
Talon Systems

The Solution Based Engineering Method (SBE)

Revision 1.0

Ralph M. DeFrangesco

September 2001

** This page intentionally left blank*

Talon Systems • 1505 Breckenridge Place • Harleysville, Pa. 19438 • (215) 362-5572
rdefrangesco@hotmail.com

Table of contents

Table of figures

rdefrangesco@hotmail.com

Abstract

This paper presents the Solution Based Engineering (SBE) method for Engineering client-server based systems. "Engineering" a system is Somewhat complex and difficult, often due to the lack of detailed requirements and the knowledge to apply those requirements to a design.

This methodology provides the tools necessary for the Systems Engineer

to design a conceptual architecture based on business and technical drivers (requirements). The SBE method incorporates a set of templates that require both business and technical inputs. The output is a conceptual set of guidelines to help aid in hardware selection, capacity planning and quality assurance.

Although some of the examples presented here relate to Unix servers (RISC), the concepts can be applied to any server including but not limited to, NT or AS400 servers.

Introduction

Every project has a set of requirements, or should have. Often the individuals we count on to supply requirements, both business and technical leaders, don't have a complete idea what those requirements are. Yet, systems are purchased and put in place to support lines of business. In many cases it's a hit or miss design. Under powering a system can be very obvious. Users compete for resources resulting in slow response times. However, over-estimating a system often goes

rdefrangesco@hotmail.com

unnoticed. This often results in spending excess money for resources that may never go used over the lifespan of the server.

The SBE method provides a Systems Engineer with a set of tools to consistently and accurately collect a set of business and technical requirements and apply those requirements against a design.

The SBE has three foundation processes. First, is the requirements collection process. This involves identifying who will have input into the templates, and making sure those input owners fill in the template. Second, take the requirements and apply them to the SBE methodology. If a server is an exact duplicate, a collection tool can be used to check server performance and capacity. If it's a new server, then fill in the templates and follow the step-by-step design process. Finally, a report is produced summarizing inputs, capacity, performance and a high-level server guideline. This detailed report can then be used in the decision making process to determine whether to duplicate an existing server, or design one from scratch.

Terminology

As with any other methodology, an understanding of the terminology is essential. A few critical terms and their definitions are given to aid the reader in understanding this paper.

Concurrent user - A user that is constantly logged on and working.

Iostat - A Unix command that provides performance output including disk input and output, queue length, utilization and transaction rate.

Methodology - A set of procedures.

Process - A course of action or proceeding.

Project - A planned undertaking with a beginning and an end.

Requirements - The needs for a project. Usually given by a business or technical leader.

SPEC – An independent testing organization.

SAR - System Activity Reporter is another Unix command that collects system activity data.

Template - A set of questions defining requirements for a project.

Vmstat - Provides virtual memory statistics including, paging, processes and memory.

SBE considerations

7 step process

The 7 step process is an iterative process. But, the SBE method is only one piece in the process (see Figure 1, process 3). The process starts at the vision process. A business leader wants to increase sales in a certain region for example. Can the server he is running his business on, handle the increase in sales? Once the vision is understood, an agreed upon timeline needs to be established. Often, business opportunities are often missed because of poor timing.

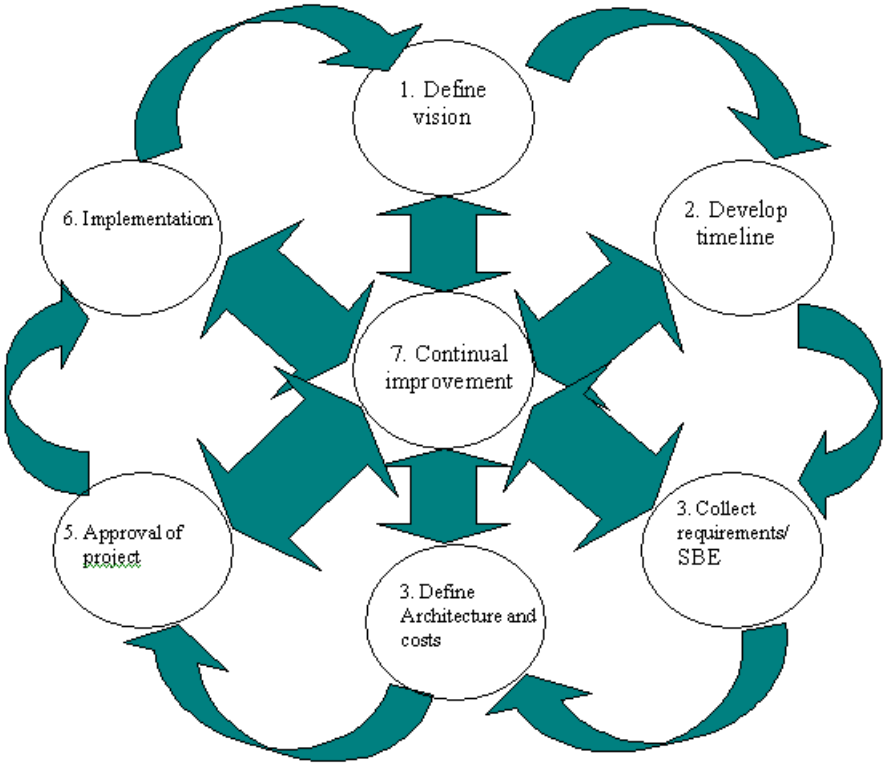


Figure 1: The 7 step process to project engineering

The question, "when does this need to be completed by" needs to be asked. Next, the SBE method is used to collect data and aid in the engineering process. Once a design is created, a better idea as to what the cost will be can be seen. In process 5, an agreement has to be made that the design will meet the requirements at a specified cost and timeline. If the new system involves an upgrade, a formal migration plan needs to be generated. You will notice that the 7 step process is centered around continued review, approvals and a comprehensive project plan. The project plan should include the following:

- A communication plan
- A list of project constraints
- Project risks
- A problem list

- A Work Breakdown Schedule

Drivers

The SBE methodology grew out the saying that, "necessity is the mother of invention". There are not many vendor independent tools for Systems Engineers to design client-server systems. There is the “it’s close to the one we just put in” or “it feels like it should be a…” method. You might as well use Tarot cards. The SBE methodology gives the Systems Engineer a way to consistently collect requirements and aid in the design process.

Quality

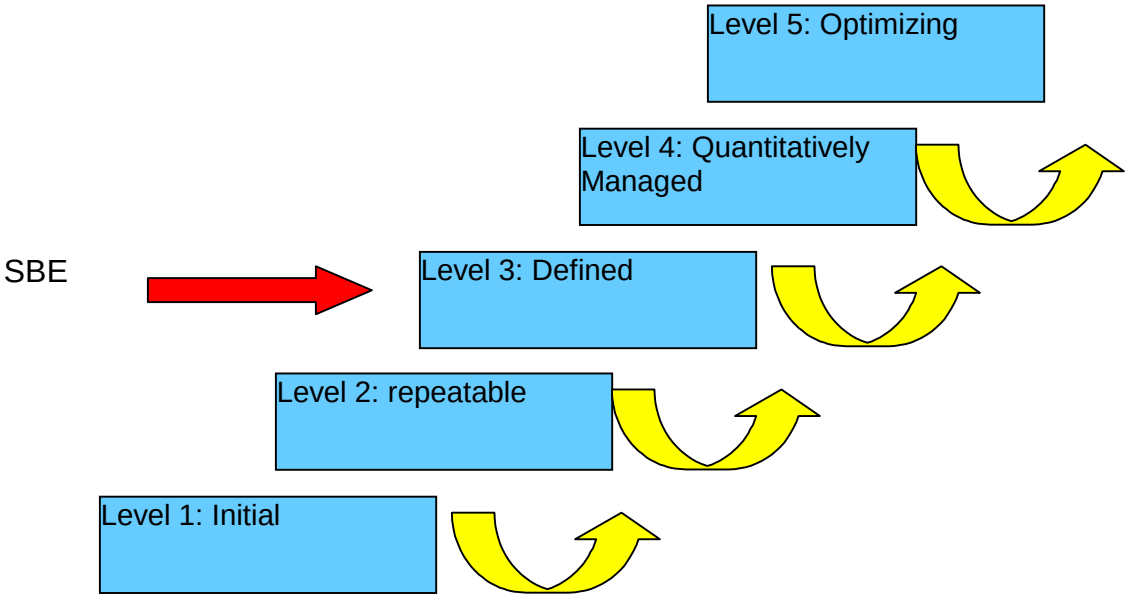
The SBE Methodology does meet CMMI, Six Sigma and most quality initiatives such as Malcom Baldridge. This of course depends on how the methodology is implemented. The key to acceptance into a Quality Assurance program is demonstrating a consistent repeatable process that is well documented and communicated throughout the organization. This document will reference the CMMI model.

CMMI

The Capability Maturity Model Integration (CMMI) was developed at Carnegie Mellon University. Its purpose is to improve on an organizations processes and to aid in the development and maintenance of products and services. The SBE method will help to meet the Engineering process area category of the CMMI. This process area covers:

- Requirements Management
- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation

CMMI has five maturity levels. Each level is characterized by the implementation and institutionalization of the process areas that make up that level. The five levels are as follows:



Capability maturity Model Process Areas

Maturity Level	Improvements Implemented
5. Optimizing	• Develop change infrastructure • Evaluate and deploy improvements • Eliminate causes of defects
4. Quantitatively Managed	• Manage Processes quantitatively • Establish capability baselines
3. Defined	• Establish improvement infrastructure • Deploy and manage processes • Coordinate with non-software groups • Provide organization wide training
2. Repeatable	• Manage requirements • Manage product configurations • Assist and assure policy compliance
1. Initial	• No required processes

Constraints

As with any other methodology, the SBE methodology has constraints. The methodology does not take into account Massively Parallel Processing (MPP). This type of processing is very complex and out of scope for this methodology. SBE also does not take into account down stream capacity. An example might be, if you are inputting data into one system, then processing it on another, SBE will only take into account the capacity on each system separately, not as a whole. Finally, this methodology, as with other methodologies, is only as good as the inputs and the understanding of those inputs into the templates. As the saying goes, "garbage in, garbage out". Therefore, do not rely totally on the output from the tool. This should be a guideline for the seasoned Engineer to help with the decision making process.

The SBE Method

The SBE methodology should not replace common sense. It is impossible to take every design variable into consideration and be correct every time. We want to believe the output because we put accurate inputs into the system. I would equate this to your cars gas gage reading full, but yet you run out of gas. You say to yourself, I can't possibly be out of gas because the gage says my tank is full. A dose of reality sets in when you realize the last time you put gas in your car was over a week ago! So to, the SBE method must be kept in check by the numbers we use for inputs, and the output we get based on those numbers. Just because the output says you need a server with a Spec cfp2000 rating of 200, does not mean you do. When the template was filled out, someone may have "fluffed" the numbers and another may have "fluffed" the numbers and so on...

Flow of the SBE methodology

The SBE methodology really takes place in phases 3 and 4 of the 7 step process to Project Engineering. Phase 3, “answer template questions”, is the requirements collection phase. Once all questions, or as many questions that are as applicable are answered, the design should begin to take on shape. The important point to keep in mind here is to provide as much information and as accurate information as possible to help in the design process. Only then should the design process begin. Once the design is completed, it needs to be priced. I think it stands to reason that an inaccurate design will lead to inaccurate costs.

Requirements

The *technical* requirements are the heart of the SBE methodology. Therefore it is important to make sure that accurate requirements are put into the templates. Inaccurate requirements can give skewed designs resulting in under or oversized systems. Since one of the goals of SBE is to engineer a solution that will meet all of the requirements and allow for growth, it is imperative that accurate data is put into the templates. The core of the SBE method is the requirements. The following should be kept in mind when collecting requirements:

- Customer requests are seldom documented completely.

  - You must identify and actively engage stakeholders to solicit their needs and expectations.
- Early engagement allows us to understand constraints, manage expectations and negotiate priorities.

Therefore, a solid understanding of requirements is in order. Project requirements will come from the people on a project and from the organization itself. They may be technical, non-technical or business in nature. Wherever they come from and whatever their nature, they help to guide the people on a project to a desired end result. One could almost think of them as written in stone, changeable only if agreed upon by all parties involved. Requirements must not be created or changed in a vacuum. The risk of a unilateral change may have undesired downstream affects.

The requirements must answer the; who, what, when and where. This cannot be stressed enough. So many times questions go unanswered until late in the design process. All too many times a simple change has caused *major* design changes and cost over runs.

Templates

If the requirements are the heart of the methodology, then the templates are the brains. These templates are nothing more than questions that need to be answered to help with the design process. All too often during the interview process, questions don't get asked or are misinterpreted. The Engineer doing the design should sit with the person(s) filling out the template(s) to insure their accuracy.

The following areas are generally the most important when it comes to server design; Business, Operations, Applications, Engineering, Helpdesk, Network and Database Administration. Templates are included for these areas as guidelines. The templates are meant to be customized as needed.

Business

The process usually starts here. A business leader will announce that they are opening a new office on a certain date or they need to develop a new

application to help sell a product. The following is a partial list of requirements that need to be collected from the Business unit to properly design a server:

Current date?
Project name?
Project number?
Project description?
Business Unit?
Business sponsor?
Implementation date?
Support criticality?
Support hours?
Does this application require failover?
Location(s)?

Database Administration

Once the business answers some basic questions on the project, the DBA team can begin to design their database, if one is required. The following questions need to be answered by the DBA team:

Database type (Oracle, SQL)?
Number of concurrent users?
Number of total users?
Memory required per user?
Memory required for the database (SGA)?
Will it run on a raw volume or on a filesystem?
Disk space required for database application (Oracle, SQL)?
Disk space required for database data?
Additional disk space for logs, imports, exports, etc?
Number of database I/O's?

Application

A server may have a custom application or COTS software on it. The following questions will need to be answered by the Application team:

Is this a custom application?

If yes, what kind (VB, C, etc)?
How much memory is required for the application?
How much disk is required for the application?
How much data will the application generate?
How much additional disk is required for the application (logs, etc)?
How many users will use the application concurrently?
How many total users of the application?
Is this a CPU, memory or disk intensive application?
If yes, which and how much?
Will there be any daemons or scripts running?
If yes, how much memory or CPU will they use?

Operations

What kind of backups will you do (full, incremental)?
What time will you do the backup?
What is the time estimate to backup this server?
How many tapes will you use?
Will you backup the OS?
Date this server will go into production?
Are there reports that need to be run?
Are there any data transfers that need to occur?
Who do I contact in case of an abbend or failure?

Helpdesk

What hours will you support this server/application?
How many people are trained to support this application?
Is after-hours support required?
What metrics will you collect/report on this application?
Is there an SLA for this application that you must meet?

Network

How many network connections will be provided to this server?
What type (10Mbps/100Mbps Ethernet, FDDI, Token Ring)?
What type of WAN connection is required for this project?
What is the connection speed (56k, etc)?
Will there be redundancy in the connection?
Will you be able to fail it over to another connection?

Engineering

Will there be other applications running on the server (C, JFS, DCE)?
If yes, what kind?
How much memory, disk or CPU will they use?
Total number of users on the system?
What kind of storage will you use (DAS, NAS, SAN's)?
What kind of connection to the storage will you use (Fiber, SCSI)?
How many connections to the storage will there be?
If failover is required, what kind will you use (manual, automatic)?
Will you cluster this server?
Server usage (light, medium, heavy)?

Collection tool

The collection tool is only used when capacity needs to be checked on an existing server because a design *truly* is like one already in production. At minimum, the following data points need to be collected and analyzed from the existing server:

CPU utilization
Memory utilization
Disk utilization
Network traffic profile

CPU utilization
To collect CPU utilization data, use the 'vmstat' command. The following is output from the 'vmstat' command:

#vmstat 5 10

procs			memory			page						faults			cpu		
r	b	w	avm	free	re	at	pi	po	fr	de	sr	in	sy	cs	us	sy	id
4	0	0	1180	2028	6	22	45	0	0	0	0	340	190	30	40	31	10
0	2	0	1190	2120	6	21	0	0	0	0	0	601	222	21	47	29	8

Memory Utilization
The output from the 'vmstat' command also shows memory information. If you are using HP-UX, you can also check /sbin/dmesg and look for the following messages:

Physical: 65536 Kbytes

available: 55336 Kbytes

lockable: 45345 Kbytes

Disk utilization

The ‘iostat’ command gives us an idea what the level of effort that the CPU is putting into disk I/O as well as bits transferred, and seek time. The following is a sample output from the ‘iostat’ command.

#iostat -t 3

```
      tty                  cpu
 tin   tout          us   ni   sy   id
  78     42           2    0   28   60

device    bps   sps   msps

c0t1d0    0     0      0
c0t4d0    33    2     21.6
c0t6d0    10    2     23.8

      tty                  cpu
 tin   tout          us   ni   sy   id
  53     44           4    0   43   22

device    bps   sps   msps

c0t1d0    0     0      0
c0t4d0    41    1     31.1
```

c0t6d0 14 1 43.1

Network traffic profile

To get network statistics, use the 'netstat' command. The 'netstat' command only shows very basic information. Look at the output from the other options given to get a better idea about output from the command.

#netstat -i

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Col
lan0	1497	192.181	a4410.e.h.c	242043	99	118181	17	11781

There are many options to use with the netstat command including:
#netstat -rn
#netstat -I lan0
#netstat -s

And of course there are other commands used to gain network statistic information:
landiag (if using HP)
ping

The System Activity Reporter (sar)

Like the other commands presented here, sar was designed to collect system information to evaluate system performance. Output from the command follows:

#sar -u 5 5

16:18:53	%usr	%sys	%wio	%idle
16:18:58	0	0	0	100
16:19:03	58	28	1	13
16:19:08	84	16	0	0
16:19:13	57	11	31	0
16:19:18	0	6	94	0
Average	40	12	25	23

As with other Unix commands, there are many options that can be used with the sar command string including:

sar -d Reports disk activity information
sar -u Similar to the iostat and vmstat commands
sar -b Reports on buffer cache activity
sar -w Reports system swapping activity

Other tools

Many other tools exist for collecting and analyzing system performance. The tools available of course depend on the platform you have. The following is a partial list of some of the tools available:

Tool	**Manufacturer**
GlancePlus	HP
Measureware	HP
PDT	IBM
PTX	IBM
PerfMon	Sun
TeamQuest Model	TeamQuest
TeamQuest View	TeamQuest

Once this data is collected, it needs to be added to the SBE method to be used in the design process. The following questions need to be asked:

- What design parameters were used on the server?
- How is the server performing?
- Are there any current bottlenecks?

Figure 2.0 gives an example of how it should be merged.

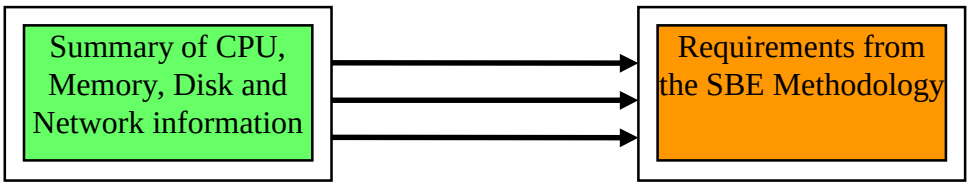


Figure 2: SBE requirements and data merger

Basic Analysis

The following guidelines are a starting point for you to use. These are not written in stone and can vary depending on circumstances and system configuration.

- The more memory used, the better. 97-98% memory utilization is not uncommon. As long as no paging or process deallocation occurs, this is a good use of memory.
- HBA performance should be within an acceptable range for the specific HBA used.
- CPU utilization will vary server to server and only experience will tell you an acceptable range for your circumstance. Again, a basic guideline is that CPU usage should be in the 60-80% range. To get this, add together system, user and wait from the sar or vmstat output.

If users are not experiencing problems and you are within the above guidelines, the server is probably good to clone. This assumes the system is laid out properly. This would include; data is spread across disk correctly and the I/O's per second to the storage are within an acceptable range and finally, that you have collected sample data at multiple times in your production cycle.

Pulling it all together

1.) Add up all the memory requirements including:

- The operating system
- Operating system applications
- The Database
- Scripts
- Any additional applications (e.g. C, Pascal)
-

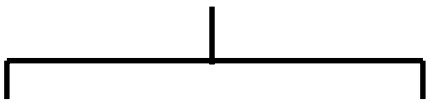
Note: If using the collection tool, use the actual percentage of memory used.

2.) Add up all the storage requirements. The following should be kept in mind when adding up storage:

- Will you be clustering this server?
- Will you be using SCSI or fibre channel connections
- Will you be mirroring, striping or both?
- Will you need an additional copy of the data for testing or backing up the data?

The answers to these questions will lead you to your choice in storage options. The decision tree (Figure 3.0) is an example of what a decision tree might look like. Figure 3.0 should be modified to represent the individual project you are working on.

Storage Option Decision Tree



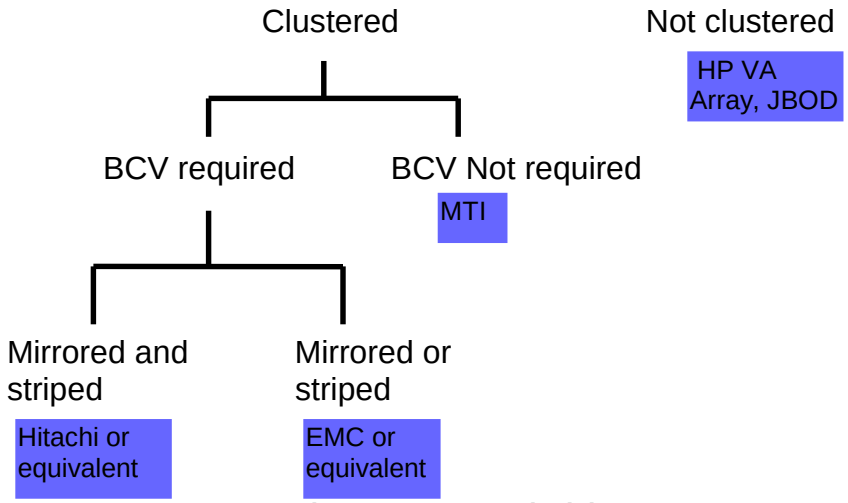


Figure 3: Storage decision tree

3.) Estimate CPU requirements. From the chart, you will have to choose your estimated CPU requirements. First decide on the number of users (from across the bottom of the chart). Then decide on your system load (from the side of the chart). The intersection of these two numbers represents the required SPEC rating. Finally, from your manufactures Configuration Guide or from the SPEC website, choose the appropriate server size.

	SPEC cfp_rate 2000 Rates		
Heavy	15.0	40.0	200.0
Medium	10.0	30.0	110.0
Light	4.0	20.0	55.0
# of users	1-50	51-500	501-1000

System Load

Figure 4: System load v.s. number of concurrent users

Note: Standard Performance Evaluation Corporation (SPEC) is an independent testing organization whose mission is : *"To establish, maintain and endorse a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems."*

Although spec was used in this example, any benchmark metric can be used including but not limited to, TPC, ICOMP or your own.

Load Definitions

Light - Commercial software involving minimal file manipulation (e.g. Netscape Certificate server, DNS, SNA, Proxy server)

Medium- Software development, database with light data entry, heavy file manipulation (http server, HP Developers Toolkit for servers, Micro Focus Cobol)

Heavy - Any database activity involving heavy data entry (e.g. Oracle, Sybase, Informix)

4.) Decide on the type of failover

You will have to decide from the requirements given as to whether you will need automated failover, manual failover, or none at all. Figure 5.0 should help with that decision.

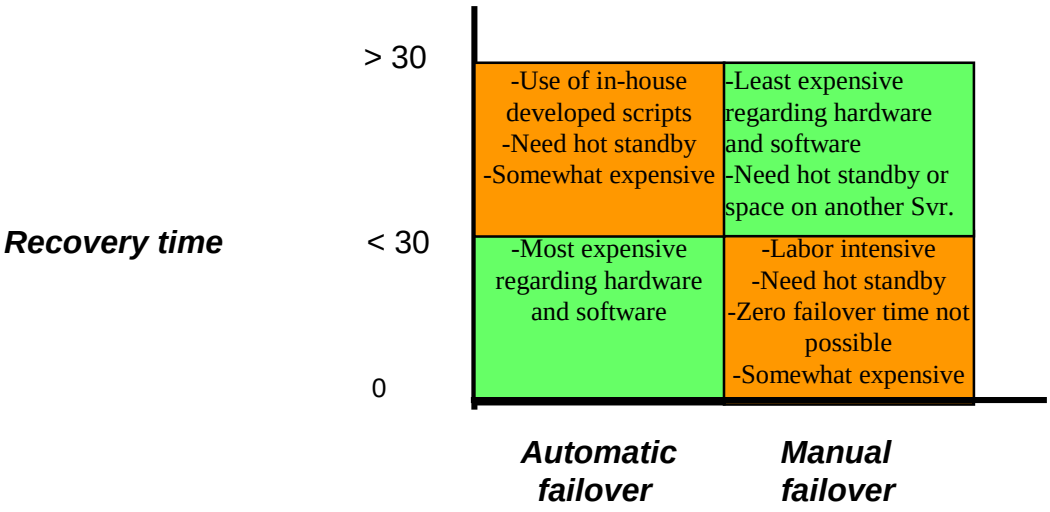


Figure 5: Failover decision matrix

Pulling it together - Summary

After items 1-4 are calculated, equipment needs to be specified and priced out. This methodology does not favor vendors. The selection of the server and disk manufacturer are a decision that more or less are already dictated by your environment. Generally, but not always, if you have HP servers, then more than likely you will specify HP equipment. Either way, you will have the information needed to choose the server size and disk quantity as you determine the manufacturer.

Output

The output of the SBE methodology is very important. It gives the Systems Engineer a paper trail showing that; information was collected from all groups involved in the project and a design was applied to those requirements. The inputs from each functional area (Business, Applications, Database, etc) should be printed along with an output from the collection tool if used and output from "Pulling it all together". Also, any Engineering calculations (I/O's per second, etc) and a copy of an Engineering drawings should be included.

The example in Appendix 'A' shows a complete case from the input phase through the design phase including all outputs.

Conclusions

The goal of the Solution Based Engineering methodology has been to produce a repeatable, disciplined, rules based, quality enforced method for Engineering client-server solutions. The task of bringing together a myriad of functional areas and collect requirements is a daunting one. Templates should be used to collect the data in a repeatable fashion and shared with all the areas involved.

Once all the data has been collected, a server Engineer can begin the design process. The templates will help guide the Engineer in server sizing, including areas such as memory, disk and CPU requirements. Most System Engineers design by an intuitive nature. They use:

- Intuition

- Pattern matching
- Experience
- Rules of thumb
- Ad hoc reasoning

In closing, all of the information that will be collected in these templates is available from someone. The key to completing a server sizing exercise starts with understanding the requirements. The Solutions Based Engineering Methodology is an attempt to bring together all the areas involved in a project, communicate their requirements and what it is they are trying to accomplish. With the cost of hardware sky-rocketing, does it make sense to stand in front of a CIO and explain how you basically guessed at a design or how you used a consistent methodology to Engineer your design?

References

Bachmann, Bass, Chastek, Donohoe, Peruzzi, **"*The Architecture Based Design Methodology*"**
Carnegie Mellon University/Software Engineering Institute, 2000

"*Capability Maturity Model Integration (CMMI)*"
Carnegie Mellon University/Software Engineering Institute, 2000

"*Guide to configuring hardware, software and support*"
Hewlett-Packard Company, 2000

Marty Poniatowski, **"*HP-UX 10.x System Administration*"**
Hewlett-Packard Professional Books, 1996

Frank Waters, **"*AIX Performance Tuning*"**
Prentice Hall PTR, 1995

Standard Performance Evaluation Corp. (SPEC)
www.spec.org

Appendix A

The problem

Note: This is a new design. There is nothing like this in company ABC's environment today, so it will not be a duplicate system.

Company ABC is about to launch a new sales initiative. A new application that is windows based will run on an NT server, as well as a database that will run on a backend Unix server. The NT server is out of scope for this exercise. The following are the completed templates used during the engineering process. The responses are in italic:

Business Template:

Current date: *September 01, 2001*

Project name: *New sales initiative for the Chicago region*

Project number: *123*

Project description: *This initiative will include opening a new office in the Chicago area and staffing it with 3 sales associates. A new application will need to be developed to help sell the new product.*

Business Unit: *National sales*

Business sponsor: *Joe Salesman*

Implementation date: *January 2002*

Support criticality: *Low, this system can withstand going down during selling hours. The system will be used from 8:00 a.m.- 5:00 p.m. Central time.*

Support hours: *7:00 a.m. - 6:00 p.m. Central time. The extra hour on each end will allow for an associate coming in early or leaving late.*

Does this application require failover: *No, this program can withstand to be down 1 day.*

Location(s): *Chicago office*

Number of users: *3-5 to start. One associate will be added per month for the first year. There could be a total of 20 users the first year and a max of 40 by the end of the second.*

Database Administration Template:

Database type (Oracle, SQL)? *Oracle*

Number of concurrent users? *10 (out of the 40 potential)*

Number of total users? *42 (40 users + 2 Admin users)*

Memory required per user? *5 MB per user*

Memory required for the database (SGA)? *1 GB*

Will it run on a raw volume or on a filesystem? *Raw*

Disk space required for database application (Oracle, SQL)? *100 MB*

Disk space required for database data? 1*0 GB*

Additional disk space for logs, imports, exports, etc? 2 *GB*

Number of database I/O's? *20 IO's/sec (based on experience)*

Application

Is this a custom application? *No*
If yes, what kind (VB, C, etc)?
How much memory is required for the application? *N/A*
How much disk is required for the application? *N/A*
How much data will the application generate: *N/A*
How much additional disk is required for the application (logs, etc)? *N/A*
How many users will use the application concurrently? *N/A*
How many total users of the application? *N/A*
Is this a CPU, memory or disk intensive application? *N/A*

If yes, which and how much? *N/A*
Will there be any daemons or scripts running? *N/A*
If yes, how much memory or CPU will they use? *N/A*

Operations

What kind of backups will you do (full, incremental)? *Full*

What day/time will you do the backup? *Sun-Sat / 11:00 p.m.*

What is the time estimate to backup this server? *1 hr*

How many tapes will you use? *(1) 4 mm*

Will you backup the OS? *No*

Date this server will go into production? *01/13/02*

Are there reports that need to be run? *No*

Are there any data transfers that need to occur? *No*

Who do I contact in case of an abbend or failure?

- *Unix Support group*
- *Database Support group*
- *Network Support group*

Helpdesk

What days/hours will you support this server/application?

- *M-F / 07:00 a.m.-6:00 p.m.*

How many people are trained to support this application? *3*

Is after-hours support required? *No*

What metrics will you collect/report on this application?

- *Avg. problem resolution time*
- *Calls per day*
- *Types of problems*
- *Dates/times of calls*
- *Contact*

Is there an SLA for this application that you must meet? *No*

Network

How many network connections will be provided to this server? *1*

What type (10Mbps/100Mbps Ethernet, FDDI, Token Ring)?

- *100Mbps Ethernet*

What type of WAN connection is required for this project? *Frame Relay*

What is the connection speed (56k, etc)? *56k*

Will there be redundancy in the WAN connection? *No*

Will you be able to fail it over automatically to another connection? *N/A*

Engineering

Will there be other applications running on the server (C, JFS, DCE)? *Yes*

If yes, what kind?

- *C compiler*
- *JFS*
- *OmniBack*
- *Glance*
- *Measureware*

How much memory, disk or CPU will they use?

Application	***Disk***	***Memory***
C complier	*68 MB*	*16 MB*
Online JFS	*600 KB*	*300 KB*
OmniBack	*5 MB + data*	*5 MB*
Glance	*5 MB*	*5 MB*
Measureware	*5 MB + data*	*5 MB*

Total number of Admin users on the system (not including DB users)? *2*

What kind of storage will you use (DAS, NAS, SAN's)? *DAS*

What kind of connection to the storage will you use (Fibre, SCSI)? Fibre

How many connections to the storage will there be? *2*
If failover is required, what kind will you use (manual, automatic)? *N/A*

Will you cluster this server? *No*

Server usage (light, medium, heavy)? *Light*

Pulling it all together

Memory

Application	Memory requirements
OS	256 MB
OS Applications	32 MB
Database	1.210 GB (1GB SGA + 42 cc users)
Scripts	500 KB
Total Memory	1.498 GB

Storage

Application	Storage requirements
Database App	100 MB
Database data	10 GB
Database logs, etc	2 GB
SWAP (2 x memory)	3 GB
Total Storage	15.1 GB

CPU Utilization

From chart 4.0 it was determined that a server with a Spec cfp_rate2000 rating of 4.0 is required. Using 1-50 users and a light system load as a variable.

Failover
No failover is required

Recommendations

HW recommendation

Since ABC company is an HP shop, based on the requirements given, an HP server would be recommended with the following configuration:

HP 'L-class' server
1 CPU @ 550 Mhz
2.0 GB memory
4 Fibre connections to the storage (2 to data, 2 to OS)
2 x 18 GB 15K rpm Fibre channel disk drive for data
1 x 18 GB 15K rpm Fibre channel disk drive for logs
1 x 18 GB 15K rpm Fibre channel disk drive for OS, OS Apps, SWAP, etc
1 HP Surestore VA storage array (7100)
1x100Mbps Ethernet NIC card
1 4mm DAT tape drive

Software recommendation

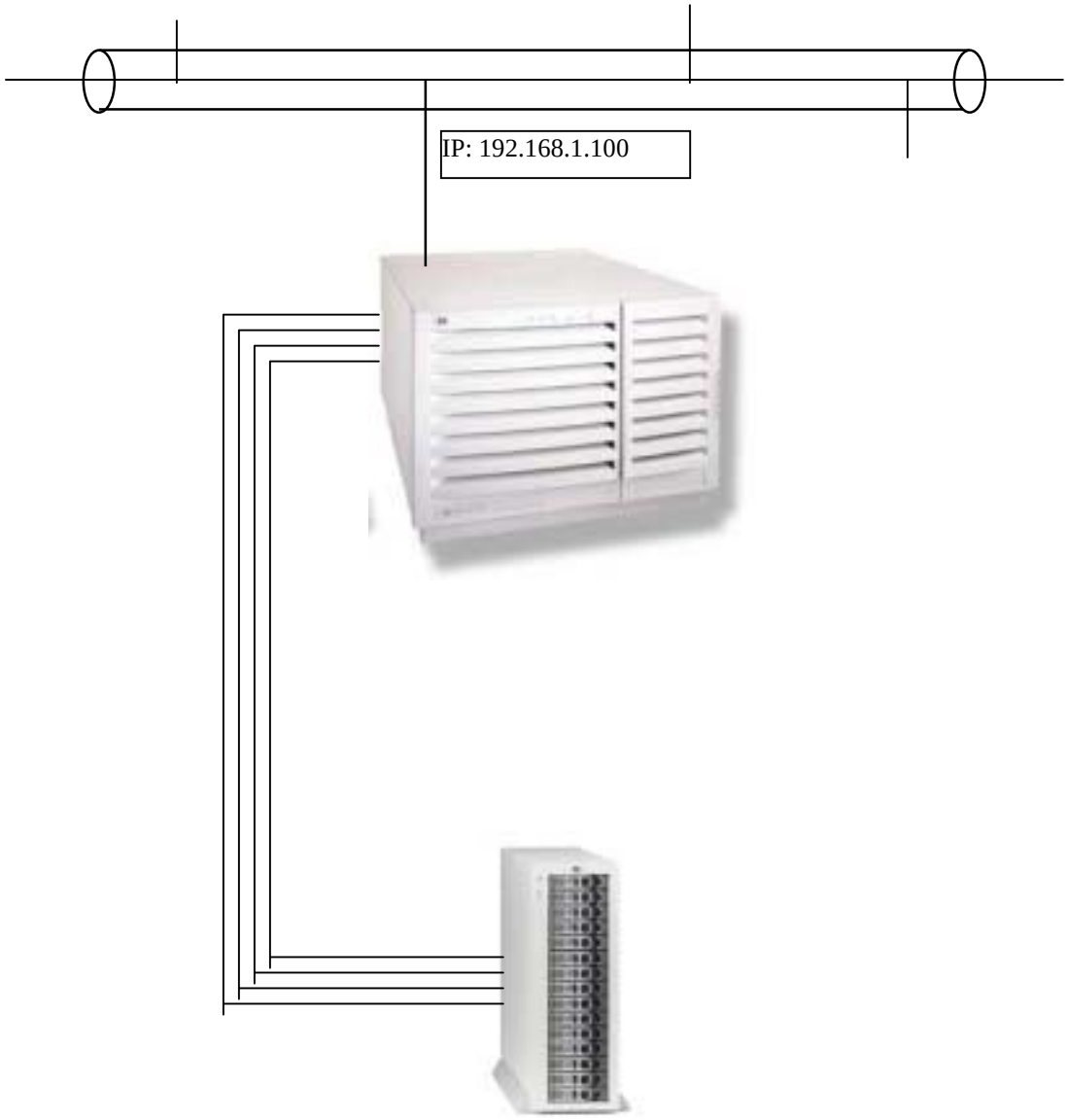
The software versions should be consistent with others in use. Licensing for products like OmniBack need to be closely scrutinized as to not pay for more than what is needed.

HP-UX 11I operating system
OmniBack backup software
Measureware
Glance Plus
Online JFS
C compiler

Support recommendation

Since the requirements for this server do not include any failover or critical production requirements, I would recommend basic 7x24 support. Again, the options you choose need to be consistent with what is in the environment today.

System Design Drawing



Local I/O path – Storage Blueprint