An Oracle White Paper
July, 2012

What’s Your Complexity Score?

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Executive Summary

Your large IT system will fail if you don’t manage its complexity[1]. The more complex a system becomes, the greater the chance of it failing. It is hard to specify the requirements for complex systems. It is hard to design complex systems. It is hard to implement and operate complex systems. Danger of failure becomes more probable as complexity increases. You should be aware of this issue and how it affects your costs, your staffing and your success. So are there better ways to mitigate this risk by reducing the intricacies of your IT environment?

Introduction

In 48 B.C., Julius Caesar wrote, “Fortune… can bring about great changes in a situation through very slight forces.” Little did Julius know that twenty centuries hence his words could refer to your enterprise system; great changes in your enterprise system from the slightest of adjustments are indeed afoot.

Your enterprise consists of large-scale servers, storage systems, networking equipment and the accompanying software that you use to run your business. The intricacies of your systems can be daunting. They may include commodity hardware, custom chips, embedded software, a myriad of device drivers, numerous operating systems, third-party hardware and software components just to name a few.

It was easy in the Eighties

Configuration complexity has been a growing issue since the 1980s. Before that, most elements of a system’s configuration – the hardware, operating system, database and storage, and applications – were simple and easily tied together into one system. Times have changed and now the typical configuration consists of several disparate piece-parts systems, each with their own level of specialization and sophistication. For example, a complex high-end server solution has separate storage area networks, networking equipment, and software stacks for the middleware, database, portals, application servers and web servers. Managing this explosion in diversity has become a costly nightmare

Software too has become much more sophisticated in an effort to meet consumer demand for ease of use and specialization. To make software easier to use, the complexity must be hidden in the software itself to shield the user. Solutions to today's complex requirements tend to emphasize hardware and software equally, rather than one or the other as in the 1980’s.

“So, how much is it REALLY going to cost?”

…is the first question you may ask when you begin considering a new software purchase, especially when you are planning a major Financial/Enterprise Resource Planning (ERP), Human Resources/Payroll, Customer Relationship Management (CRM) or other enterprise software selection and implementation. Implementing and deploying an enterprise business solution is not like buying a tangible product such as office equipment for you will need to budget for software licenses, database licenses, hardware environments - development, testing, quality assurance, training, production, disaster recovery - implementation, training and internal costs.

But you need to consider more than purchase price. You need to understand the halo of other expenses surrounding your decision. Internal costs include number of hours taken away from normal staff activities, duplicate work effort (entering information into multiple systems for testing purposes is one example), time required for designing, evaluating, creating the RFP, time for pilot testing, time focused on the project, 80% for the first few months and then perhaps 50% until completion before your staff can get back to their regular work, temporary work spaces for meetings, demonstrations, pilots or for the implementation team, travel expenses for remote workers, partners and contractors, legal work for contracts, overtime or bonuses for a job well done. For your hardware, your IT staff will evaluate, assemble and implement, hopefully on time. After the implementation and you are operational you must maintain your enterprise system, sometimes around-the-clock. While you know what tasks you need to perform, there are those tasks that you need to react to when there are operational issues. This approach is not cost effective. Many, too many, factors drive cost. And there are hidden or surprise costs that you cannot anticipate. The more complex the environment, the less predictable it becomes.

The following diagram, derived from an actual implementation, shows the workflow that you follow for a typical deployment:

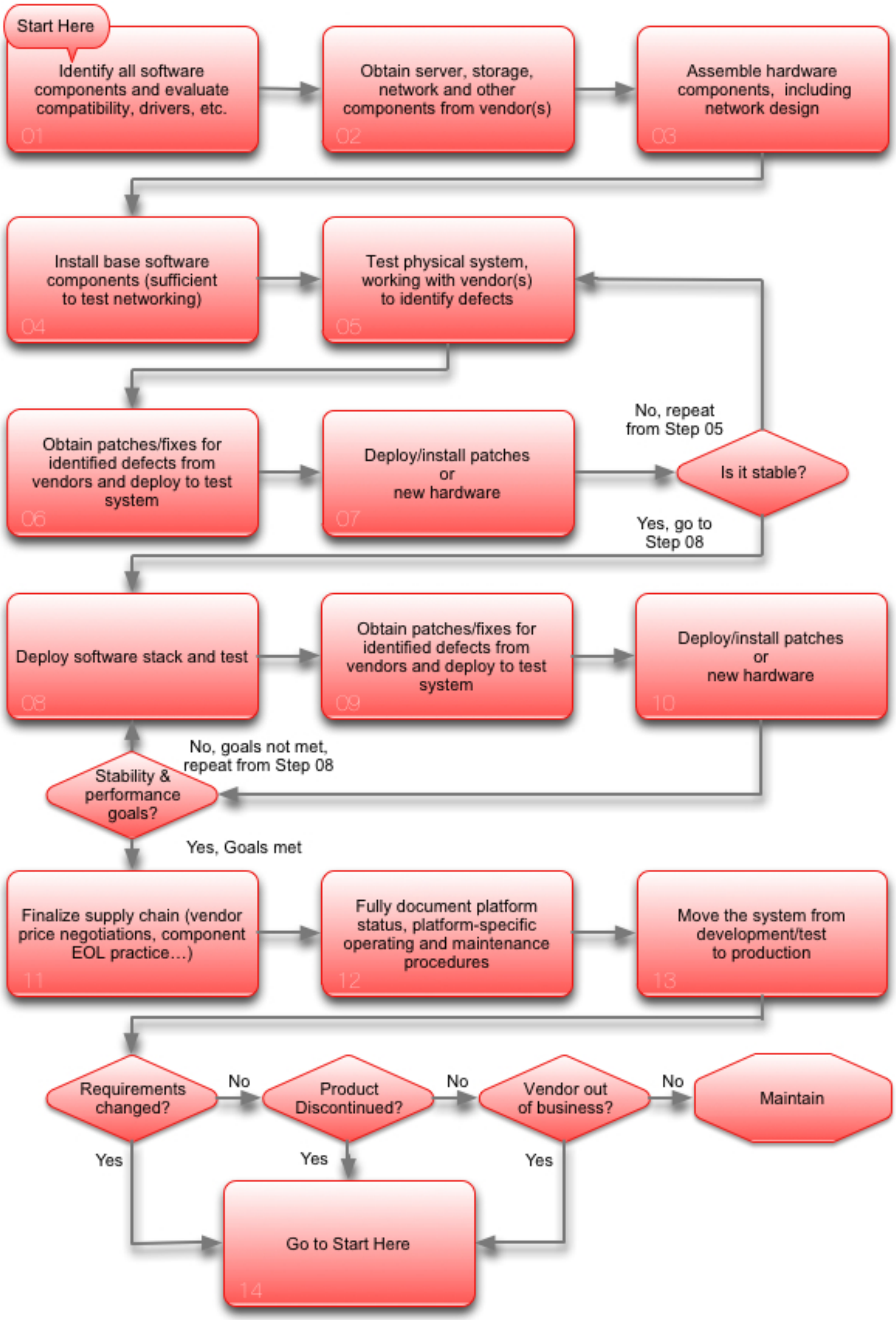
Start Here
Identify all software components and evaluate compatibility, drivers, etc.
01
Obtain server, storage, network and other components from vendor(s)
02
Assemble hardware components, including network design
03
Install base software components (sufficient to test networking)
04
Test physical system, working with vendor(s) to identify defects
05
Obtain patches/fixes for identified defects from vendors and deploy to test system
06
Deploy/install patches or new hardware
07
No, repeat from Step 05
Is it stable?
Yes, go to Step 08
Deploy software stack and test
08
Obtain patches/fixes for identified defects from vendors and deploy to test system
09
Deploy/install patches or new hardware
10
No, goals not met, repeat from Step 08
Stability & performance goals?
Yes, Goals met
Finalize supply chain (vendor price negotiations, component EOL practice…)
11
Fully document platform status, platform-specific operating and maintenance procedures
12
Move the system from development/test to production
13
Requirements changed?
No
Product Discontinued?
No
Vendor out of business?
No
Maintain
Yes
Yes
Yes
Go to Start Here
14

Figure 1 – List of tasks in a workflow for a typical deployment.

“Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clarke

There is a better way.

What’s Your Complexity Score?

In order to understand the complexity of any environment, it is helpful to quantify it. Let C stand for complexity, F, functional units and let D represent dependencies between those functional units. An example of a functional unit can be a component of software, an operating system, a hardware driver, a patch, a network board, a storage unit and so on or all of the above. Dependencies describe what functional unit depends upon others. The model below provides a means to measure and understand complexity independent of the underlying type of the functional units, an ability to compare hardware and software complexity at the same time.

If you assume that a 25% increase in functionality leads to twice as much complexity[2] then you can create a simple algorithm of the form:

$$C = F^3 + D^3$$

If you have one functional unit, F, and therefore no dependencies then the simplest level of Complexity is C = 1. As our functional units increase, say, a database (1), a middle tier (2), the application tier (3) and they all run on one piece of hardware (4), then you have four functional units and four dependencies just to keep it simple. This situation may be compared to adding a die to a set of dice. The number of outcomes is 6^n where n is the number of dice.

In our example, complexity can be calculated as $4^3 + 4^3 = 64 + 64$ or 128. As the number of functional units and the dependencies between them increase, you tend towards more and more complexity. While the quantification of functional units and the dependencies between them are subjective in nature, you can see how quickly complexity increases as shown in Figure 2:

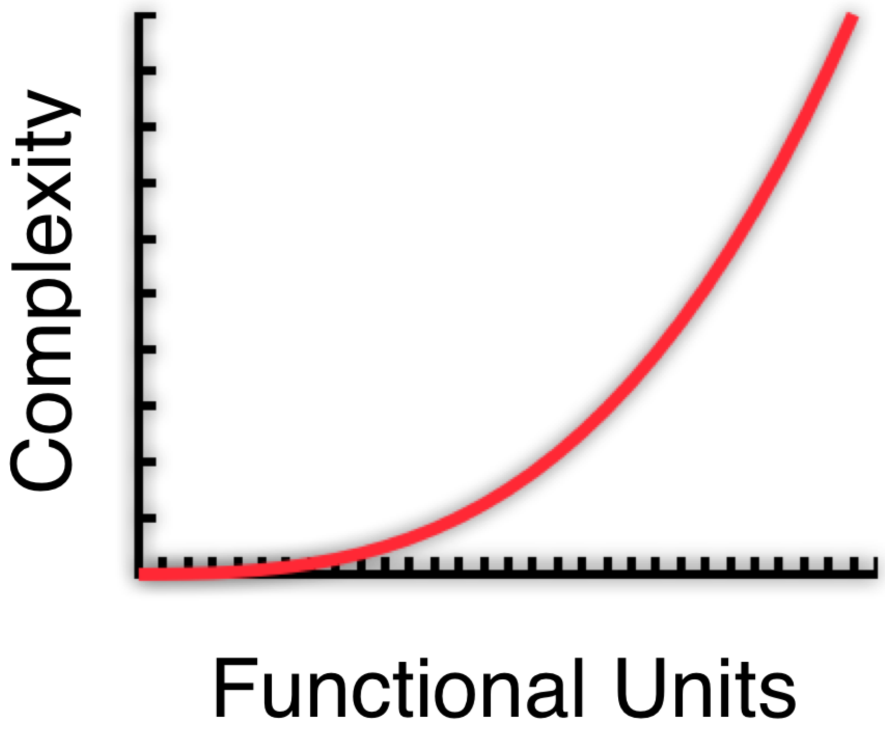

Figure 2. Functional units and their corresponding Complexity score.

The data in Figure 2 ranges from 0 to 300 for Functional Units and 1 to 27 million for Complexity. Note how fast complexity or costs increase with moderate increases in functional units. Increase or decrease your complexity and you will increase or decrease your cost tremendously. But there is another term to add to the equation, the unknown factor, X, commonly known as chaos:

$$C = F^3 + D^3 + X$$

Oracle minimizes the X Factor. The complexity in the system is still present but by factory integrating, testing and supporting it as a single stack, Oracle minimizes the X Factor. The complexity gets compressed and hidden, and its impact is minimized versus a conventional system. Oracle helps decrease complexity which reduces your costs, moderates your risk, and provides more predictable outcomes. Complexity leads to cost. However with a vertical integration strategy, your costs can be reduced and you become exposed to a variety of other benefits. Oracle submits that this complexity may be reduced through the acquisition of an Engineered System, Exalogic, Exadata and/or Exalytics, and the support surrounding it. Not vendor lock-in but vendor buy-in. Wouldn't it be nice if you could ease your complexity and make your environment simpler? Look at the picture below and imagine slicing your implementation costs by 20%? 30%? A half?

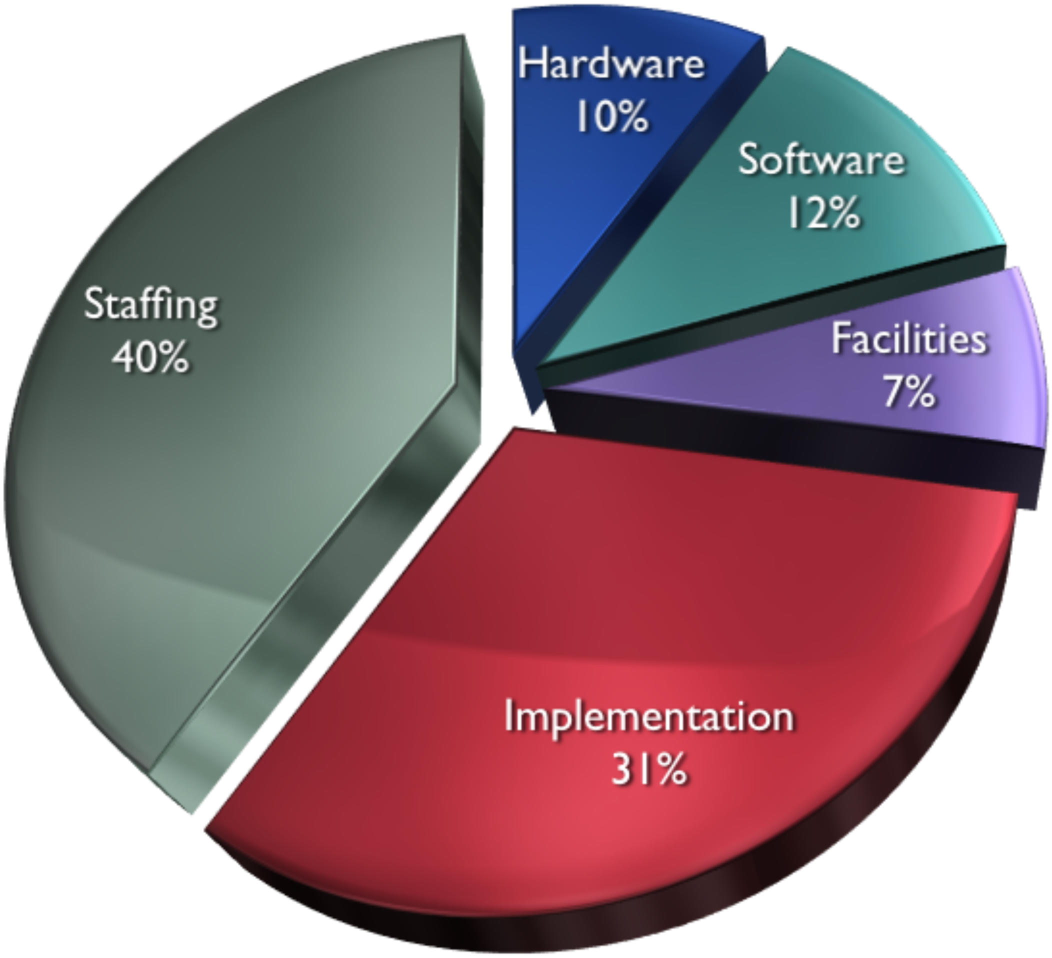

Figure 3. Enterprise IT budget expense breakdown, from Gartner, Credit Suisse[3]

Gartner[3] estimates that staffing and implementation account for 71% of a system's total cost of ownership over five years. How can you slice your implementation costs and then make more efficient use of staffing? We take care of several functional units so you are free to focus on your mission. Remember a decrease of 25% of the number of your functional units will reduce your complexity by half. Engineered Systems and the integrated Oracle Enterprise Manager capabilities have been designed to significantly reduce these costs. Together, these benefits come together to create a game changing technology for your system. Here are some of the tasks we take care of for you.

With Oracle's Engineered Systems you can eliminate Sizing and Deployment Planning, Installation and Configuration, Deploying and Scaling, Patching and Maintenance and Platform Administration and Figure 2 may now be represented by Figure 4:

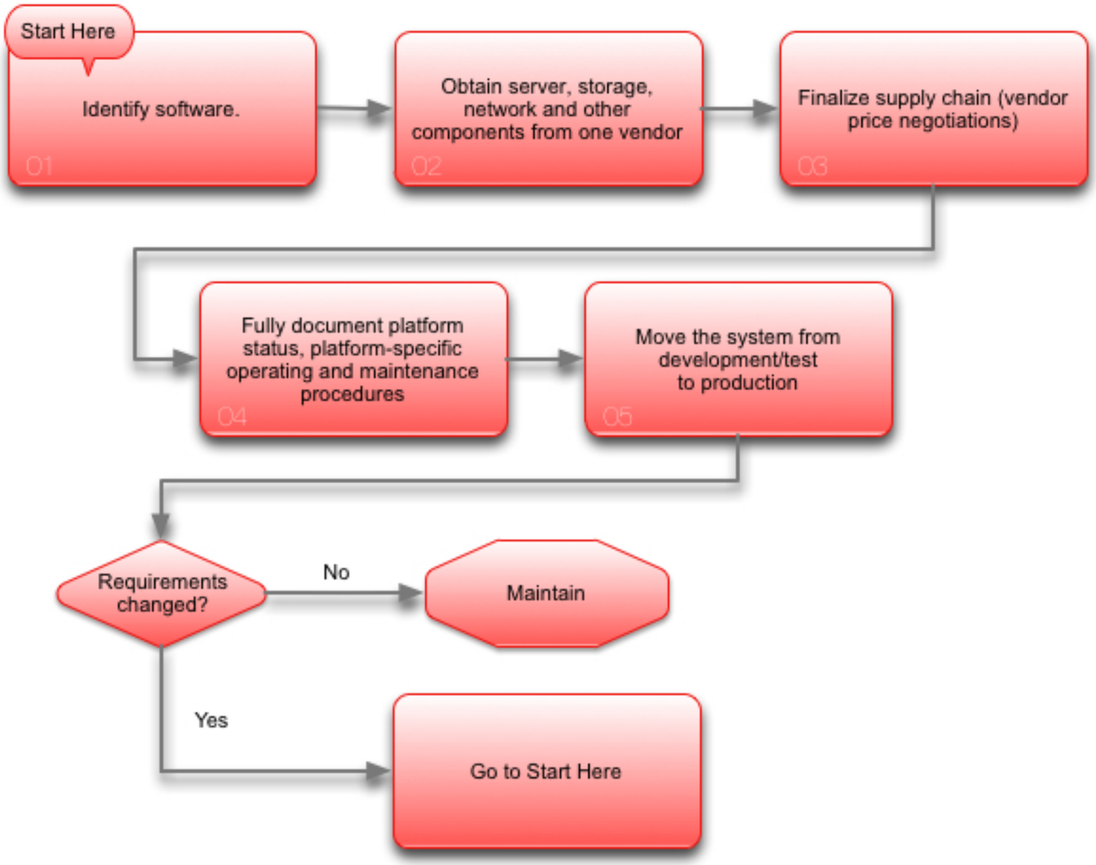
Start Here
Identify software.
01
Obtain server, storage, network and other components from one vendor
02
Finalize supply chain (vendor price negotiations)
03
Fully document platform status, platform-specific operating and maintenance procedures
04
Move the system from development/test to production
05
Requirements changed?
No
Maintain
Yes
Go to Start Here

Figure 4. With an Engineered System from Oracle, a smaller list of tasks results for the workflow required for a typical deployment.

You can reduce your ownership costs dramatically by leveraging an Engineered System's pre-integrated architecture, end-to-end vertically integrated monitoring via Oracle Enterprise Manager and because software runs more efficiently on Engineered Systems, you can eliminate 2/3 of the CPU cores, as compared to traditional systems.

In addition to fantastic performance which implies there is less hardware to purchase, an Engineered System allows your deployments to get up and running faster than traditional systems. An Engineered System contains everything needed for a particular new environment whether a database or an application. An Engineered System contains networking switches, storage, flash, RAM, and compute nodes, which are all pre-integrated and optimized to work together. Exabus (InfiniBand networking technology, related protocols, and Java APIs), connect all components within an Engineered System, between different racks of the system, and between other Engineered Systems from Oracle. Furthermore, the software on top of the hardware has been vertically optimized to work with other software as well as with the hardware inside of the Engineered System on which it is running. New environments can be up and running and ready for application installation in as little as a day as opposed to the months it traditionally takes.

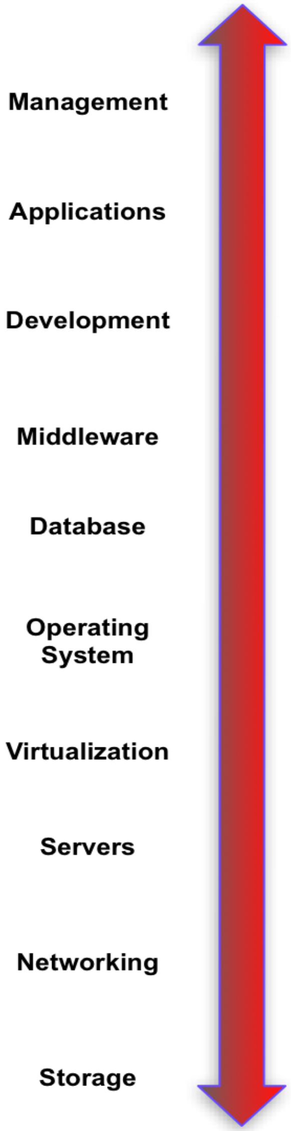

Figure 5. Oracle’s vertical integration strategy across all the tiers in an enterprise.

How does an Engineered System reduce the total cost of ownership? All patches and firmware updates below the WebLogic or Middleware tier in Figure 5 are provided in one certified update, similar to how hardware and software updates are all bundled in single iPhone updates. Certified means we have already tried and tested this patch so that you don’t need to spend time on it. Oracle can do this because of knowledge of each system’s configuration. With a finite and well-known number of system configurations, it becomes easy for Oracle to release consolidated patches. Finally, embedded hardware diagnostic capabilities allow for Engineered Systems to “phone home” to file Oracle Service Requests in the case of hardware failures. The integrated nature of the Engineered Systems, the quality of the products on their own, and the value of the integrations between these best-of-breed products inside an Engineered System allows for unparalleled management, monitoring and ease of maintenance.

Stories from the Field

A customer experience a data and service loss due to a controller card firmware update failure which resulted in corrupt data blocks getting copied. It took four days and 1000 person hours to restore the database to a sane point and with some acceptable data loss. This issue wouldn't have happened if they had used Oracle’s practices which are built into Engineered Systems.

Another rather large and prominent customer experienced an 11g subversion update failure but the real issue was this: the customer's custom script was untested and obliterated the parameter file because there were no controls on changes to the parameter file which was in PRODUCTION! The issue was resolved in TEN DAYS with over 400 person hours expended. This would not have happened in an Engineered Systems environment as parameter files are pre-tested and pre-tuned and with no need to change.

Internally, Oracle uses Engineered Systems for performance testing, quality assurance testing, and when hardware is required. Oracle also uses Engineered Systems as the foundation for its Public Cloud. Why? Less hardware is needed and Engineered Systems are easy to set up and perform exceptionally well and they provide an optimal environment for Oracle applications. Oracle leverages the benefits of Engineered Systems: it reduces complexity and costs less to operate.

Consider these results:

- On average, more than 70% of IT budgets are spent on operations and maintenance[4]

- 55% of IT professionals experience downtime, from minutes to over a week while performing an infrastructure upgrade[5]

- It can take up to 4 to 6 months just to establish hardware and software infrastructure[5]

- Nearly two-thirds of organizations fall behind schedule when deploying new IT capabilities[6]

Oracle developed Engineered Systems to address these issues, to decrease your risk, to make your environment more predictable, to reduce complexity and to increase the productivity of your team.

A Sample Complexity Calculation

"Entities must not be multiplied beyond necessity."

Occam's Razor

Let's do a comparison of the traditional stack versus the Engineered Systems stack and calculate the complexity score for each.

	Generic x86 Server	
Component	**Description**	**Quantity**
Chassis	Chassis + redundant power supplies	1
CPUs	Six-core Intel X5675 (3.06GHz)	2
Memory	8GB LV Memory DIMMs	12
Ethernet	10GbE dual-port adapter	2
SAN connectivity	8Gb dual-port Fiber Channel HBA	2
Local storage	300GB 10K RPM disks	4
Operating system		1
Misc	Server License	1
Server Hardware Component Total		**25**

Storage Hardware BOM	**Quantity**
Base with 128GB memory	3
Standby Power Supply	3
100GB Flash Drive, 4Gbps interface	72
15 Slot Disk Array Enclosure (DAE) - 600GB	5
2TB 7,200rpm SATA Drive, 4Gbps interface	315
15 Slot Disk Array Enclosure (DAE) - 2TB	21
8Gb 8MM FC PORTS	48
Storage Hardware Component Total	**467**

Software BOM	Quantity
Base Software BASE	1
Software for non-SATA Tier	1
Software for SATA Tier	1
Advanced Software Suite SATA 1TB	630
Storage Management for non-SATA Tier	1
Storage Management for SATA Tier	1
SAN Manager for non-SATA Tier	1
SAN Manager for SATA Tier	1
Performance Manager for non-SATA Tier	1
Performance Manager for SATA Tier	1
Software Component Total	**639**

Let’s use the grand total of 1131 for functional units and not include the dependencies. The complexity score is 1106^3 or 1,446,731,091 or almost one and a half billion!

An Engineered System that out-performs the above configuration has a complexity score much less than this. Oracle pre-assembles the hardware and tests the software at the factory so you don’t need to assemble the pieces, worry about versioning and testing. You can be confident that all of the pieces will work together right out of the box. This huge reduction in complexity allows you to focus on getting your application up and running faster and with better performance.

"First I have tried to eliminate everything unnecessary to conveying experience to the reader so that after he or she has read something it will become a part of his or her experience and seem actually to have happened. This is very hard to do and I've worked at it very hard."

Ernest Hemingway

Conclusion

You are more aware of the risks to system resilience due to the history of horizontal IT procurement and you can better account for the hard costs of IT layer integration and calculate your complexity. You understand that there is a major issue of time lost with IT professionals within your IT department attempting to triage issues. As this is typically not a core competency of your IT professionals, their efforts are typically ineffective (luck and super human efforts result in success rather than a rigorous engineering process). Your command and control requires a great deal of effort (head count) and investment (time and money) to maintain a horizontal IT environment.

Another major issue is the burgeoning number of retirements impacting the operational effectiveness of many organizations. With so many mid-level IT managers retiring, every organization is losing the ability to command and coordinate sophisticated IT platforms. Finally your agility is low for the relative cost of the IT investment.

Engineered Systems combined with the Oracle team provides you a safe, predictable, lower cost environment that will help you get up and running quickly, stay up and running and free you and your team to focus more on your mission and less on triage, expending wasted effort and confront a lack of qualified resources. We take care of 70% of the stack in Figure 5, so you are free to focus on more pressing challenges. This vertical integration strategy for pre-engineered systems is one of the reasons Engineered Systems provide tremendous value to your organization.

As Julius Caesar wrote, "Fortune… can bring about great changes in a situation through very slight forces", Oracle can help you change your system complexity with "slight forces" – Engineered Systems.

References

[1] http://www.cioinsight.com/c/a/IT-Management/Why-IT-Projects-Fail-762340

[2] James S. Collofello, Scott N. Woodfield, Norman E. Gibbs. Software productivity measurement. In *American Federation of Information Processing Societies: 1983 National Computer Conference, 16-19 May 1983, Anaheim, California, USA*. *AFIPS Conference Proceedings*, pages 757-762, AFIPS Press, 1983

[3] Philip Winslow, “Dr. Exalove: How I Learned to Stop Worrying (about Sun) and Love Exalogic Too,” Credit Suisse, Nov. 23 2010

[4]IDC, Analyst Matt Eastwood, IDC Directions Presentation, 2011

[5]From a commissioned study by Forrester Consulting on behalf of IBM

[6]IBM Market Insights Study – 2011 Business Benchmarking Time-to-Value Study

The Complexity analysis was derived from work by Roger Sessions, CTO, ObjectWatch, http://www.objectwatch.com

Oracle is committed to developing practices and products that help protect the environment

What's Your Complexity Score?
July 2012
Authors: Peter Doolan and William Indest
Contributing Authors: Tai Khan and Todd Lesser

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together