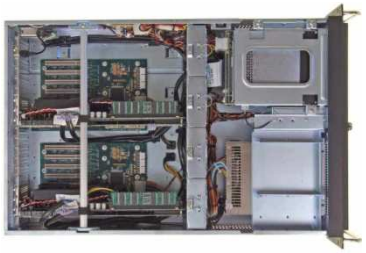
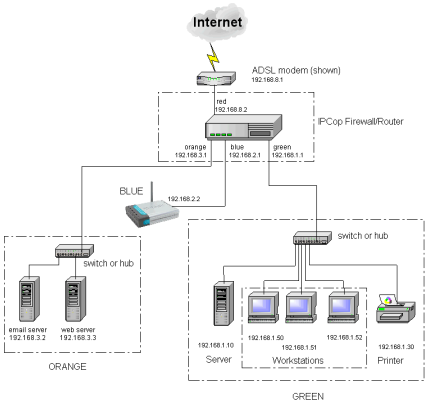
Bab I: Kinerja dan Evaluasi

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

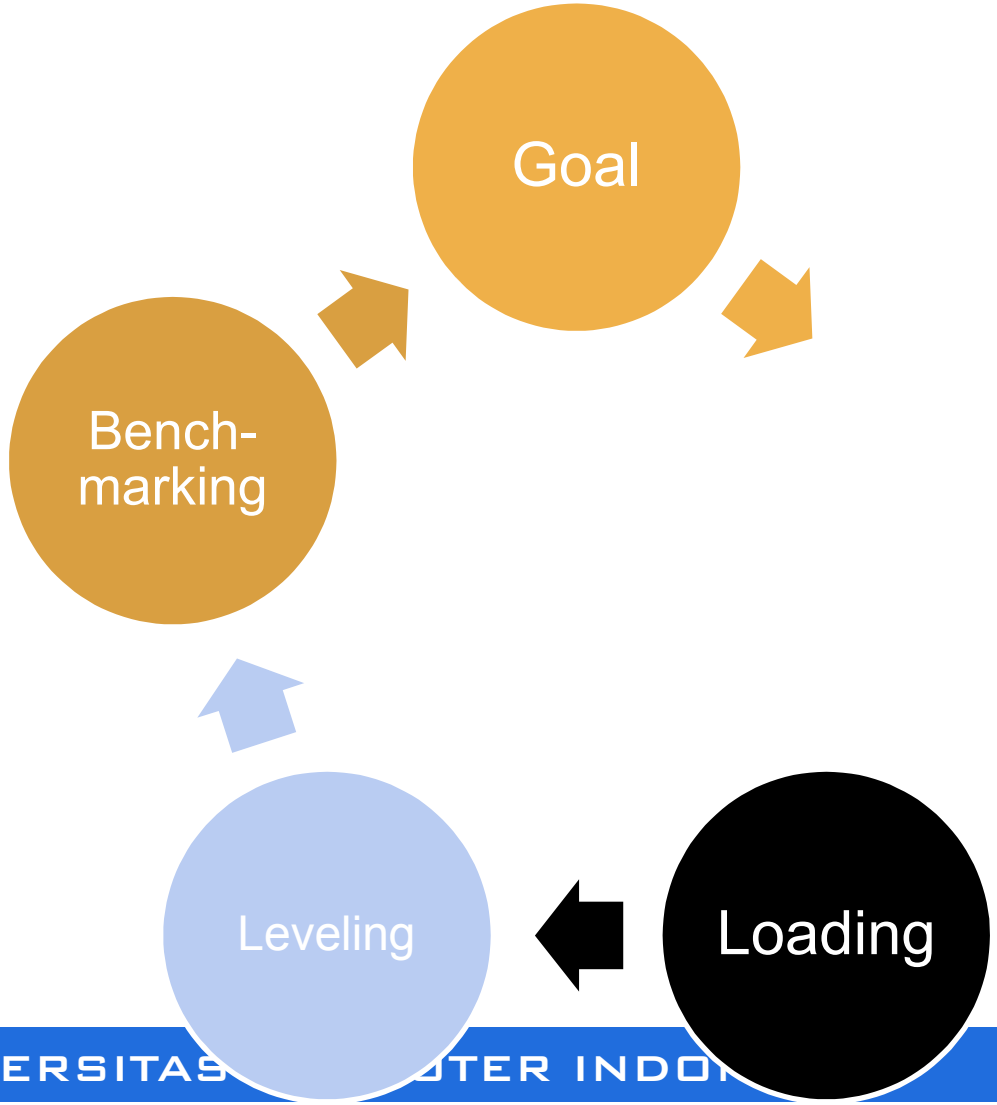
- **Spesifikasi/Requirement**:
  output ideal yang diinginkan (demanding) dari usaha pengeloaan kinerja

Terminologi Kinerja (2)

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

- **Kapasitas**
  Kemampuan sistem mengelola beban

Tahapan Kinerja



Measurement
Simulation
Modeling

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)

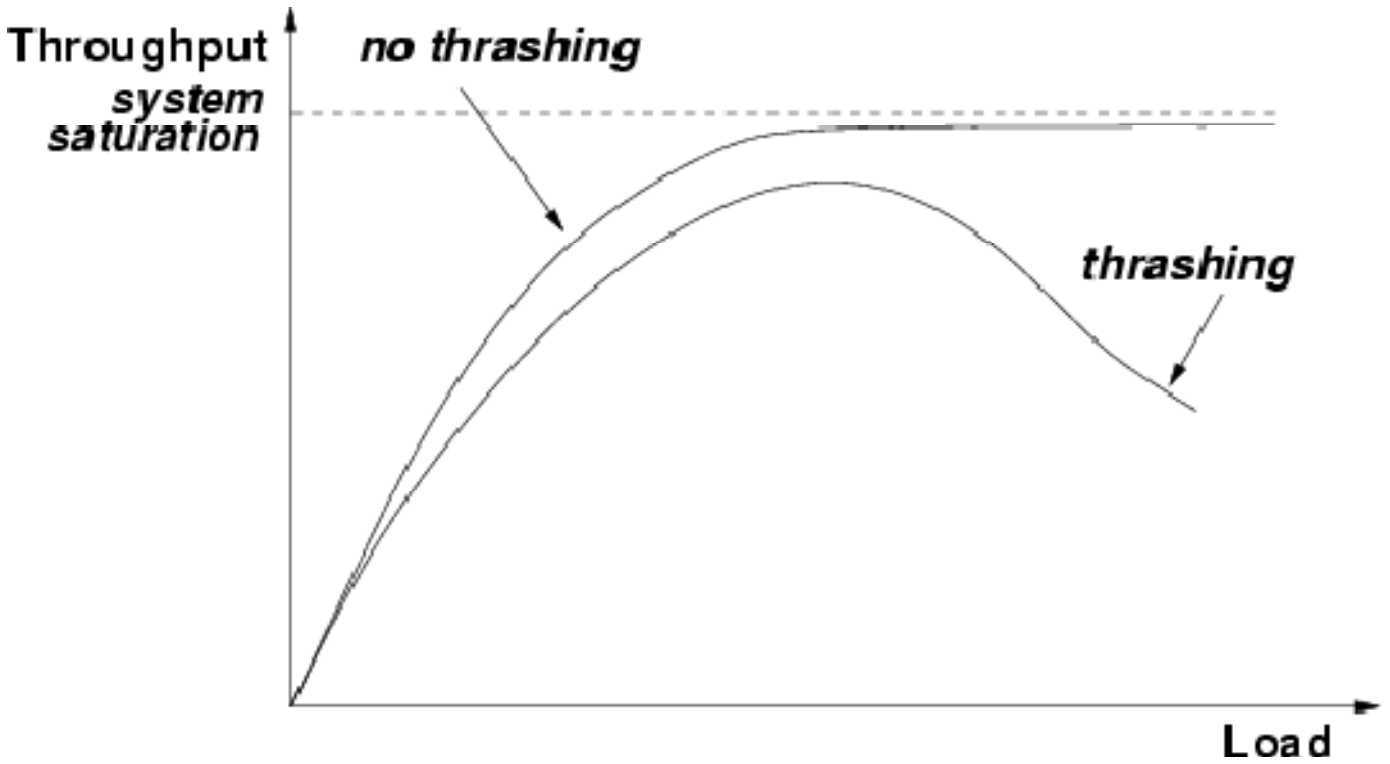
Throughput

- 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

We are temporarily closed - GreatBookstore.com

File Edit View Favorites Tools Help

Address http://www.greatbookstore.com

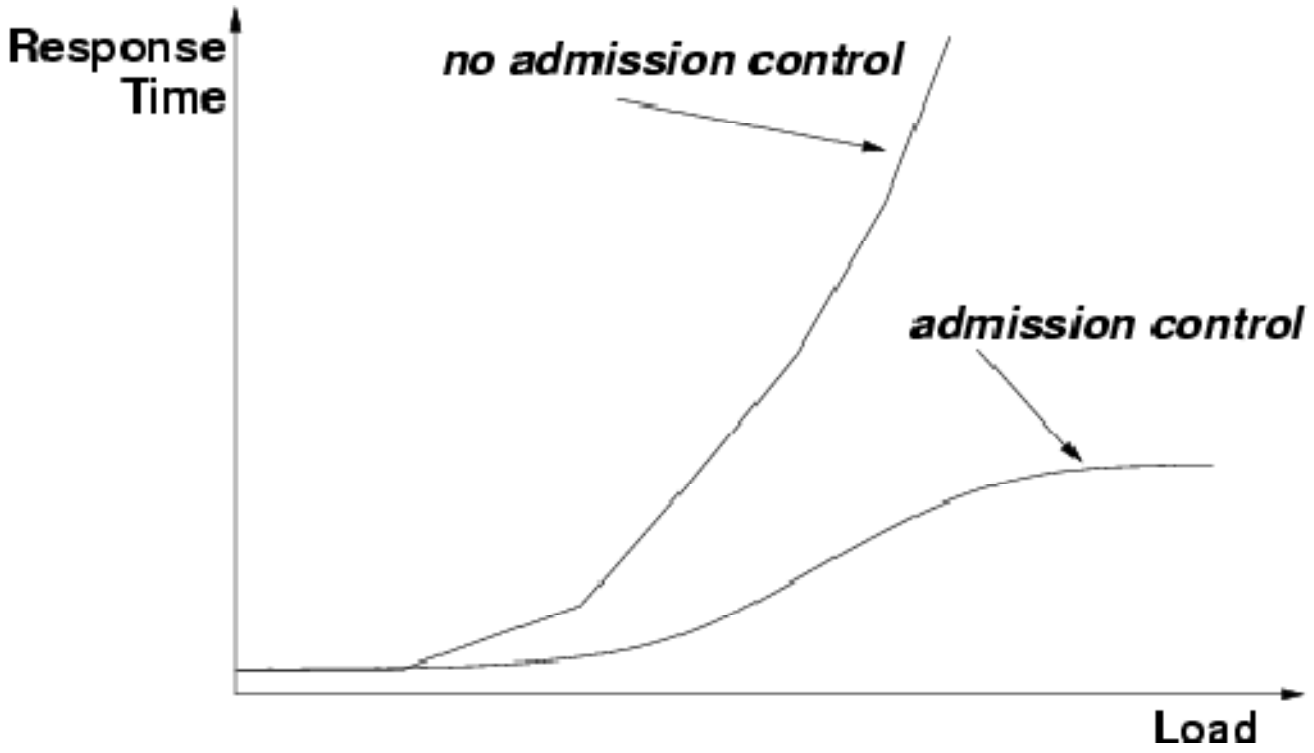
GreatBookStore.com

We will be right back!

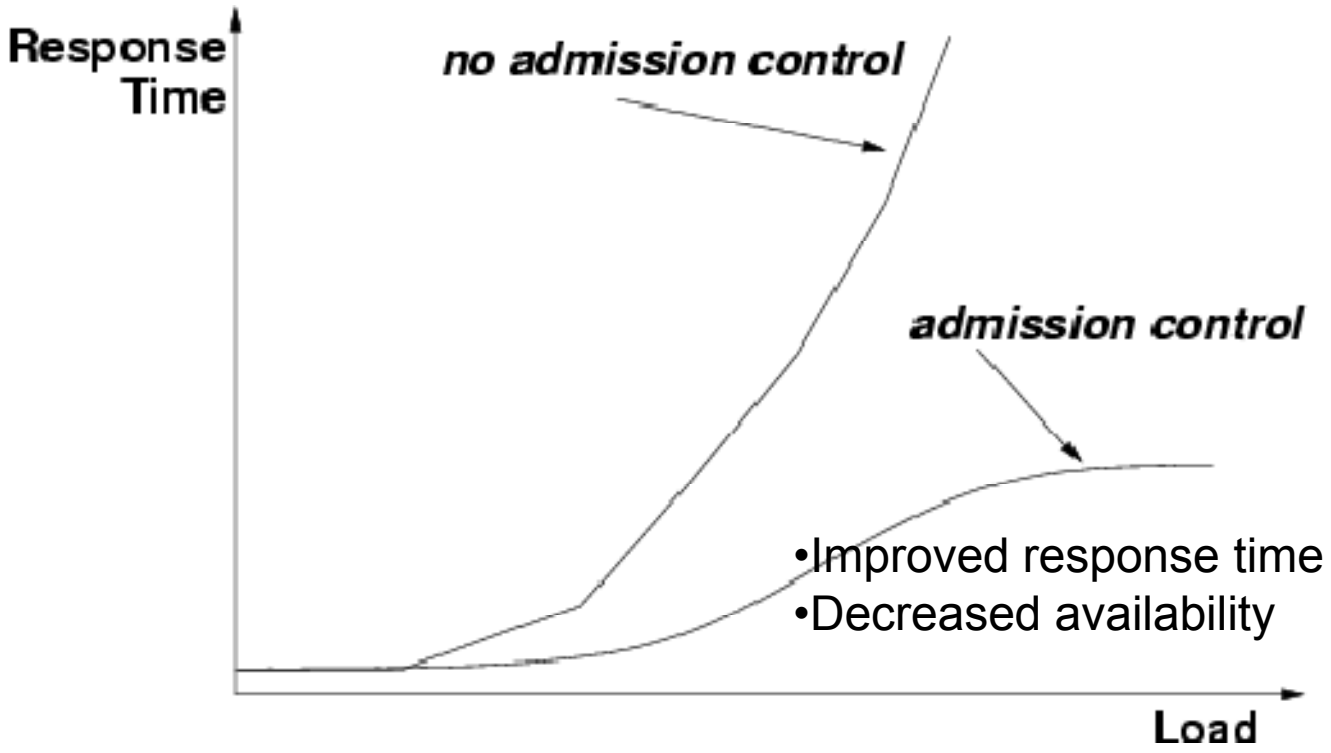
We are sorry, but our store is temporarily closed.

We expect to be back soon.

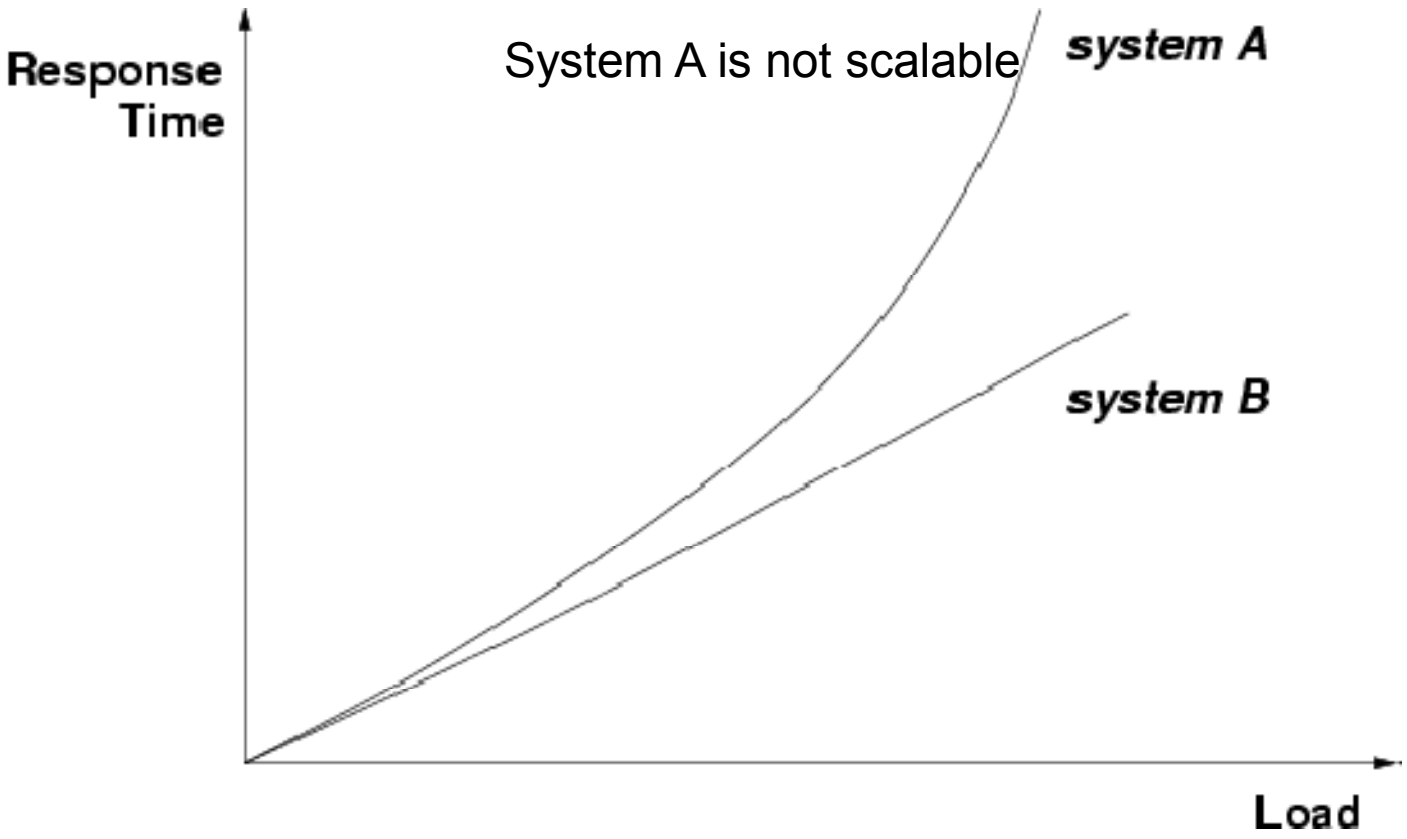
Admission Control



Admission Control



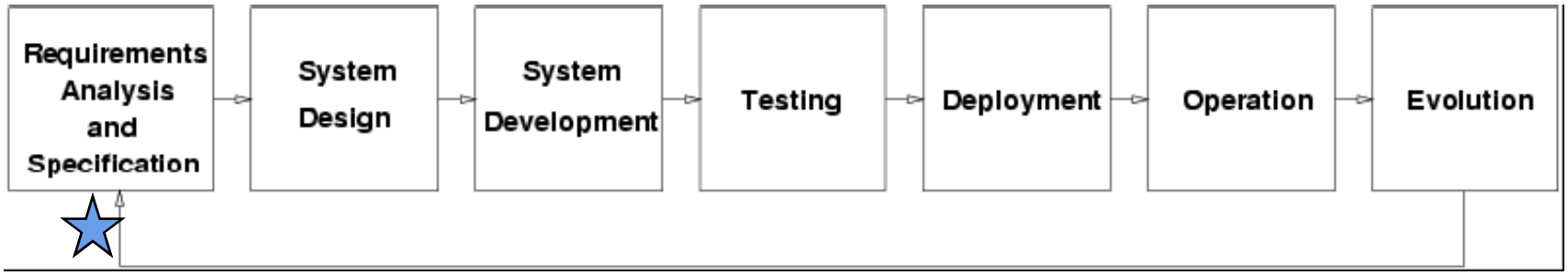
Scalability



Extensibility

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

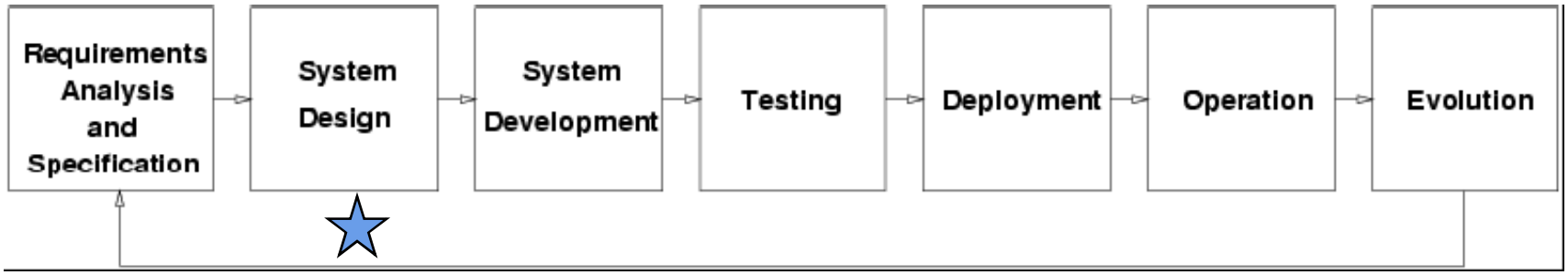
Computer System Lifecycle



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

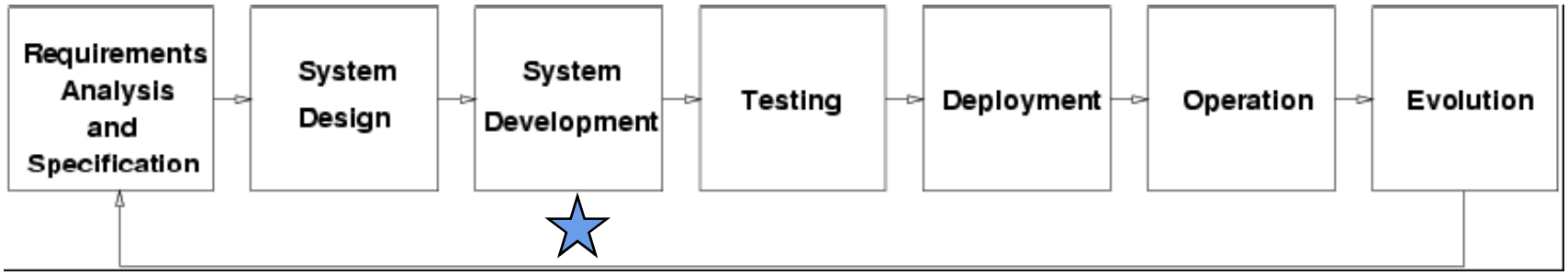
•Non-functional requirements: how well the system has to accomplish its functions. Service Level Agreements (SLA) are established.
In many cases, non-functional requirements have been neglected or considered only at system test time!

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

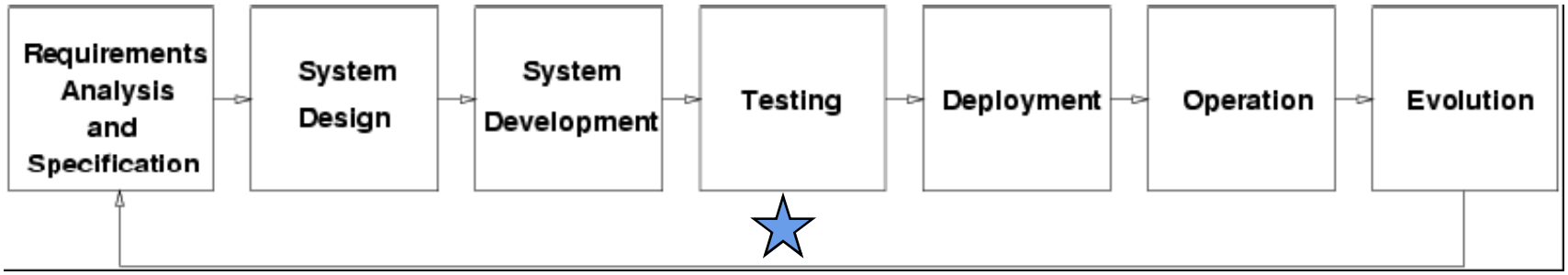
Computer System Lifecycle



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

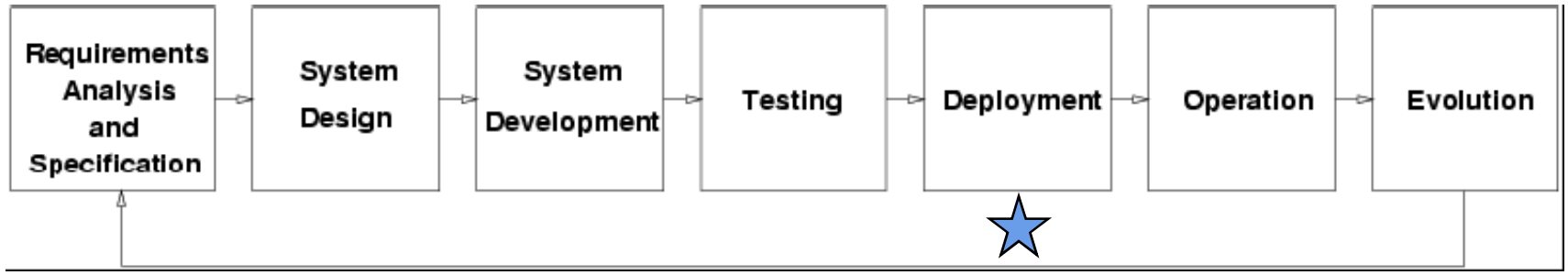


Concurrent with system development, as components become available (unit testing)

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

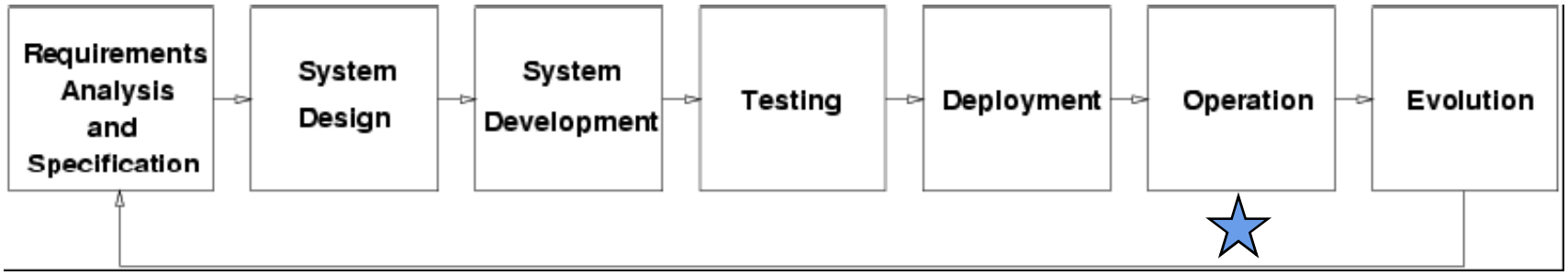
Often, more time is spent in testing functional requirements than in testing non-functional requirements.

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

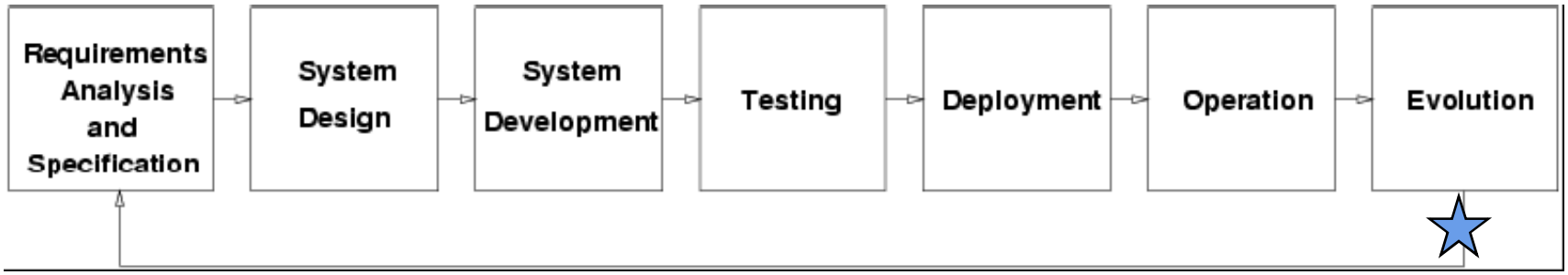


Constant monitoring to check if the system is meeting demands:

- 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

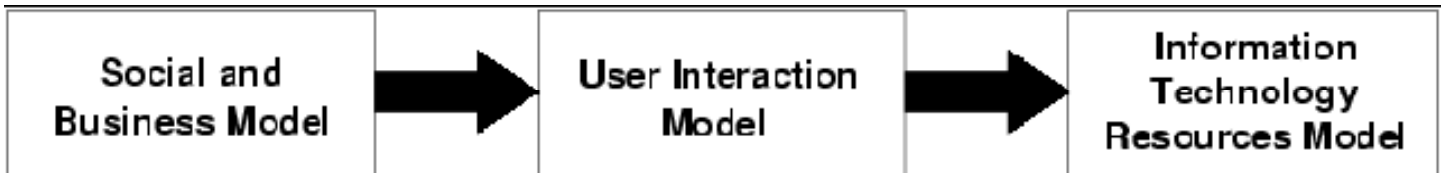


- Systems may need to evolve to cope with new laws and Regulations (e.g., HIPPA)

- 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?

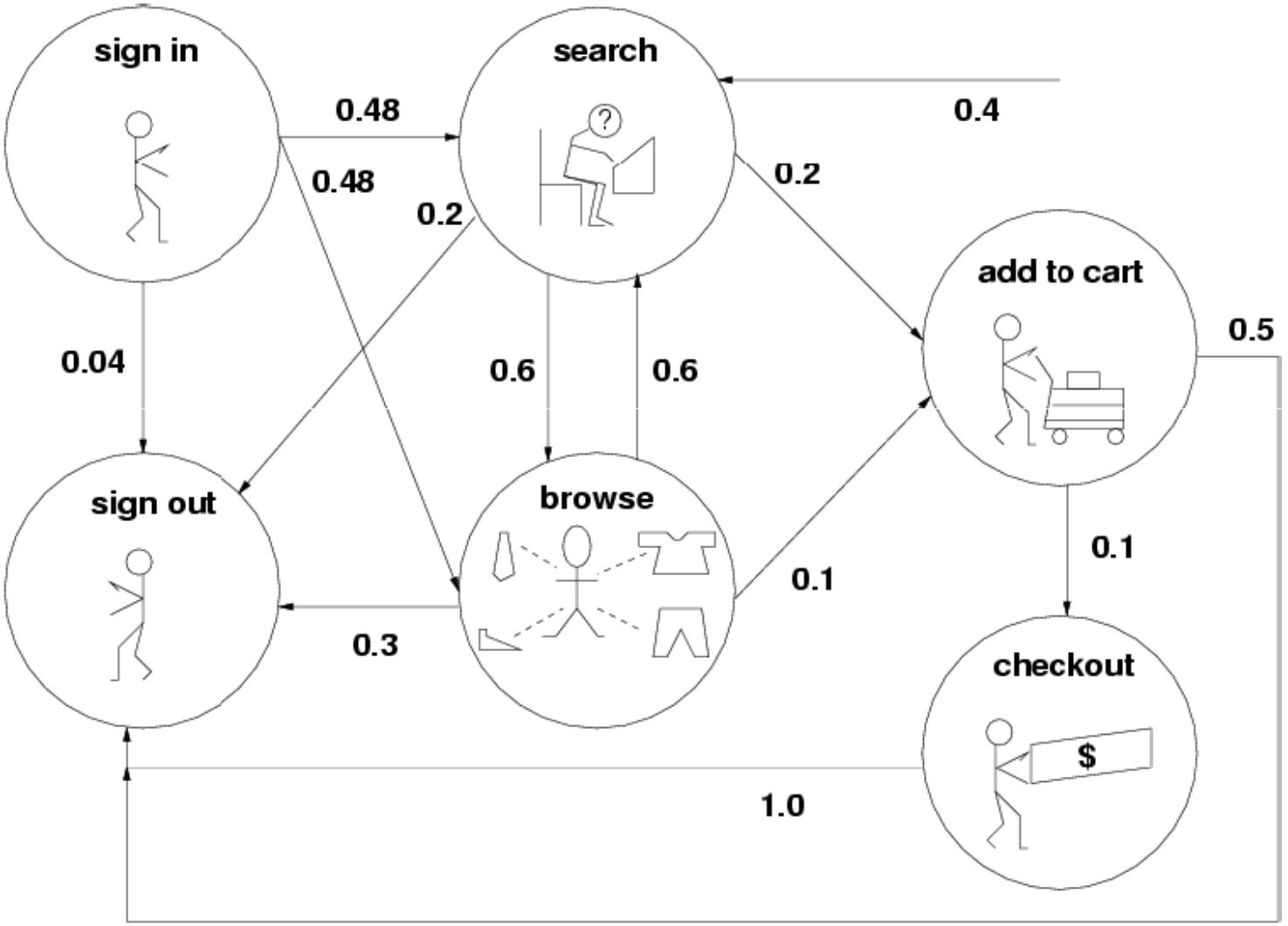
Reference Model for IT



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