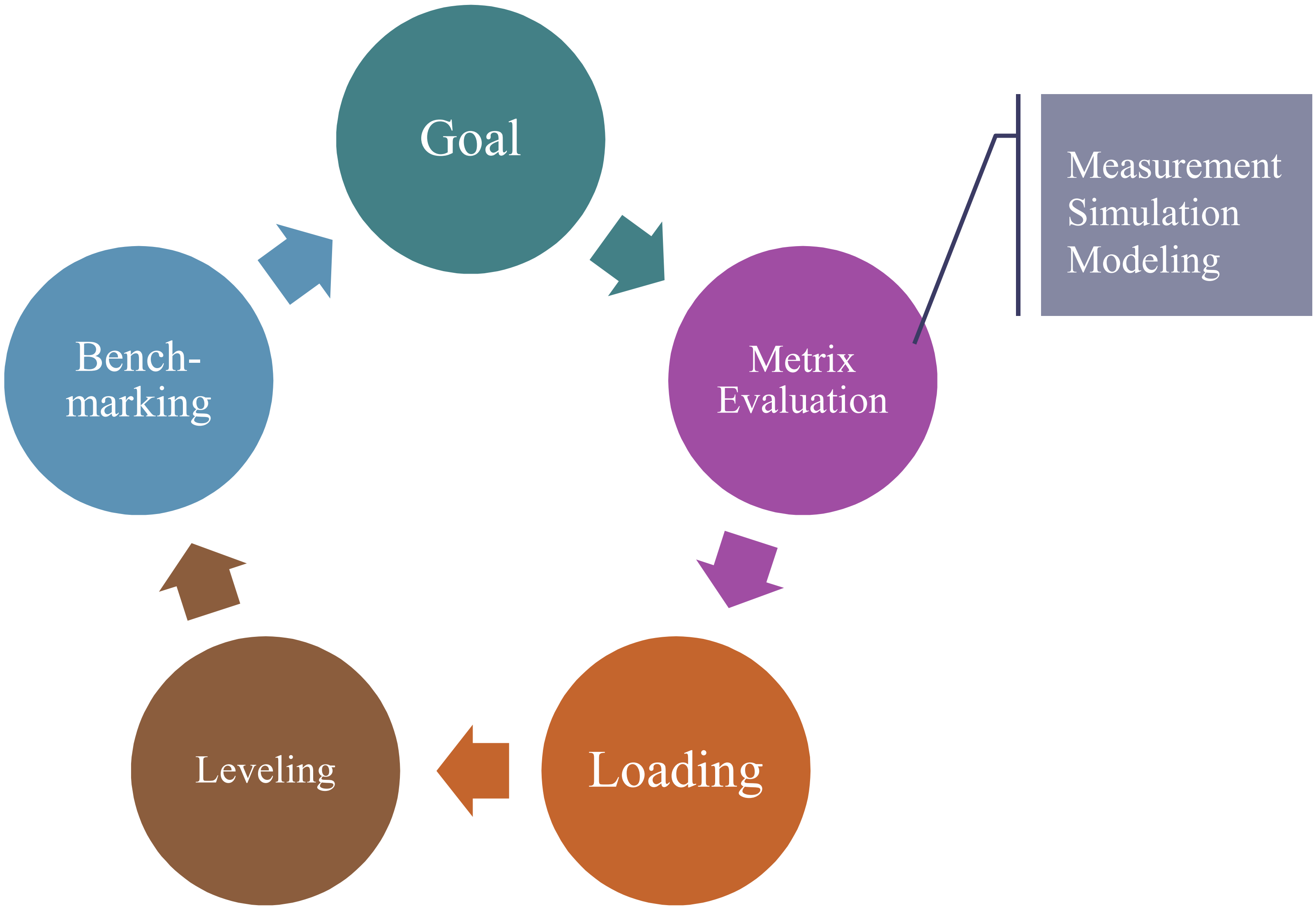
- **Spesifikasi/Requirement**:
  output ideal yang diinginkan (demanding) dari usaha pengeloaan kinerja

Terminologi Kinerja (2)

- **Beban/Load** :
  Variabel yang mempengaruhi kinerja

- **Kapasitas**
  Kemampuan sistem mengelola beban per satuan waktu. Misal keluaran prosesor dinyatakan dengan throughput / s

Tahapan Kinerja



Masalah dalam analisa kinerja sistem [Jain]

- *Specifying Performance Requirement*
- *Evaluating design alternative*
- *Comparing two or more system*
- *Determining optimal value of parameter (tunning system)*
- *Bottleneck Identification*
- *Workload Characterization*
- *Capacity Planning*
- *Forecasting performance*

Metriks berdasarkan

Waktu

- lifetime, MTBF, MTTF, MTTR
- Settling time, rise time, peak time, transmission time

Jumlah

Availabilitas, Capacity channel , throughput, cost

Rasio

Signal to Noise Ratio, bit error ratio, CMMR

konsistensi

Fidelity, liniearitas, reliabilitas (fault tolerance, hazardous/fail rate),

Beban/load

Hysterisis, capasitas channel, download speed, bitrate

Acauan

Akurasi, presisi, bias, error

QoS Metrics usual in Computer Engineering

- Response time
- Throughput
- Availability
- Reliability
- Security
- Scalability
- Extensibility

Response Time Breakdown

Browser Time		Network Time			E-commerce Server Time		
Processing	I/O	Browser to ISP Time	Internet Time	ISP to Server Time	Processing	I/O	Networking

CONGESTION

- Service time (does not depend on the load)
- Congestion (load-dependent)

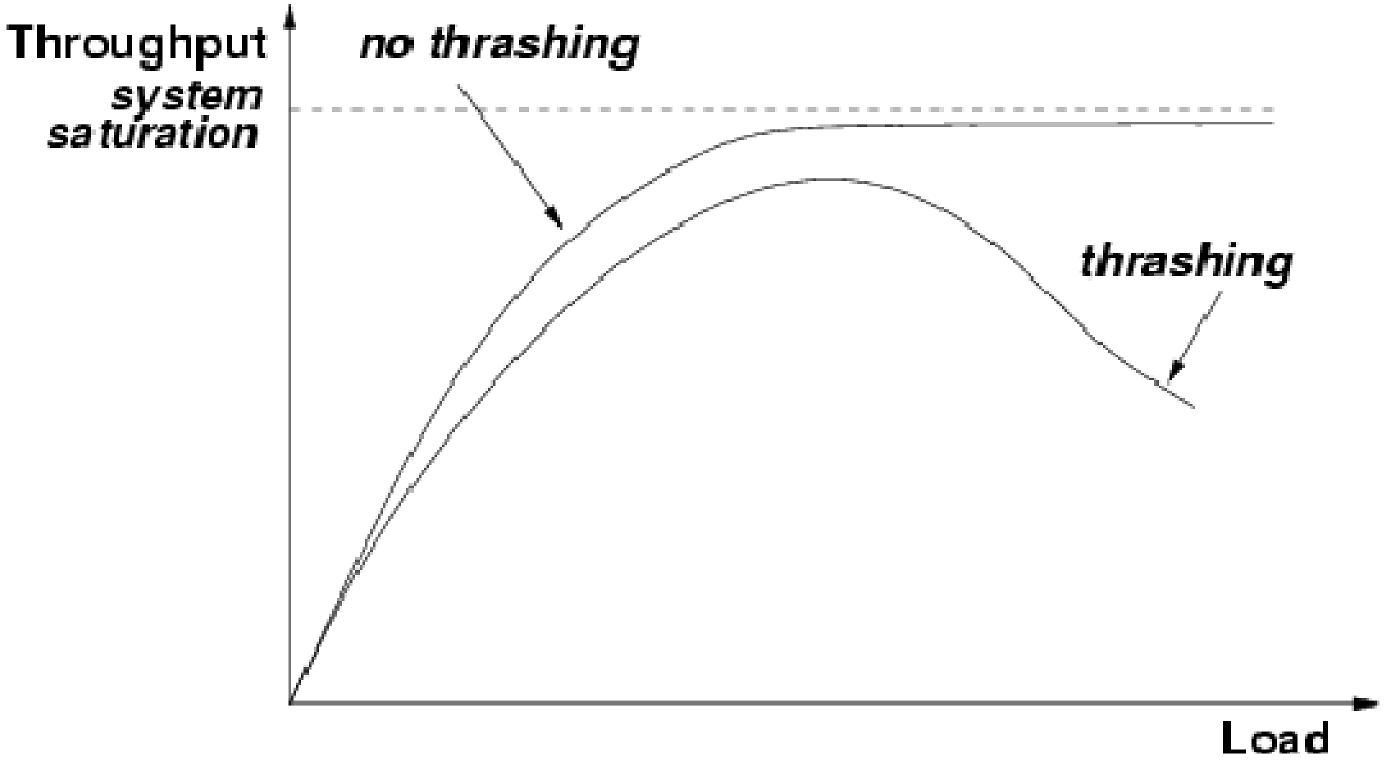
Throughput

- Measured in units of work completed over time. It's a rate.
  - I/O's/sec
  - Page downloads/sec
  - HTTP requests/sec
  - Jobs/sec
  - Transactions per second (tps)

Throughput Example

- An I/O operation at a disk of an OLTP system takes 10 msec on average.
  - What is the maximum throughput of the disk?
  - What is the throughput of the disk if it receives I/O requests at a rate of 80 requests/sec?

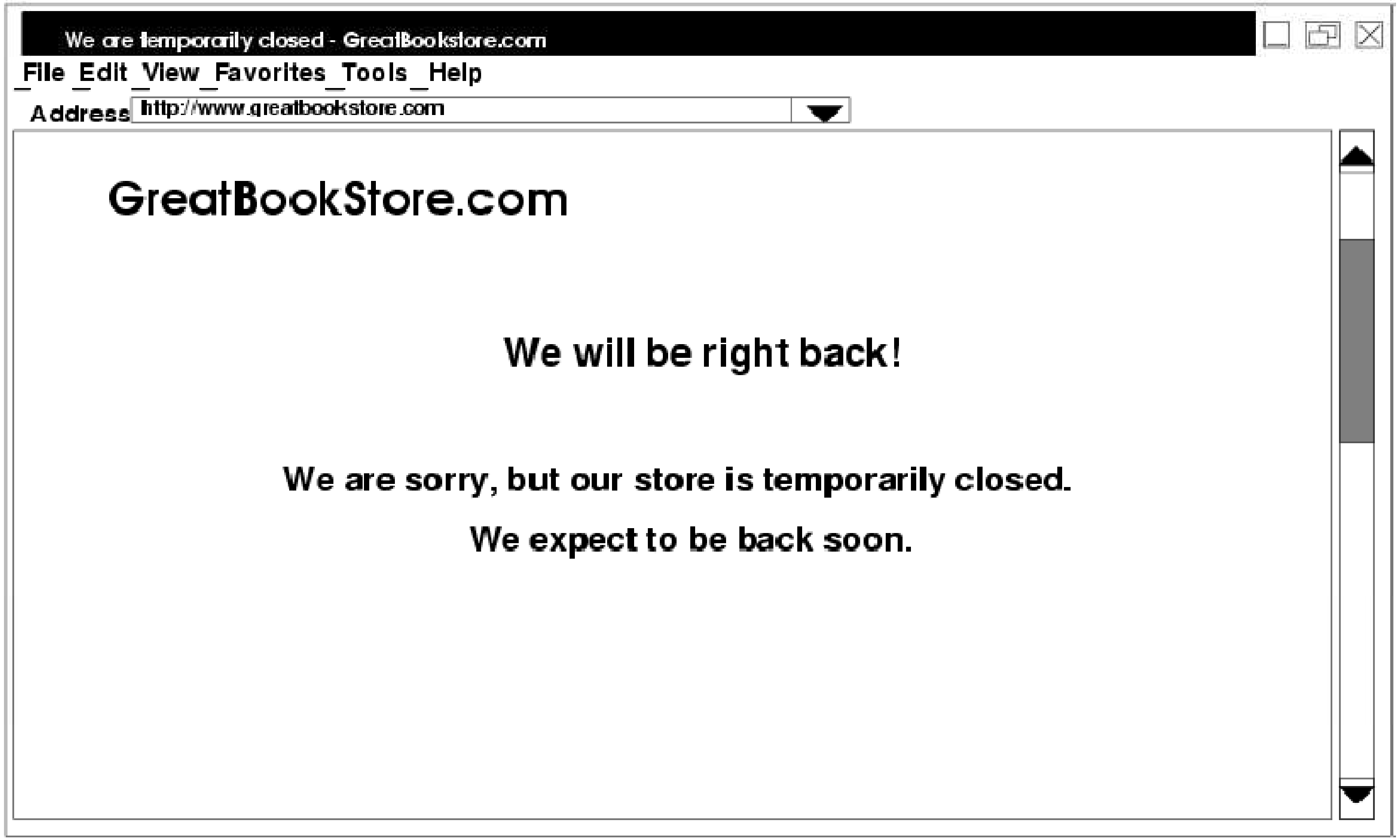
Throughput example



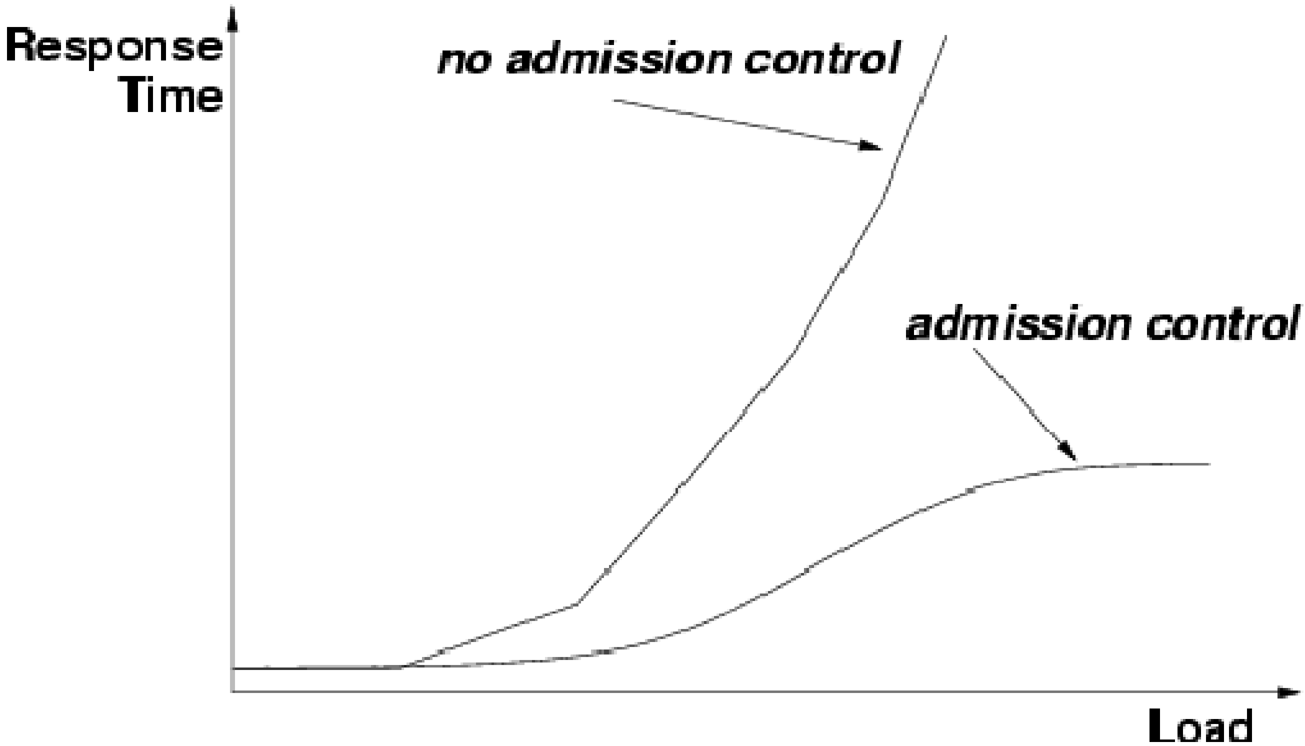
Availability

- Fraction of time a system is available (i.e.,operational).
  - Service interruptions can damage the reputation of a company, may endanger lives, and may cause financial disasters.
  - A system with 99.99% availability over 30 days is unavailable (1-0.9999) x 30 x 24 x 60 = 4.32 minutes.

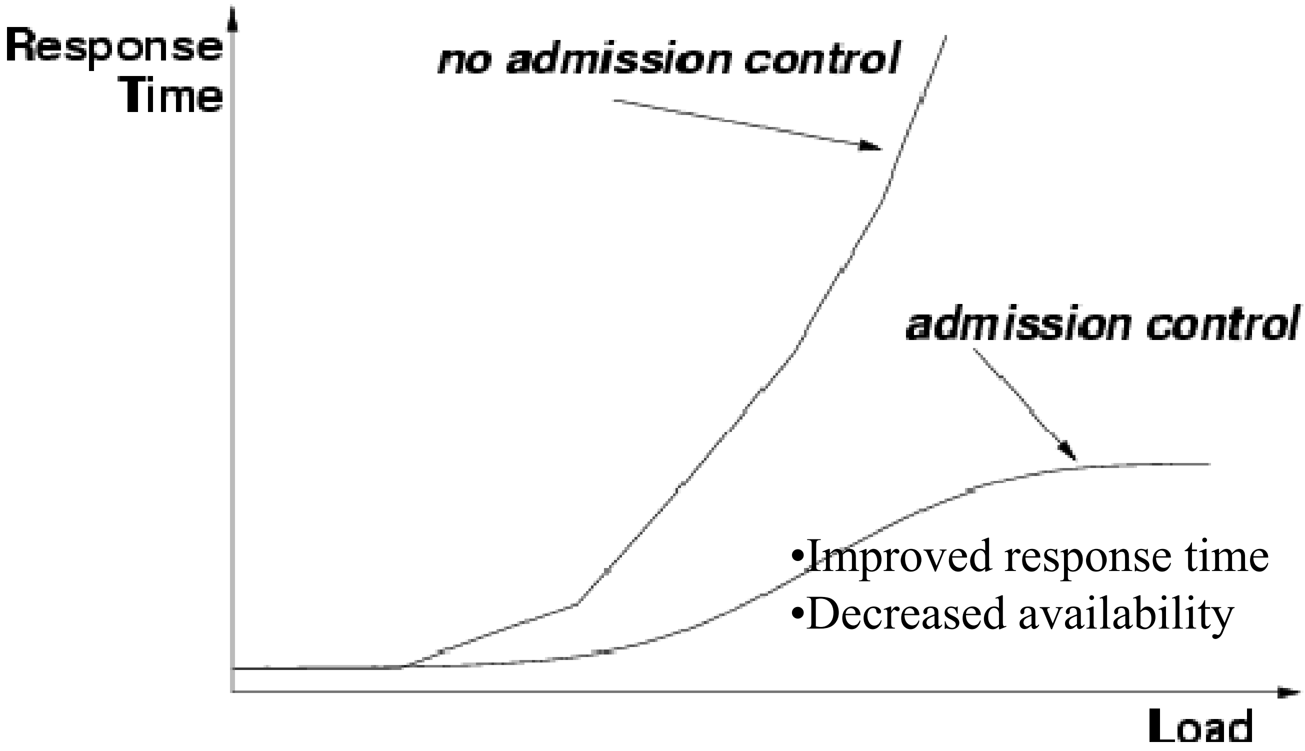
Availability Problems



Admission Control

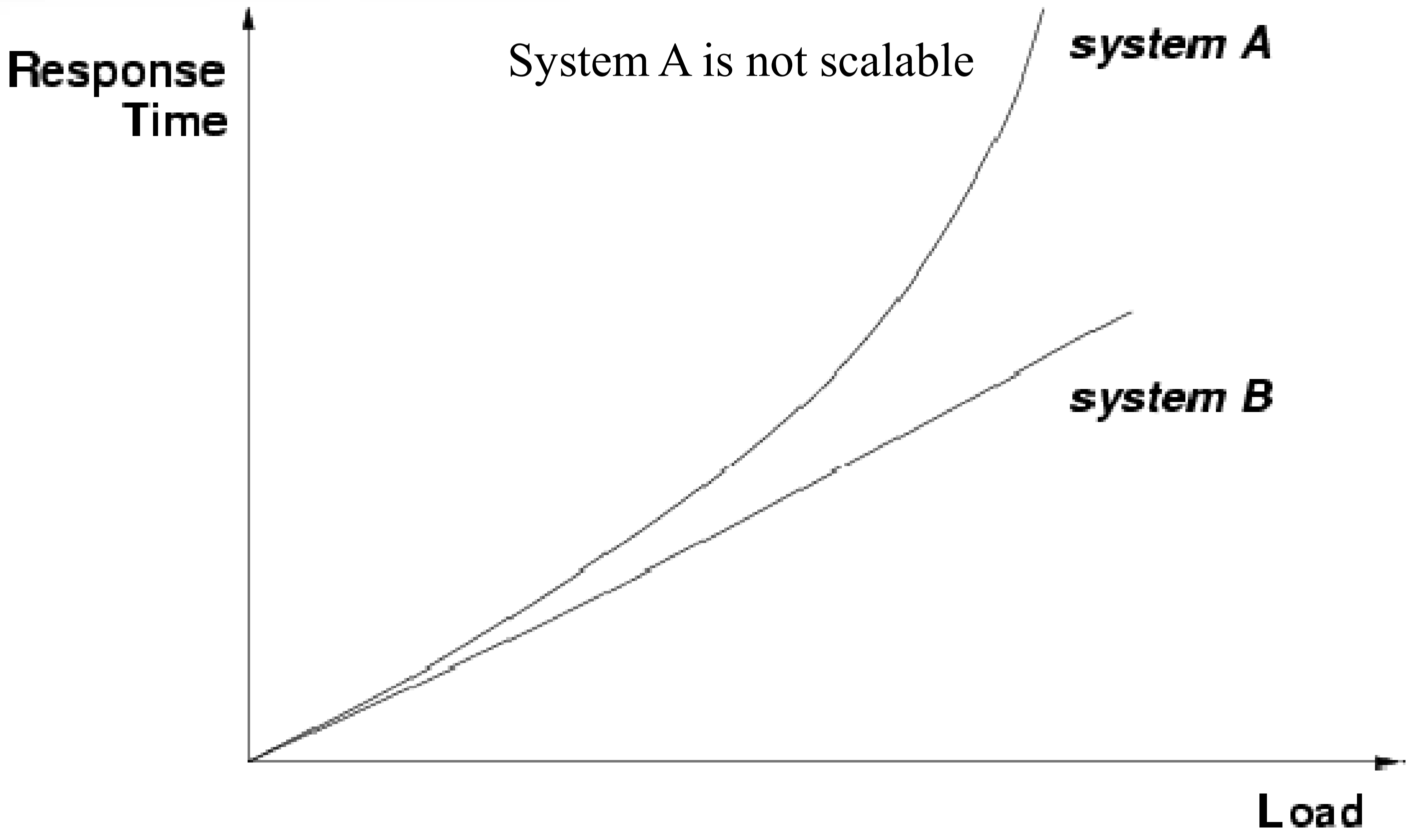


Admission Control



- Improved response time
- Decreased availability

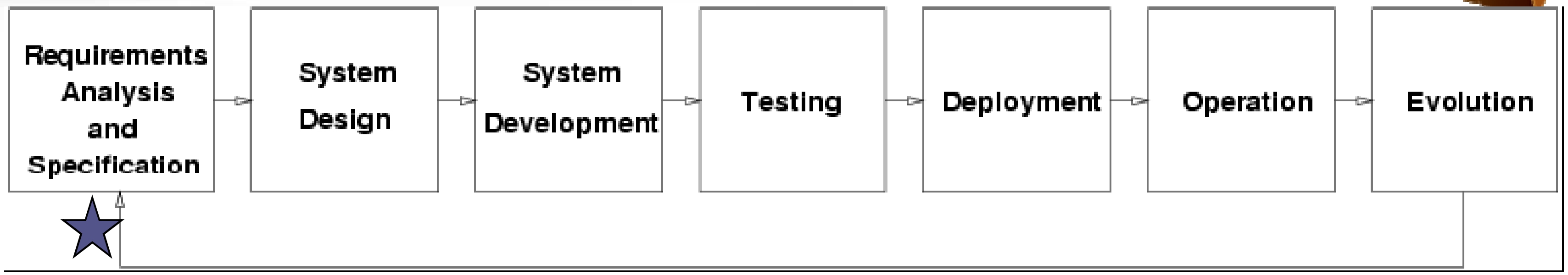
Scalability



Extensibility

- Property of a system to constantly evolve to meet functional and performance requirements.
  - Autonomic computing, self-managing systems, self-healing systems.

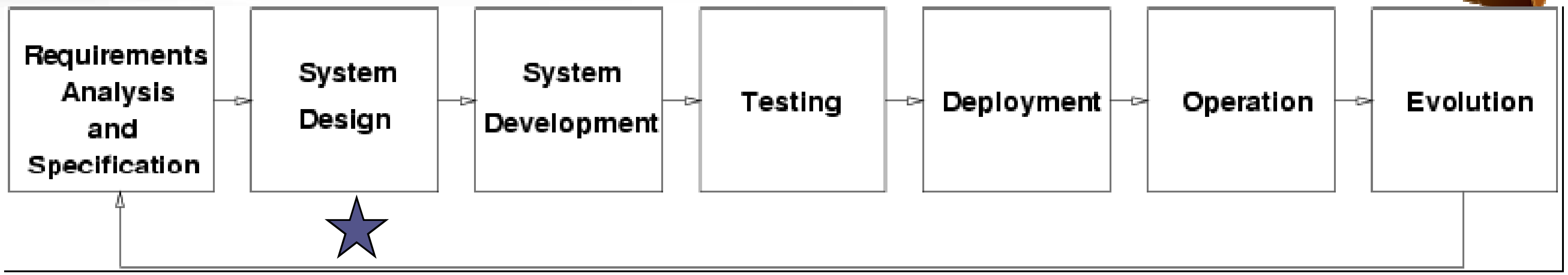
Computer System Lifecycle



- Functional requirements: what the system has to do and on what type of platforms.

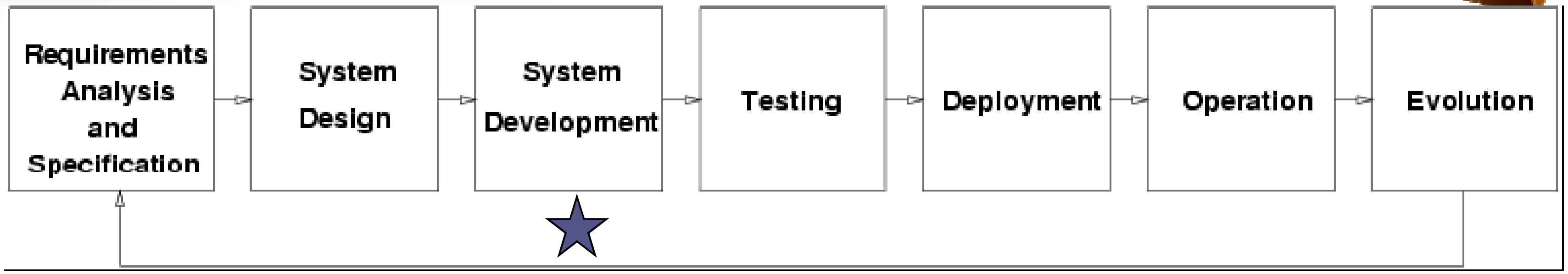
- Non-functional requirements: how well the system has to accomplish its functions. Service Level Agreements (SLA) are established.
In many cases, non-functional requirements have been neglected or considered only at system test time!

Computer System Lifecycle



How will the requirements be met?
- System architecture
- System broken down into components
- Major data structures, files, and databases are designed.
- Interfaces between components are specified

Computer System Lifecycle

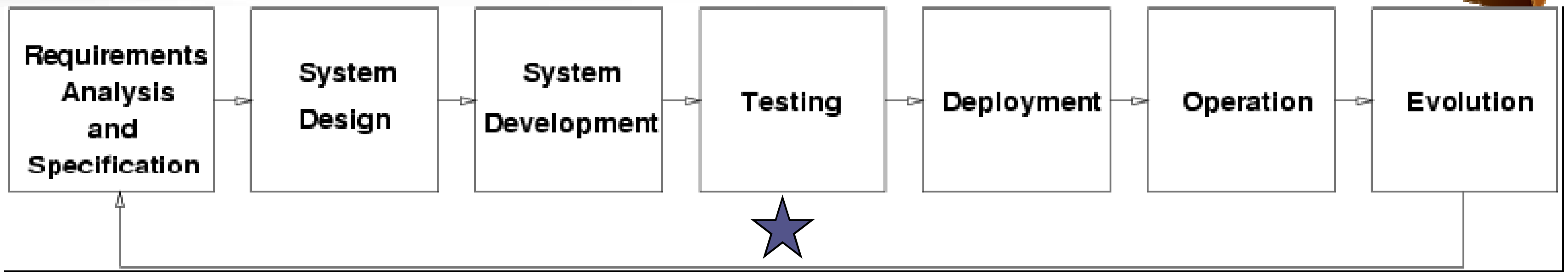


Components are implemented.
- some are new
- some are re-used
- some are adapted

Components are interconnected to form a system

Components should be instrumented as they are built

Computer System Lifecycle

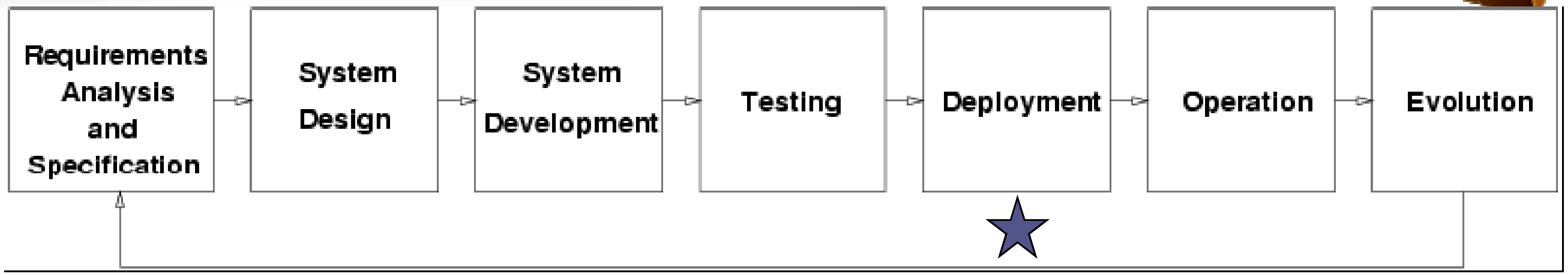


Concurrent with system development, as components become available (unit testing)

Integrated tests are carried out when the entire system is ready.

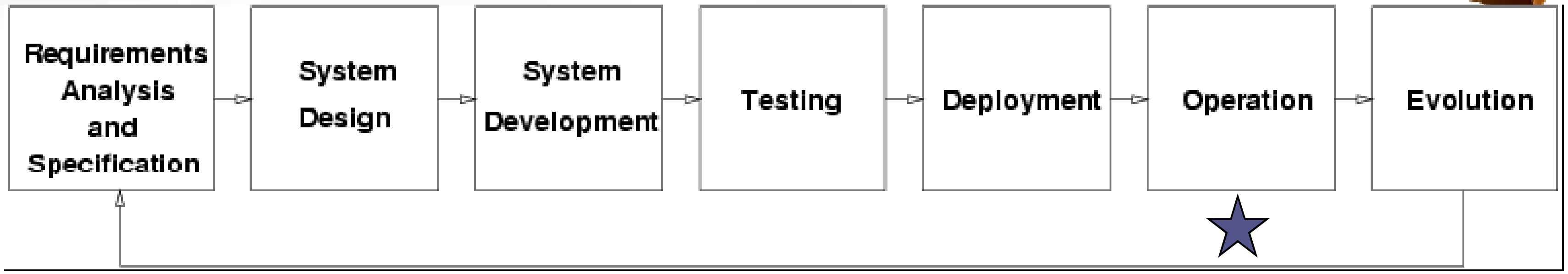
Often, more time is spent in testing functional requirements than in testing non-functional requirements.

Computer System Lifecycle



- Configuration parameters have to be set in order to meet the SLAs.
  - e.g., TCP parameters, database poolsize, maximum number of threads, etc.

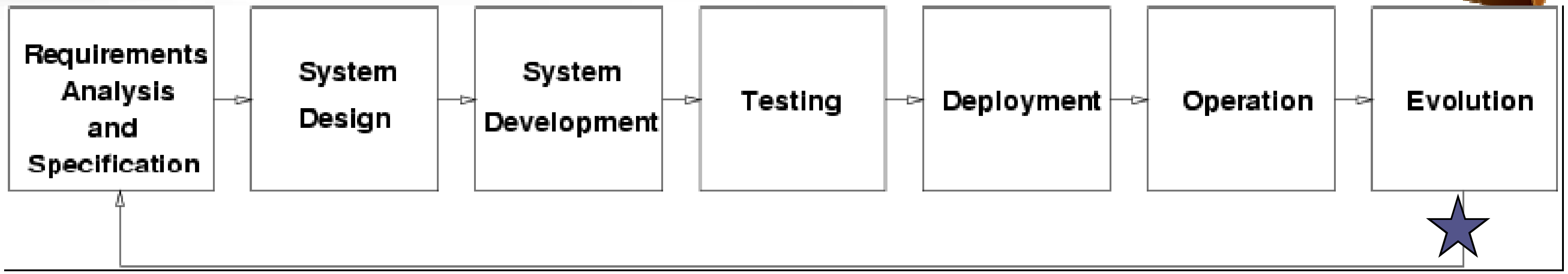
Computer System Lifecycle



Constant monitoring to check if the system is meeting demands:
- workload (peak periods, unusual patterns)
- external metrics (user-perceived)
- internal metrics (help to detect bottlenecks and to fine tune the system)
- availability (external and internal)

May need to dynamically adjust configuration parameters

Computer System Lifecycle

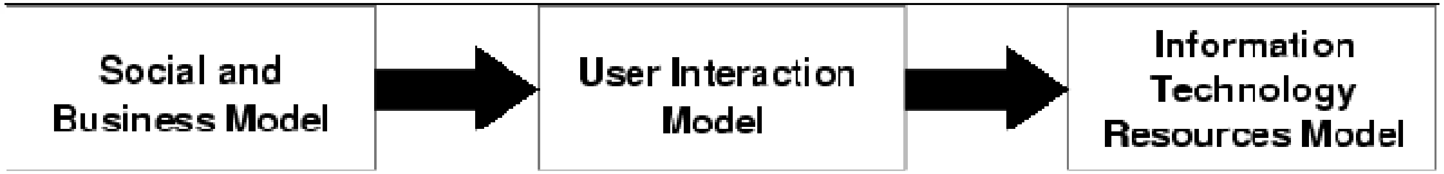


- Systems may need to evolve to cope with new laws and Regulations (e.g., HIPPA)

- Systems may need to evolve to provide new functions (e.g., sale of downloadable MP3 music in addition to CDs)

- How are the IT resources going to cope with evolution in terms of SLAs?

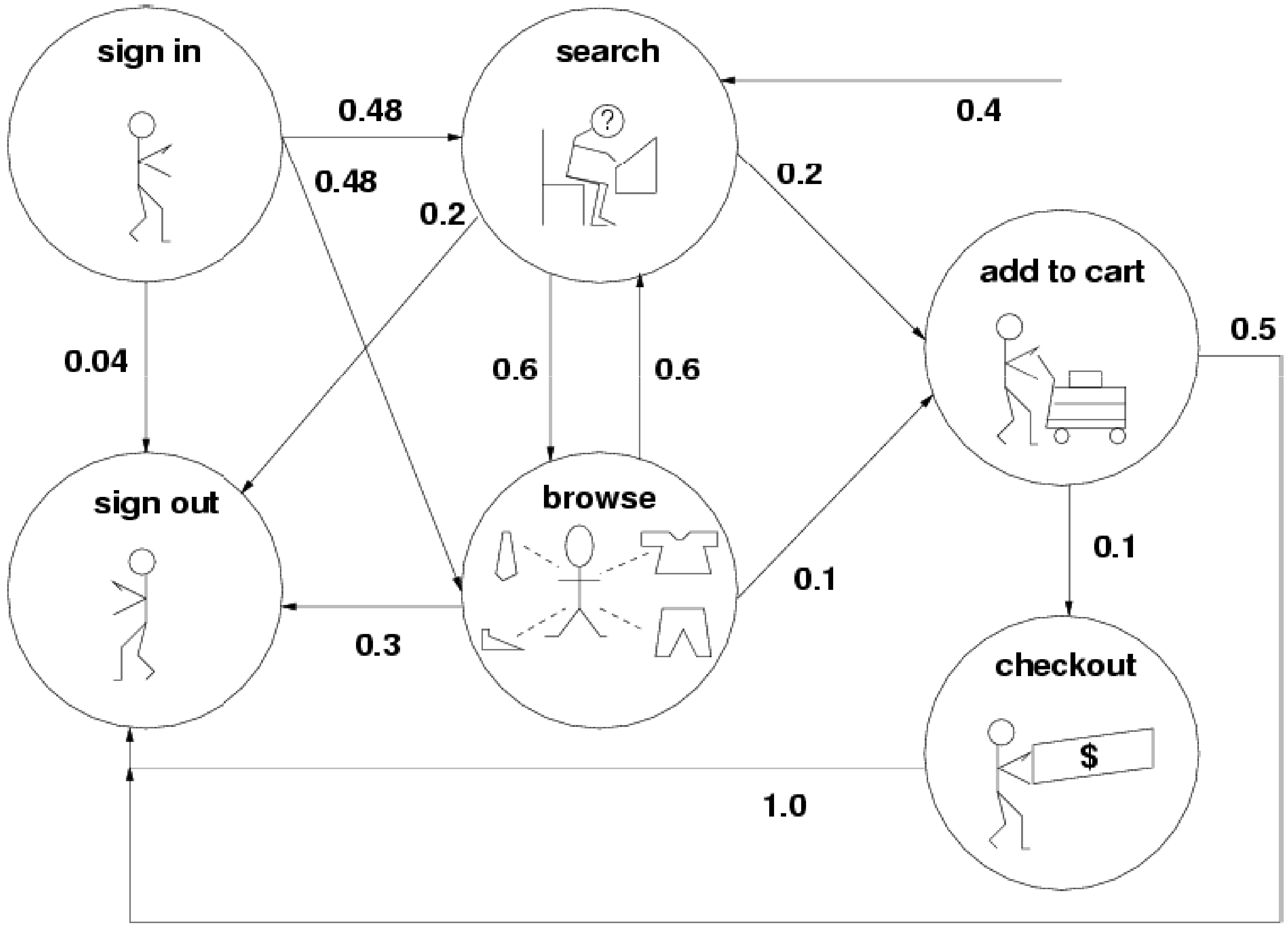
Reference Model for IT



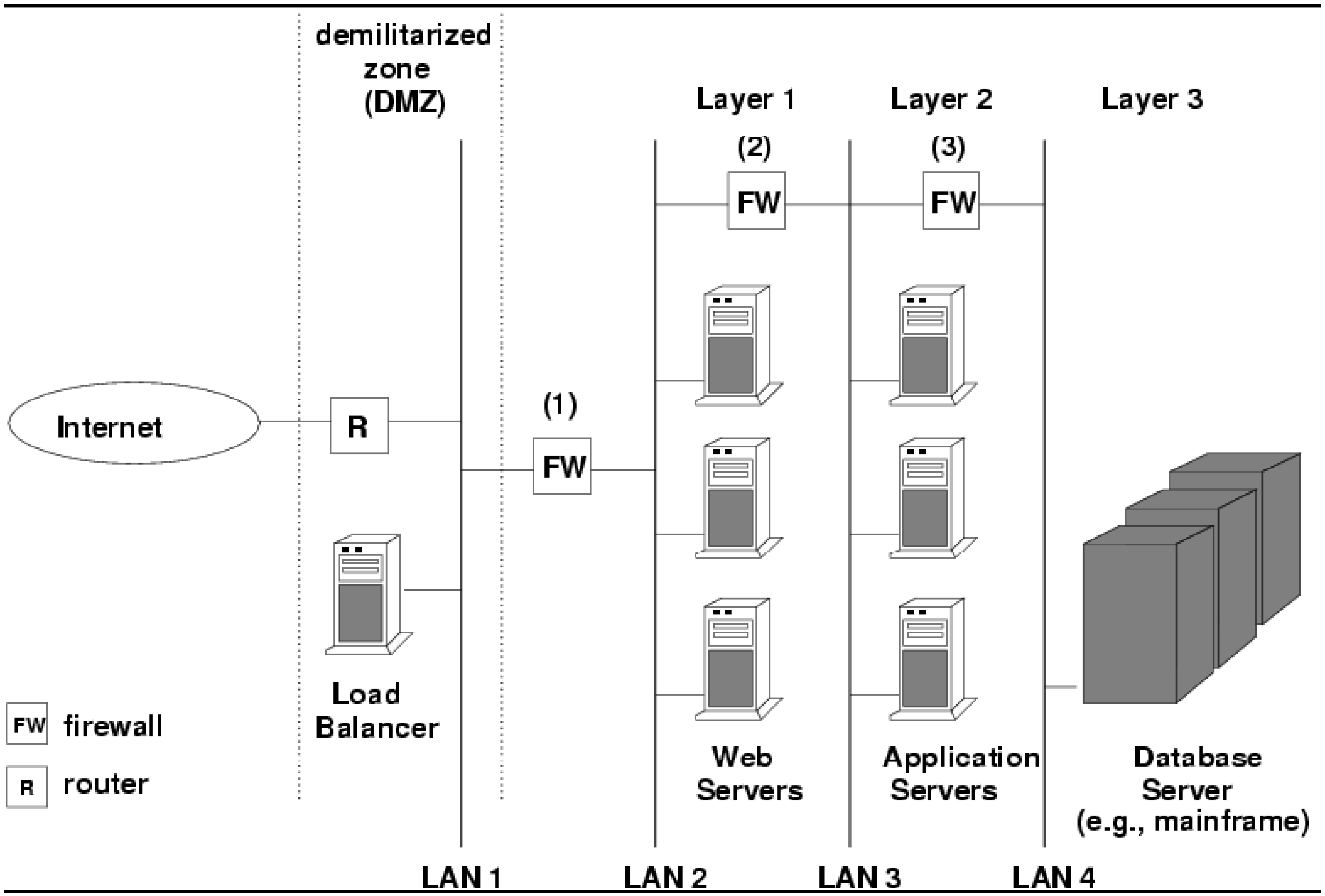
Business Model:
- number of branches
- number and location of ATMs
- number of accounts of each type
-business evolution plans (e.g., mergers)

Social Model
- privacy policy
- accessibility policy

User Model: Customer Behavior Model Graph



IT Infrastructure: Example



PERFORMANCE PROJECT

I HEAR AND I FORGET. I SEE AND I REMEMBER. I DO AND I UNDERSTAND.

—CHINESE PROVERB

Performance Project

1. Measure and compare the performance of window systems of two AI systems.
2. Simulate and compare the performance of two processor interconnection networks.
3. Measure and analyze the performance of two microprocessors.
4. Characterize the workload of a campus timesharing system.
5. Compute the effects of various factors and their interactions on the performance of two text-formatting programs.
6. Measure and analyze the performance of a distributed information system.
7. Simulate the communications controllers for an intelligent terminal system.
8. Measure and analyze the performance of a computer-aided design tool.
9. Measure and identify the factors that affect the performance of an experimental garbage collection algorithm.
10. Measure and compare the performance of remote procedure calls and remote pipe calls.

Performance Project (2)

11. Analyze the effect of factors that impact the performance of two Reduced Instruction Set Computer (RISC) processor architectures.
12. Analyze the performance of a parallel compiler running on a multiprocessor system.
13. Develop a software monitor to observe the performance of a large multiprocessor system.
14. Analyze the performance of a distributed game program running on a network of AI systems.
15. Compare the performance of several robot control algorithms.

Exercise

❖ The measured performance of two database systems on two different work-loads is shown in Table 1.6. Compare the performance of the two systems and show that **a.** System A is better **b.** System B is better

TABLE 1.6 Throughput in Queries per Second

System	Workload 1	Workload 2
A	30	10
B	10	30

The Art of Performance Evaluation

- ❖ Analisa dan evaluasi kinerja sistem merupakan suatu seni pemilihan pengetahuan, metodologi, beban kerja dan perangkat ukur
- ❖ Kemampuan untuk mengartikan nilai, pola gambar, sinyal, nilai statistik

TABLE 1.2 Throughput in Transactions per Second

System	Workload 1	Workload 2
A	20	10
B	10	20

Tahapan Evaluasi

1. Pilihlah teknik evaluasi (pengukuran, simulasi, pemodelan) yang sesuai, metriks kinerja, beban kerja untuk sistem untuk membandingkan kinerja sistem berikut:
   - Dua disk drive
   - Dua sistem pemrosesan transaksi (Transaction Processing System)
   - Dua Algoritma paket pengiriman ulang

2. Melaksanakan pengukuran kinerja dengan tepat. Terdapat perangkat yang meload sistem (load generator) dan perangkat yang mengukur (monitor)
   - Gunakan remote terminal emulator untuk mengukur timesharing
   - Tentukan sistem monitor yang cocok untuk mengukur jumlah instruksi yang dieksekusi prosesor, derajat multiprograming, kestabilan sistem kontrol, respon time dari suatu paket data

Contoh : (lanjutan)

3. Gunakan teknik statistik yang tepat untuk membandingkan beberapa alternatif.

TABLE 1.1 Packets Lost on Two Links

File Size	Link A	Link B
1000	5	10
1200	7	3
1300	3	0
50	0	1

4. Desain pengukuran dan simulasi untuk mencari nilai optimum

TABLE 1.3 Comparing the Average Throughput

System	Workload 1	Workload 2	Average
A	20	10	15
B	10	20	15

TABLE 1.4 Throughput with Respect to System B

System	Workload 1	Workload 2	Average
A	2	0.5	1.25
B	1	1	1

TABLE 1.5 Throughput with Respect to System A

System	Workload 1	Workload 2	Average
A	1	1	1
B	0.5	2	1.25

PROFESSIONAL ORGANIZATIONS, JOURNALS, AND CONFERENCES

- *ACM SIGMETRICS*: The Association for Computing Machinery's
- *IEEE Computer Society*: The Institute of Electrical and Electronic Engineers (IEEE) Computer Society
- *ACM SIGSIM*: The ACM's special interest group on simulation
- *CMG*: The Computer Measurement Group, Inc.
- *IFIP Working Group 7.3*: The International Federation for Information Processing
- *The Society for Computer Simulation*:
- *SIAM*: The Society for Industrial and Applied Mathematics
- *ORSA*: The Operations Research Society of America
- **Standard Performance Evaluation Corporation**

SYSTEMATIC APPROACH TO PERFORMANCE EVALUATION

SYSTEMATIC APPROACH TO PERFORMANCE EVALUATION

1. *State Goals and Define the System*
2. *List Services and Outcomes*
3. *Select Metrics*
4. *List Parameters*
5. *Select Factors to Study*
6. *Select Evaluation Technique*
7. *Select Workload*
8. *Design Experiments*
9. *Analyze and Interpret Data*
10. *Present Results.*

SYSTEMATIC APPROACH TO PERFORMANCE EVALUATION

- **State goals and define the system**

  Given same hardware and software, definition varies depending upon the goals.

Example:

1. Given:two CPUs

Goal: impact on response time of interactive users

Wanted: better timesharing system

(external component)

2. Given:two CPUs with different ALUs

Goal: which ALU

Wanted: best implementation of an architecture (internal component)

What are the boundaries?

❖ **List services and outcomes**

Services:

- Network transports user packets
- Database system responds to queries

Outcomes:

- Packets are lost or delayed.
- Database queries are answered wrongly or hung due to a deadlock.

  Define what is acceptable - what not?

The list of services and outcomes helps later in selecting the metrics and workloads.

❖ **Select Metrics:**

Criteria to compare performance

- Speed
- Accuracy
- Availability of services

List parameters

System parameters:

Software and hardware characteristics which generally do not vary among various installations of the system.

Workload parameters:

Characteristics of the user's requests, that typically vary from one installation to another.

Keep the list as comprehensive as possible!

- **Select factors**

  Factors are the parameters that vary during evaluation, their values are called **levels.**

  Parameters with high impact on systems performance should be factors.

- **Select evaluation technique**
  - analytic modeling
  - Simulation
  - measuring a real system
- **Select workload**
  - Analytic model: Probability/Distribution of requests.
  - Simulation: Traces of requests.
  - Measurement:
  - User scripts, sample problem.

- **Design an experiment**

  Sequence of steps (simulations, measurements) that offer maximal information, maximal coverage with minimal effort.

  Fractional factorial experimental design.

  Two phases:

  1. many factors, few levels
  2. reduced factors, increased levels

- **Analyze and interpret data**
  - Deal with randomness and variability.
  - Interpret the results of the analysis.
  - Prepare to draw conclusions.

- **Present data properly**
  - Spreadsheets, charts and graphics
  - no statistical jargon

The complete project consists of severalcycles through theses steps

Case study :

- Consider the problem of comparing remote pipes with remote procedure calls. In a procedure call, the calling program is blocked, control is passed to the called procedure along with a few parameters, and when the procedure is complete, the results as well as the control return to the calling program. **A remote procedure call** is an extension of this concept to a distributed computer system. A program on one computer system calls a procedure object on another system. The calling program waits until the procedure is complete and the result is returned. Remote pipes are also procedure like objects, but when called, the caller is not blocked. The execution of the pipe occurs concurrently with the continued execution of the caller. The results, if any, are later returned asynchronously.

1. *System Definition*: The goal of the case study is to compare the performance of applications using remote pipes to those of similar applications using remote procedure calls. T
2. *Services*: The services offered by the system are the two types of channel calls—remote procedure call and remote pipe.
3. *Metrics*: Due to resource limitations, the errors and failures will not be studied. Thus, the study will be limited to correct operation only. For each service, the rate at which the service can be performed, the time taken for the service, and the resources consumed will be compared. The resources are the local computer (client), the remote computer (server), and the network link. This leads to the following performance metrics: **(a)** Elapsed time per call
   (b) Maximum call rate per unit of time, or equivalently, the time required to complete a block of *n* successive calls
   (c) Local CPU time per call
   (d) Remote CPU time per call
   (e) Number of bytes sent on the link per call

4. *Parameters*: The system parameters that affect the performance of a given application and data size are the following:

(a) Speed of the local CPU
(b) Speed of the remote CPU
(c) Speed of the network
(d) Operating system overhead for interfacing with the channels
(e) Operating system overhead for interfacing with the networks
(f) Reliability of the network affecting the number of retransmissions required

The workload parameters that affect the performance are the following:

(a) Time between successive calls
(b) Number and sizes of the call parameters
(c) Number and sizes of the results
(d) Type of channel
(e) Other loads on the local and remote CPUs
(f) Other loads on the network

5. *Factors*: The key factors chosen for this study are the following:
   (a) Type of channel. Two types—remote pipes and remote procedure calls—will be compared.
   (b) Speed of the network. Two locations of the remote hosts will be used—short distance (in the campus) and long distance (across the country).
   (c) Sizes of the call parameters to be transferred. Two levels will be used—small and large.
   (d) Number *n* of consecutive calls. Eleven different values of *n*—1, 2, 4, 8, 16, 32, ..., 512, 1024—will be used.
   The factors have been selected based on resource availability and the interest of the sponsors. All other parameters will be fixed. Thus, the results will be valid only for the type of CPUs and operating systems used. The retransmissions due to network errors will be ignored (not included in the measurements). Experiments will be conducted when there is very little other load on the hosts and the network.
6. *Evaluation Technique*: Since prototypes of both types of channels have already been implemented, measurements will be used for evaluation. Analytical modeling will be used to justify the consistency of measured values for different parameters.

7. *Workload*: The workload will consist of a synthetic program generating the specified types of channel requests.
8. *Experimental Design*: A full factorial experimental design with $2^3 \times 11 = 88$ experiments will be used for the initial study.
9. *Data Analysis*: Analysis of Variance (explained in Section 20.5) will be used to quantify the effects of the first three factors and regression will be used to quantify the effects of the number n of successive calls.
10. *Data Presentation*: The final results will be plotted as a function of the block size n.

EXERCISES

Choose a system for performance study. Briefly describe the system and list

a. Services

b. Performance metrics

c. System parameters

d. Workload parameters

e. Factors and their ranges

f. Evaluation technique

g. Workload

COMMON MISTAKES AND HOW TO AVOID THEM

Wise men learn by other men's mistakes, fools by their own.
—H. G. Wells

How to avoid Common Mistake

General

1. **No goals**
2. **Biased goals**
3. **Unsystematic approach**
4. **Analysis without understanding the problem**
5. **Incorrect performance metrics**
6. **Unrepresentative workload**
7. **Wrong evaluation technique**
   - Measurement
   - Simulation
   - Analytical modeling

How to avoid Common Mistake

Engineering

8. **Overlooking important parameters**
9. **Ignoring significant factors**
10. **Inappropriate Experimental Design**
11. **Inappropriate Level of Detail**
12. **No Analysis**
13. **Erroneous Analysis**
14. **No Sensitivity Analysis**

Optional

15. **Ignoring Errors in Input**
16. **Improper Treatment of Outliers**
17. **Assuming No Change in the Future**
18. **Ignoring Variability**
19. **Too Complex Analysis**
20. **Improper Presentation of Results**
21. **Ignoring Social Aspects**
22. **Omitting Assumptions and Limitations**

Checklist for Avoiding Common Mistakes in Performance Evaluation

1. Is the system correctly defined and the goals clearly stated?
2. Are the goals stated in an unbiased manner?
3. Have all the steps of the analysis followed systematically?
4. Is the problem clearly understood before analyzing it?
5. Are the performance metrics relevant for this problem?
6. Is the workload correct for this problem?
7. Is the evaluation technique appropriate?
8. Is the list of parameters that affect performance complete?
9. Have all parameters that affect performance been chosen as factors to be varied?
10. Is the experimental design efficient in terms of time and results?
11. Is the level of detail proper?
12. Is the measured data presented with analysis and interpretation?
13. Is the analysis statistically correct?
14. Has the sensitivity analysis been done?
15. Would errors in the input cause an insignificant change in the results?
16. Have the outliers in the input or output been treated properly?
17. Have the future changes in the system and workload been modeled?
18. Has the variance of input been taken into account?
19. Has the variance of the results been analyzed?
20. Is the analysis easy to explain?
21. Is the presentation style suitable for its audience?
22. Have the results been presented graphically as much as possible?
23. Are the assumptions and limitations of the analysis clearly documented?