A CIO’S TOOLSET TO RETHINKING APPLICATION MANAGEMENT

5 Strategic Approaches to Effective Application Management Services
INTRODUCTION

In the last decade, CIO priorities have shifted considerably; from technological issues of scalability, security and accessibility, to business-driven issues of alignment with strategic contributions to the business. Decreasing enrollment in college and university IT programs coupled with the increasing pressures of baby boomers retiring is resulting in a growing skills shortage. Issues of globalization and the constant demand to maximize the return on investment equates to today’s CIO literally being asked to do more with less.

This paper illustrates five strategic approaches for IT executives and managers to address these challenges through more effective and productive enterprise Application Support (AS). Simply by extending certain best practices most organizations can double their productivity and decrease AS costs by up to 40%, while mitigating many of the risks normally associated with such initiatives.
APPLICATION DEVELOPMENT AND APPLICATION MANAGEMENT

Today’s CIO is facing ever increasing challenges. A study by Alinean Corporation estimates 60-90% of IT operating budgets is consumed by support and maintenance costs. Consider the difference in the roles of these two divergent groups:

Application Development (AD) is a visionary role within the organization, designing the world of tomorrow. AD teams thrive on the constantly changing world of projects and technology. Despite often daily heroics, Application Support (AS) teams are not showered in praise because often their efforts go unnoticed.

Implementing departments based on these two disciplines will help develop intellectual capital that is less susceptible to attrition. AD teams move from project to project and application to application often making it difficult for them to become subject matter experts. Separating AD and AS recognizes and celebrates the differences.

Five key strategies to help you achieve higher levels of productivity and decrease costs by up to 40%:

Strategy 1
Understand the similarities and differences between Application Support (AS) and Application Development (AD), and treat them as separate and distinct disciplines.

According to research from Alinean Corporation, support and maintenance can make up as much as 60-90% of IT operating budgets. Controlling these budgets and ensuring the maximum value is being extracted from each dollar is creating a shift away from the traditional thinking that the skills required for AS and AD are the same.

The primary role of AD in the organization is designing the world of tomorrow; its focus is therefore on technology and projects. The AD group - in addition to possessing technical skills - must also be visionaries and pioneers with a demonstrated sense of adventure. They thrive on constantly changing environments and expectations, moving from project to project and learning new technologies. The success of an AD group is measured on its ability to meet deadlines, budget, and deliver as defined in the scope.

Contrast this to the AS role, which deals with the here and now. An AS group is not designing tomorrow’s world it is ensuring today’s world is operating as efficiently and smoothly as possible. The skill set of AS focuses more on problem solving and trend analysis. Although, like AD the technical skills of AS are important too.

Often an AS teams deliverables are not clearly visible to the organization as they are not delivering big projects or large applications. However, they are keeping the business running and typically are not showered in praise. The source of pride within AS is therefore within the metrics that demonstrate their value to the organization. AS teams are measured on their time to respond and time to resolve problems or turn around requests. They are also measured on the volume of reactive vs. proactive work executed.

A Cabinet Maker Is A Carpenter, But A Carpenter Is Not Necessarily A Cabinet Maker
Both groups require a high degree of business acumen, but developers must have the ability to learn application business requirements quickly, this knowledge is often lost, forgotten or outdated as they move to new projects. Typically developers are not involved in an application for a great enough amount of time to be considered viable subject matter experts beyond a few months. AS teams not only learn the business function of an application, but must become intimately knowledgeable of the code, and take pride in the fact that they are subject matter experts.

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>Application Development Services (ADS)</th>
<th>Application Management Services (AMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role in organization</td>
<td>Strategic, forward thinking</td>
<td>Operational, current thinking</td>
</tr>
<tr>
<td>Natural way of thinking</td>
<td>Projects and technology</td>
<td>Assets (applications), users, projects and process</td>
</tr>
<tr>
<td>Source of pride within the organization</td>
<td>New technology to leverage the pride of the organization</td>
<td>Metrics showing continuous improvement</td>
</tr>
<tr>
<td>Inherent capability</td>
<td>Visionary</td>
<td>Problem solver</td>
</tr>
<tr>
<td>Methodology followed</td>
<td>Linear with defined beginning and end</td>
<td>Cyclical with no beginning and end</td>
</tr>
<tr>
<td>Personal Benchmarks</td>
<td>Project timelines and cost</td>
<td>Time to respond and time to resolve the incoming issues</td>
</tr>
</tbody>
</table>

Companies which recognize these differences differentiate themselves within the marketplace, highlighting to employees that they recognize and value each unique discipline. This can lead to higher employee satisfaction as their career path and specific knowledge provides unique challenges and opportunities within the organization.

**You Can’t Delegate Culture**

**Strategy 2**

*If you are going to go to the trouble of implementing continuous improvement, make sure you instill a deep seated culture around it.*

The concept of Continuous Improvement (CI) is as ubiquitous as it gets. Everyone talks about CI, most claim they’re doing it, and many actually are in one form or another. But are the anticipated benefits being realized? For those who answer no to this question, we provide this tip. Failure to do so can, and probably will, greatly reduce the benefits you’d expect to achieve. Remember too, that you can’t delegate culture. Culture must be cultivated at the top and nurtured throughout the organization.

Making CI the rallying cry of your AS teams not only focuses on ensuring your applications are more reliable, useful and low-cost; it actually provides a sense of value and purpose to the AS teams within the organization. In fact, providing reliable, useful, low-cost applications is the primary responsibility of an AS team. To accomplish this they must focus on driving down reactive work, which only squanders scarce financial and personnel resources, and re-focus effort on proactive work – in other words, work which provides value to the organization, be it in the form of CI or the fulfillment of application enhancement requests from the business.
When creating a culture around CI make sure you recognize the AS team’s value and necessity in the process. Ensure all levels of management are educated of the role the AS team play and what each party brings to the table. Meaningful metrics should then be developed, objectives set, performance measured, and success rewarded. You should find that aside from the increased productivity and cost savings this process realizes, there is a byproduct of better rapport throughout the business which ultimately provides for better business alignment.

Are you reaping the benefits of your Continuous Improvement initiatives?

Make CI the rallying cry of your AS teams. The primary responsibility of AS teams is to provide reliable and useful low-cost applications. AD and AS are separate but tightly integrated disciplines. Set the dividing line between work flowing to AS and AD to the “installed base”. Other dividing lines are acceptable if they better meet your corporate culture but extra rigor around the interfaces between AD and AS will be necessary. You don’t have to choose a pure black box model, but the evidence to support it is compelling.

Strategy 3
Breakdown your IT applications into a series of black boxes.

Take a lesson from NASA

NASA engineers adopted the concept of black boxes in order to aid the design of complex machines such as the space shuttle. No one individual possessed the expertise required to design and build the space shuttle, so the concept of black boxes was used to simplify the design by breaking it down into smaller components. By thoroughly defining the intended functionality of each black box component and the exact input and output specifications, NASA was able to farm out the design and development to companies with expertise in each specific black box area. The black box delivery model can be seen below.
This is the beauty and simplicity of the black box and forms the basis for this tip. Utilizing the black box delivery model is an effective way of breaking down complex IT departments into manageable segments. This process also provides an opportunity to organize the segments by business unit, thereby improving alignment. These smaller black boxes can then be farmed out, whether to internal or external teams either onshore, nearshore or offshore as appropriate. The importance to the organization of black boxing applications is to ensure only one team of programmers is responsible for the application, only they have access to the code. This is crucial to the AS team charged with making the application more reliable, useful, and low-cost. As a NASA vendor who designed a propulsion system to exact specifications, would you warrant the work if you knew NASA’s own development team modified the design without your knowledge or consent?

The level of cost savings and productivity increases are highly dependent on how close to the black box model you implement and the interfaces developed between AD and AS. Black boxing can lower staff turnover rates by decreasing levels of both burn-out and rust-out. AD and AS are separate but tightly integrated disciplines. The current ratio of AD to AS methodologies is within the 200:1 range.

The AS life cycle is cyclic, repeating itself year after year throughout an applications service term. Applications are corporate assets and like all assets must be managed through a cyclical life cycle. The cyclical life cycle drives not only applications through the continuous improvement process year after year but also processes, methodologies, frameworks, standards and even governance.

To harness the maximum return on investment a cyclical life cycle should be adopted. You wouldn’t build a house without blueprints so why build an AS team without them? Regular assessments help to ensure AS teams are adopting and complying with pre-determined best practices. It is important to leave room for growth within assessments to allow for cultural differences between teams. A little compromise can go a long way to achieving overall success.
Many organizations have taken the step to divide AD and AS into two distinct disciplines, yet struggle dividing the work flowing to these two groups. One method is to set a dollar figure, e.g. everything over $250,000 goes to the development team, while a second method is to use the hours involved, e.g. less than 50 hours of time goes to the support team. In the purest sense of the black box delivery model what these methods allow is two separate teams access to the same code; teams with disparate goals and objectives as well as distinctly different roles within the organization. These methods essentially allow the NASA developers to tweak the vendor’s original design.

However, the black box delivery model can be a powerful and effective technique. Although many companies choose not to implement a pure black box model, the argument for doing so can be quite compelling as evidenced by the diagram below charting cost and productivity indices through a black box implementation. The dramatic effects a pure black box model can have on both productivity and costs are exposed.

Note: Although this data demonstrates the extreme effects of black boxing, it is actual data based on empirical data from a client of Sierra Systems.

The tangible benefits of the black box model do not stop at cost and productivity benefits, black boxing provides flexibility in how you want these benefits to pay off; they can translate into cost savings, productivity gains or both. With more stable applications the AS staff may suffer from lower burnout rates, less stress, and ultimately lower staff turnover. Levels of rust-out among your support staff may also be reduced because now they are being tasked with enhancements and continuous improvement rather than responding to, and resolving, incidents.

The win for the development side of the business is just as strong. The role of developers may also be enhanced as they can be relieved of support rotation duties and allowed to focus on what they do best – delivering projects and utilizing newer technologies to help build the future of the organization.
Strategy 4

Your applications are corporate assets so manage them as such.

A search on Google for “Application Development Methodologies” returns over 11.7 million results while a search for “Application Support Methodologies” provides only 65,400 results. This speaks volumes on how misunderstood and undervalued AS really is. Most organizations still force AD methodologies upon their AS teams, once again failing to recognize the very distinct differences that set them apart.

AD methodologies are designed for work that will have a defined beginning and end. AS methodologies typically do not have a defined beginning or end, the life cycle is not linear as is the case with project-based methodologies. ITIL has recognized this with the development of their cyclical life cycle on which ITIL version3 is based. The author of this paper makes use of another cyclical lifecycle, depicted below, that is designed specifically for AS.

Why a cyclical life cycle? Your applications are corporate assets so the life cycle of them should treat them as such. Companies have been using cyclical asset management life cycles for decades. This is because year after year assets are driven through the life cycle which is designed to provide preventive maintenance, extending the life span of the asset and ultimately providing the maximum return on investment. The life cycle also provides for annual assessments of the asset against pre-defined criteria, these criteria may indicate when it is time to consider decommissioning the asset.

The AS life cycle should follow a similar process path. Each year work is carried out to ensure an application continues to run. Each year any problems encountered are trended and analyzed to see if support and maintenance activities are focused primarily on “workarounds” or resolving root issues. The life cycle looks not only at applications but at all processes, methodologies, frameworks, and standards associated with AS, even the governance structure should be reviewed to ensure all are effective and efficient. This method drives continuous improvement over years, continuously reviewing costs and ensuring applications are as reliable and useful as possible. Ensuring adequate and timely reporting demonstrates what support teams are accomplishing and where the bulk of their time is spent. This reporting can be delivered in monthly scorecards, semi-annual presentations and annual stewardship reports.

The cyclical life cycle essentially ties together all AS activities into a practical, pragmatic approach that ensures your applications are receiving the best of care from implementation to decommissioning.
Strategy 5
If you want to ensure consistency in your IT department, develop or adopt an assessment strategy.

If a builder is hired to construct a house but instead of blueprints being provided a description of the completed house is given by the owner, the completed house will likely not resemble the owners described vision; rather it will look exactly how the builder envisioned it. This highlights the value of blueprints, as they leave nothing to the imagination. The layout, dimensions, measurements, and materials are all specified in great detail. The builder has little margin for error unless they choose to deviate from the plans. For this reason we have inspectors who reference the blueprints throughout the process to ensure the house is being built to the specifications provided.

Assessments within IT departments can be used much like blueprints in the above example. Assessments allow communication to an AS team of what is wanted, how it should be carried out, and what tools will be used. Assessments should leave little to the imagination and provide little margin for error. Regular assessments can help ensure teams are adopting and complying with processes that are in place. However, unlike blueprints a degree of movement can be built into assessments that allow for a slight deviation to recognize differences between companies, divisions and teams. It is important that everyone satisfies the original intent, even with slight variations in how activities are carried out. Tampering with or ignoring cultural, environmental or political differences between teams can, and likely will, be seen as stripping the team of their identity which may act as a barrier to success.

When designing assessments, consider what the intent is and measure that rather than full blown compliance. By focusing on the original intent of an assessment buy-in may be easier to obtain and success may be more likely. Often incorporating compromise provides a better result than not, 80 to 90% success rates can be achieved rather than 10 to 20% with no compromise.
CONCLUSION

The consistent implementation of these and other strategies can be a very effective way to lower AS costs, improve productivity, and increase Business/IT alignment and rapport. These strategies are based on best practices that have been adopted by some of North America’s most successful companies including Canadian Pacific Railway, Toyota Canada, and TransCanada Pipelines. The strategies outlined here have been adopted on top of, or in conjunction with, ITIL and other industry frameworks and have proven their ability to stand alone without compromising their ability to deliver the intended results.

Sierra Systems has created tools to assess your application portfolio, resource skill gaps and measure AMS practice maturity level. Sierra’s SMART Methodology™ is a well-documented ITIL compliant framework consisting of 12 best practice how-to’s for application maintenance. It allows organizations to coordinate, manage, and implement their chosen AMS transformation. Sierra has also authored the three book series Maximizing IT Value, which describe how to improve AS performance.

For more information on best practices in supporting, maintaining, and enhancing enterprise applications or more details on the topics discussed here please contact us:

contact@sierrasystems.com
1-877-688-1371
sierrasystems.com

References

[Carr] Nicholas G. Carr, IT Doesn’t Matter Harvard Business Review
[Sierra] Canadian Pacific Railway Case Study, 2004
[Thompson] Dr. Peter Thompson, Maximizing IT Value through Operational Excellence Trafford Publishing, 2006