Shedding Light on Shadow IT

Whitepaper

Microsoft Enterprise Architecture Roundtable

Ramit Luthra, Karun Pothacamury, Donald Stahl, Tom Valva

September 2014

Table of Contents

Introduction

As an IT professional, perhaps you've observed one or all of the following phenomena.

> *A small group of people in a department have created a system that the business users are very impressed with, even ecstatic over. These people are not from the IT department, but they've created something using desktop technologies that successfully fulfill a critical business need.*
>
> *The Sales department has contracted with an outside service provider to create a mobile application to connect to APIs provided by Salesforce.com; no one from IT was invited to the planning meetings.*
>
> *An entire division of a firm has what can only be described as its own independent IT department. This division uses services provided by "corporate" IT, such as email and hosting, but they maintain their own application servers, and they have their own programmers who use completely different languages and tools than the sanctioned "corporate" software stack.*

In a world of monolithic corporate IT departments, these individual efforts are often referred to as "Shadow IT". They operate outside of the purview of the traditional, centrally controlled IT department, and often have their own infrastructure, tools, vendors, and programmers. The solutions these groups provide can vary in sophistication from simple to highly complex, and the business problems they solve can be anything from producing mailing labels to performing complex calculations in specific domains such as finance, engineering, operations management, etc.

One of the first reactions of a professional firmly ensconced in the corporate command and control IT environment so common today might be to scoff at these efforts. Indeed they can often seem small and insignificant compared to the vast server and network infrastructures and millions of lines of code often managed by corporate IT. Another reaction might be to feel uneasy about them. Why do they exist, and at what scale? What services are they providing that corporate IT isn't, or perhaps should? Are they secure? What risks do they present? How are they budgeted? Are they redundant?

Clearly the phenomena of Shadow IT, once confronted, raises some difficult questions. Indeed we may find that it can tell us things about our use of technology that we've been ignoring, perhaps to our disadvantage. But before we consider such concerns, a definition is in order.

> *We can define Shadow IT as those technology related activities undertaken by groups in an enterprise where such activities are not controlled and administered by the centralized IT function. The centralized IT function believes that these activities are part of the IT organizations purview, and would normally be under its control. The individual IT departments in a decentralized structure are not Shadow IT groups. If there is no formal, centralized IT organization, there cannot be Shadow IT. Furthermore, the Shadow IT groups themselves desire to operate independently of the centralized IT function for various reasons including, but not limited to a desire for*

independence, enhanced productivity, specialized domain knowledge, control over the development life-cycle, competitive drivers, and budget autonomy.

Put more simply, Shadow IT can be defined as "doing what IT can't, or won't do". Enterprises should be concerned about Shadow IT because it represents a gap in services not provided by the centralized IT function. Additionally, risks associated with data loss, and costs arising from duplication and fragmentation of services must be addressed. However, the very existence of Shadow IT often points to tangible business needs, and in many cases represents an opportunity to affect significant positive outcomes in an enterprise.

In this paper, we will look at the reasons these shadow efforts exist and persist, often despite efforts to fold them into the traditional centralized IT model. We will attempt to define the structures these efforts most often assume, and identify the risks and benefits associated with them. Ultimately we will offer recommendations for dealing with the Shadow IT phenomena, which may result in some surprising conclusions.

Section I - Models of Shadow IT in the Enterprise

Model 1: Practice Driven Development

This model is not typically what people refer to when they use the term Shadow IT. In this model, a group that is part of a distinct practice within the organization performs its own technology management partially or entirely independently of the central IT organization. These groups are intimately familiar with specific aspects of the business and this intimacy directly influences the technology solutions they produce and use. Yet no one considers these groups an IT function in the traditional sense. This subject matter expertise affords these groups an independent status, and CIO's often turn a blind eye unwilling to incur the risk of absorbing them. These practice-driven organizations generally come in one of three different forms:

- **Legacy** – this first type of practice driven independent IT effort in a business function precedes the formulation of a mature IT organization. The group uses its own resources and expertise to create and/or adopt technology as needed. These groups were never considered a technology function. They are distinct from the groups that traditionally used the first computer software, typically accounting and finance departments. Often these are delivery-critical operational groups (warehouse management, media productions, sales, etc. Typically, these groups deal with technologies and vendors of which the IT organization has no hands-on experience, or operational knowledge.

- **Organic** - type of practice driven IT are those that have a strong focus on customer/consumer facing technologies. These pseudo-product teams often manage B2C web presences, mobile application development, and other public-facing technology assets for the business. These groups are driven by market pressures and competition, and are often part of marketing departments. Often, corporate IT does not have these capabilities, particularly in the mobile design areas, and may have failed in such endeavors previously. Therefore, the business decides to undertake these activities independently of the official IT group.

- **Expert** – One of the more compelling reasons to be independent of traditional IT is when the solutions needed require such a high degree of subject matter expertise that the development of technology must be very tightly coupled with the experts. Highly complex solutions for specific business needs, close client contact for customized solutions and complex algorithm development can drive these groups to seek independence and a high degree of control over development. Often these organizations bring their high-level capabilities to solutions for scientific or complex financial applications where traditional corporate IT resources cannot (or are perceived to not be able to) deliver.

In each instance, it is very difficult to move these functions into the corporate IT organization. Often, this would demand a top-level corporate restructuring to accomplish. Additionally, IT may not be able to absorb these functions due to a lack of appropriate technology skills and high degree of expert knowledge

which increases the risks of trying to consolidate these teams into IT. Most often, that perceived risk allows the practice driven IT efforts to continue unmolested.

Model 2: Rogue Development

When most people use the term "Shadow IT," what they really mean is Rogue Development. That really shouldn't come as a surprise though- Rogue Development is one of the oldest and most common types of Shadow IT. If you've worked in IT for any length of time then you probably know of, or have participated in, several rogue projects during your career. Just what does Rogue Development really mean though?

Definitions

It turns out that there are actually two distinct types of Rogue Development- "black ops", and "skunkworks." Black ops are just like they sound- they are rogue projects that are never meant to see the light of day. Skunkworks projects, on the other hand, are semi-sanctioned projects that are eventually meant to become a legitimate part of the company.

Black ops are usually spawned by one or two employees that are just trying to get their job done. In many cases the employee has actually tried to work with IT, following protocol and using official channels, but nothing gets done. Eventually they come to feel that the only way they are going to make any progress is if they do it themselves. Black ops have almost no oversight, and the quality of the final product depends entirely on the skills of the employees involved. Because employees are just trying to get their job done, black ops projects can have serious problems including limited usefulness beyond the immediate problem, poor long term support, and little or no documentation.

Skunkworks tend to be slightly larger projects that are created to address a legitimate need that the central IT organization can't (or won't) fill. An employee will see such a need and will put together a small team to develop a solution- often with the blessing of a manager or business unit. If successful, a skunkworks project will often be folded back into the larger IT organization- either directly, or by providing a template that IT can use to provide a similar service offering. The team nature of skunkworks projects coupled with the oversight provided by managers or business unit's results in a much more useful end product. Because they aren't trying to solve an immediate problem- skunkworks teams can give more thought to things like documentation and how the final product will be used in the company.

Driving Rogue Development

No one shows up to work and decides to start a Shadow IT project on a whim- it's a lot of effort, it takes up time needed for assigned work, and it can get you in trouble. So if that's the case- why do we find Rogue Development projects in companies of every size, and in every industry? Despite the differences between black ops and skunkworks projects, they share many of the same drivers. These include the consumerization of IT, the introduction of mobile platforms, BYOD, the desire to increase productivity, IT project backlogs, lack of domain expertise, and off-cycle release requirements.

Let's take a look at each of these in a little more detail-

The *consumerization of IT* is having a big impact on organizations today. Employees are beginning to expect the same functionality and capability at work as they have at home. Why do they need to connect a cable to a projector when they can do it wirelessly at home with their Apple TV? It takes time for IT to test and approve technologies and that is a problem when you realize the rate at which new devices and technologies are being introduced.

Mobile platforms and BYOD are also hot topics with companies today- both from a development standpoint, and from an employee use case. IT can't support everything so the problem becomes "which platforms do they support?" Employees rarely notice when they have the technologies that they need to be productive- but they complain loudly when they don't. There are so many new productivity tools and services available that it would be impossible for the centralized IT organization to be familiar with them all, let alone try to support them.

Sometimes the driver for rogue development is simply the fact that even though they want to help, IT already has too much work to do. This leaves an employee with two choices; wait around for IT to get to their project, or implement something themselves and move on to something new.

What happens if IT doesn't have the experience or expertise to get a project done? Additional talent may need to be hired, or a contractor brought in. Both options are slow and will result in delays. If a developer has the expertise needed to get the job done, should they jump in? Or should they delay the project until IT can find or free up a resource?

Finally- what happens when you need to operate on a different schedule than that used by the traditional IT organization? IT may be able to support your requirements- but at what impact to the rest of the company? If you follow the IT organization's release schedule, you have to sacrifice agility.

Model 3: Purpose Driven Development

Purpose driven development can take on many forms, and can be as simple as a user creating a spreadsheet with special macros, to a sophisticated platform designed to perform complex tasks using specifically coded logic. Alternatively, the advent of Software as a Service (SaaS) platforms enables departments to contract with outside vendors to deploy a platform for employees to use completely outside of the enterprise's technology infrastructure. While these instances are similar to Practice Driven Development, they are less domain-knowledge focused. The story is familiar though; a need arises to perform a specific function, *and IT cannot, or won't* respond to this need. Let's examine several examples.

A mainframe-based record keeping system in a financial services firm is the system of record and primary computing platform. However, over the years as microcomputers proliferated among business users, it became more productive to replace manual functions with automated or semi-automated ones using common desktop software. Specialized documents were created using 'mail merge' functions that accessed a desktop database. Data was keyed into a desktop database from paper reports generated by the system of record. These documents serve a critical business function as they are required for revenue realization process, so they clearly serve a purpose, though the technology used and the logic implemented is not algorithmically sophisticated. What's critical here is that IT knows *nothing* of this system. In fact, this system probably comes to IT's attention when the users try to scale it by allowing multiple users to access the desktop database over the LAN, thereby triggering its corruption, which in turn triggers a disrupted business process.

Another scenario is when a business group accesses data from existing systems for reporting. Contract programmers hired directly by the business area are granted read only access to production databases. Ultimately, the reports written slow down the production databases which had been optimized for transactional performance rather than multi-join queries that fetch large amounts of data. In these cases, the size of the 'shadow' is often unknown. How many reports are out there? How many of them could be run concurrently? How optimized are the queries they contain?

Yet another scenario is when a business area determines there is a need, and they engage a SaaS vendor to fill it without IT's knowledge (at least in the planning phase). This has become a common scenario in the client service area resulting in the proliferation of social media tools, previously in the Sales force automation arena. Traditionally, IT would build systems often focused on the core competency of the business, such as record-keeping systems in the financial services area, or inventory control and procurement systems in the commerce space. When called upon to automate sales processes such as territory management, lead distribution, and activity logging, traditional IT organizations had a choice. Build such systems, or refuse. Some built them, and found that dealing with sales organizations was significantly different than dealing with financial or manufacturing groups. Requirements management was more difficult when interacting with sales professionals who loved the idea of showing up with a laptop computer in the early 90's, but knew nothing of the rigor required to manage technical projects. This, and the fact that most sales organizations do the same activities (lead management, territory

management, contact tracking and activity logging) led to commercially available products for sales automation, and ultimately to completely public cloud-based solutions. Today, very few sales organizations would consider building their own sales force automation system, or engaging IT to build one either. Such systems lie almost exclusively out of the realm of traditional IT, with their data and processing performed entirely on the vendor's infrastructure.

Section II - Evaluation of the models

Evaluation of Model 1: Practice driven development

Practice Driven development is requires subject matter expertise deployed in the service of some organizational goal, such as providing a very specific service to internal or external clients of the organization. As such, its technological requirements can be distinct from other areas of the organization.

Pros:

- Unique Value – The organization definitely benefits from the subject matter expertise inherent in these efforts. Often, these efforts have direct client impact, and are part of an overall revenue generating event.

- Alignment – the close alignment of business and technological staff in Practice Driven efforts ensures tightly coordinated requirements definition, development and testing efforts that would be impossible in more 'generic' development environments. This alignment is essential to the efficacy and quality of the delivered services, and often exhibit software development life-cycles similar to the best run Agile practices.

Cons:

- Duplication of Infrastructure – often, Practice Driven efforts can spawn their own infrastructures due to their focus on their unique requirements. Additionally, Practice Driven practitioners may want to retain control of their environments and keep them "under the radar" from traditional IT organization to avoid any interference. This invisibility may result in the inability of the IT organization to consolidate functions such as database administration, tools development and license management to achieve cost reductions. If the Practice Driven practitioners realize this, they could actually devote more effort to more valuable activities than infrastructure management.

- Compliance Risk – Any effort running outside of the IT infrastructure could result in a compliance risk. Security, privacy, and data governance requirements around data retention and deletion policies may not be implemented by the groups engaging in Shadow IT projects.

Evaluation of Model 2: Rogue development

Now that we understand the different types of rogue development and what drives them let's take a closer look at the impact they can have on your company.

Black Operations (Black Ops)

In order for a black operation to stay black- it has to have a small footprint. If the project has a large impact or involves a lot of employees then chances are it isn't going to stay hidden for very long. The small size of these projects is both a blessing and a curse.

Pros:

- The upside to black operations is that because these projects are smaller and more focused- they tend to be simpler- and thus easier to implement correctly. A simple project can also be documented after the fact- even if that isn't ideal.
- Small teams can be incredibly agile. They do not need to coordinate with a lot of people or with other teams which means they can adapt to changing conditions and new problems very quickly.
- The smaller footprint of these projects also means that there is usually a smaller risk to the company. Undoing an architectural mistake or fixing a security problem in a small project is easier than in a larger one. Even if a mistake isn't caught, the impact to the company is likely to be contained to the project in question (there are obviously exceptions to this).
- If you have a talented employee then the benefit of a black operation to the company can be enormous. The business gets the results they want faster and the IT department doesn't have to tie up valuable resources to do it.

Cons:

- A small project means a smaller pool of talent and fewer eyes that can catch a mistake. There is less knowledge and experience to draw on and the team may lack familiarity with specific technologies. Without any oversight it is easy for serious architecture and implementation mistakes to make it into the final product.
- Security is a big concern with black operations because it probably isn't a high priority for the employees involved and because they may not understand the security implications in the first place.

- Small teams also have less time at their disposal. That means they are much less likely to design their solutions with the larger IT organization in mind or to create documentation. After all- they're the only ones that are going to use it- right?

- The desire to solve a specific problem means that anything developed as part of a black operation is likely to have limited applicability to the larger organization. The time and effort put into these projects may have been better spent on a slightly larger project that would meet more of the organization's needs

- There is also the question of licensing. With no oversight employees may end up deploying software for which the company does not have appropriate or sufficient licenses. If the company gets audited they can end up paying a lot of money to resolve license issues.

- Lastly- what happens when something breaks? Operations teams are unlikely to know anything about the black operation and that can severely impact their ability to troubleshoot and respond to a problem.

Skunk Works Projects

Skunk Works projects may be rogue, but the larger teams and project sizes mean they can't stay completely hidden. That, coupled with the involvement of managers means there are different implications for the organization. Just as with black operations though, there are pros and cons.

Pros:

- Larger projects and larger teams mean more talent and more experience. That allows the right resources to be assigned to architecture and planning, documentation, and even security reviews. More eyes on the project also allow problems to be caught quickly and with a smaller impact.

- Limited management oversight preserves much needed agility, but also ensures that the goals of the project align with the larger goals of the organization as a whole. Things like licensing can be dealt with ahead of time resulting in fewer surprises later on. Good managers can also ensure that thought is given to operationalization of the final solution.

- Done right, a good Skunk Works project can deliver a complete solution to the organization quickly, and with minimal risk. The end result is both useful to and easily integrated into the larger IT organization. This frees up IT resources to tackle other problems and that is a win-win situation for any company.

Cons:

- Although the goal is to preserve agility in a Skunk Works project- some agility is going to be sacrificed in favor of reduced risk and more usefulness to the organization. The key is to strike a reasonable balance; you want agility not chaos.

- Even though Skunk Works projects are a little more open, there is still the very real possibility of duplicating effort. If one team sees a problem, then chances are good that another team did too. The last thing you want to do is have two teams working in the shadows and building the same solution (or worse yet, building a solution to a problem that IT is already working on).
- It is very easy for a Skunk Works project to get out of hand. Teams that have success solving the initial problem can become overconfident and attempt to tackle too much. A good manager can ensure that the project remains focused and on target.
- Finally, there is always the chance that a Skunk Works project will be seen by the larger IT organization as a challenge or insult. Keeping the scope small and solving a specific problem can help alleviate this perception. Good communication and management buy in can smooth out the rest.

Evaluation of Model 3: Purpose Driven development

While it may be tempting to say that Purpose Driven Shadow IT efforts are similar to Practice Driven efforts, they are in fact different. The key to differentiating Purpose Driven and Practice-Driven Shadow IT lies in the subject matter expertise of the practitioners. Practice Driven Shadow IT efforts require significant subject matter expertise from the practitioners, whereas this is not a characteristic of the Purpose Driven effort. Many Purpose Driven efforts often exist *because* there is no expertise in a particular area, yet there is a need.

Pros:

- Efficiency and Productivity - Purpose driven efforts often highlight areas where automation can have a significant effect on efficiencies and productivity. Requirements for salesforce automation drove the creation of internal private and successful commercially available solutions like 80/20, ACT!, and ultimately Salesforce.com. The same is true in the social media space when leveraged in client-service and self-service scenarios. Most enterprises would not build these tools on their own when numerous cloud-based solutions are offered. In many cases, these deployments may never interact with traditional IT. When they do, it's usually at the authentication level where Single-Sign-On (SSO) is desired.

- Integration - The popularity of mobile computing and the "consumerization" of IT drives the need for more integration of data from different sources. This makes it more difficult for IT to stay within its traditional boundaries, and ultimately for business areas to completely ignore IT. Blending data between the sales and finance organization in mobile apps would certainly drive new requirements for integration, authentication, and consumption of data *across* traditional and cloud-based platforms. Such issues inevitably pull IT back into the mix.

Cons:

- Duplication of Effort - Unfortunately, in large organizations where there may be several client service areas and/or sales teams, it may be possible (even probable) that multiple solutions can exist, leading to administrative, operational, and financial inefficiencies.

- Compliance Risk – when development is done by 'rogue' teams or by vendors secured directly by non-IT groups, this development may not comply with the design, security, data governance and privacy standards of the enterprise.

- 'Surprise' Requirements – Shadow 'productivity projects' often arise from enthusiastic application of desktop technologies. These tangential infrastructures are often 'discovered' by IT when there is an issue. Desktop file systems deployed as multi-user databases predictably fail when they are needed most, which is when high concurrent use occurs. If the system is important enough, IT might be engaged to write a system using a suitable technology stack, including a reliable RDBMS. In these cases, IT will often agree to do so since such automation can be seen as an extension of the traditional system of record, albeit on different technologies.

This effectively brings the formerly hidden systems out of the shadows, but at significant cost in resources. If a re-write on more reliable technologies is not feasible, IT is often called on to revive non-standard systems when the fail, often by restoring backups that inevitably result in at least some data loss and operational impact to the business area.

Section III – Shadow IT Service Provider Model

Figure 1 illustrates the intersection of traditional IT with the different Shadow IT efforts. Tranditional IT often provides at least some services, even to Rogue efforts. It is hard to imagine a Rogue effort not using at least email and telephony services provided by the traditional IT organization. The Practice and Purpose driven efforts have a larger intersection area than Rogue efforts. These types of Shadow IT are often offshoots or dependent on certain amounts of infrastructure provided by the IT organization. However, it's important to note that the traditional IT organization is not in primary control of these efforts. **Figure2** attempts to further illustrate the different service models under which the previously defined types of Shadow IT can operate. The red-shaded Service Providers are the non-IT providers. These could represent elements within the enterprise, or SaaS, PaaS or consultative elements operating independently under contract. The green-shaded elements represent traditional IT service providers. What becomes apparent is a "red-shift" as the Shadow IT forms move from Purpose Driven to Skunk Works. In other words, as the effort becomes more independent of the traditional IT organization, the amount of red in the diagram dramatically increases.

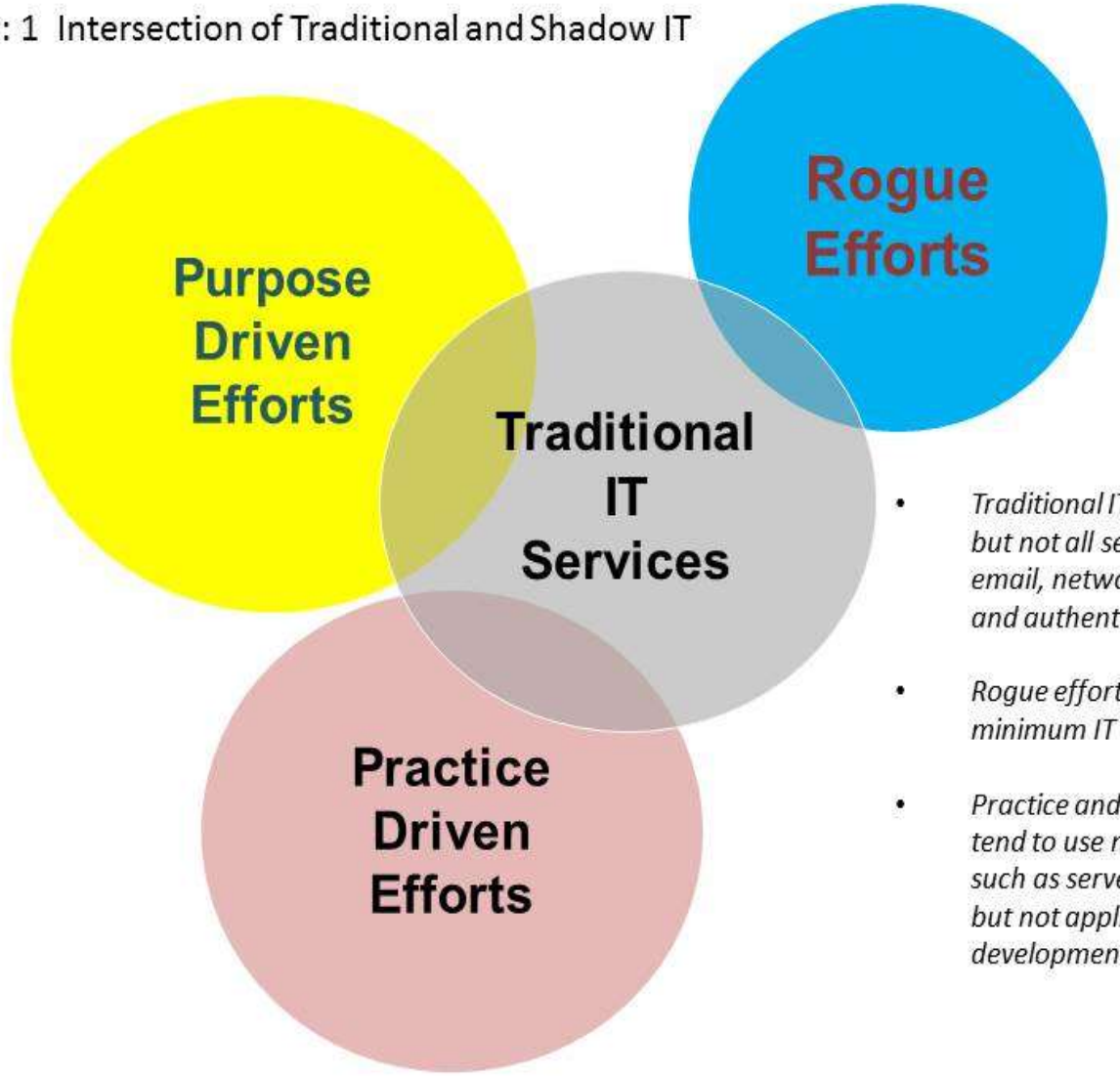
On the extreme left, the Traditional IT section of the diagram is almost entirely green indicating that the sourcing of services is almost entirely from the IT organization within the enterprise. The single exception is the involvement of Subject Matter Experts (SMEs) who would perform User Acceptance Testing (UAT). At the extreme right of the diagram is the Skunk Works, which is almost entirely red, and consuming very few significant services from the IT group. This is the extreme case, and in reality it's not out of the question to imagine the Black Ops or Skunk Works teams using traditional IT for fungible services such as electronic file storage and email, but for development efforts, the rogue elements of Black Ops and Skunk Works are clearly the most independent of the traditional IT organization.

The middle areas of the diagram, the Purpose Driven and Practice Driven areas represent the lessening of IT involvement in development efforts. The reason of the half-green, half-red services in these areas is that in many cases, development efforts may leverage previous or current efforts undertaken by the IT organization. A Purpose Driven program that utilizes existing or newly created tables in a relational database maintained and modified by IT DBAs in one example. Another is that the actual network infrastructure of cabling, switches, and routers may be under control of the traditional IT I&O organization, but the devices hooked to this infrastructure and the programs run on them may be the result of Purpose, Practice, Black Ops, even Skunk Works development projects. The red half of the hybrid services indicate that some of the infrastructure may in fact be partially or completely outsourced to SaaS/PaaS vendors . Some network VPN connectivity from the traditional IT infrastructure may be required even in the fully outsourced models, so the green half of I&O Support blocks remain throughout all of the sections.

There is a danger represented by the hybrid red-green blocks. Issues of compliance with enterprise policies regarding security, data governance, design and architectural standards are difficult enough to enforce within the traditional IT organization. They are much harder to enforce within organizations that are purposely operating shadow efforts, partially or completely.

The Shadow IT Service Provider Model is offered here as a tool to help illustrate the phenomena of Shadow IT, and to help analyze its impacts and risks to an organization.

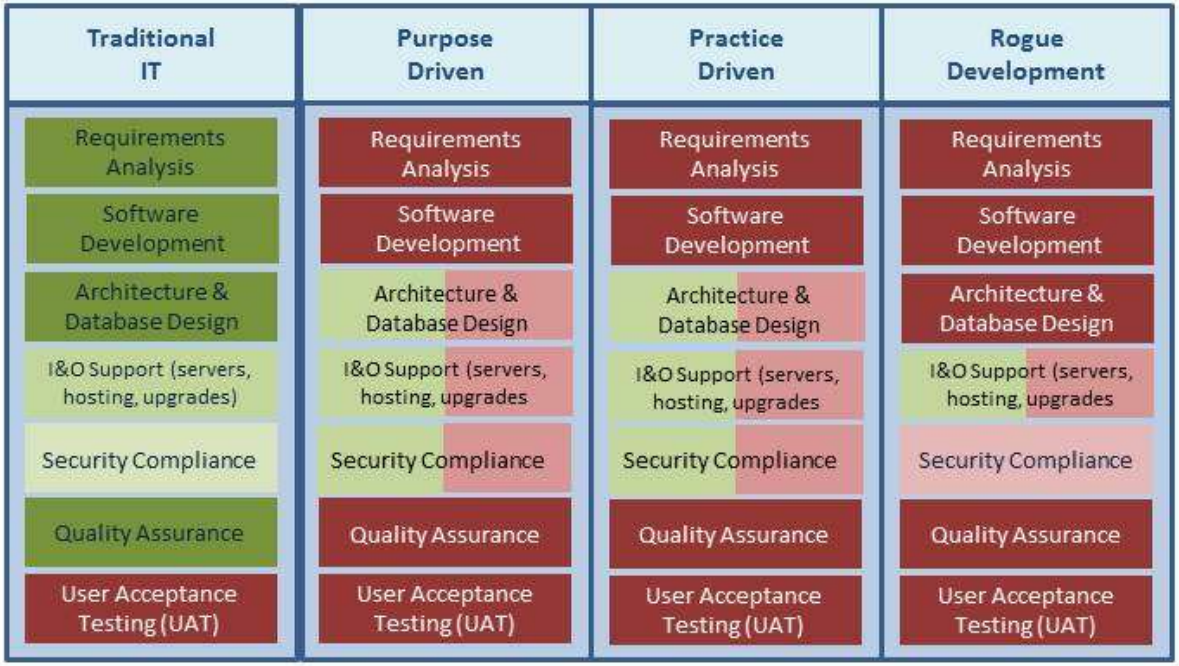
Figure: 1 Intersection of Traditional and Shadow IT



- *Traditional IT provides some but not all services, such as email, networking, directory and authentication.*
- *Rogue efforts tend to use a minimum IT services*
- *Practice and Purpose driven tend to use more services, such as server provisioning, but not application development and/or support*

Figure: 2

Shadow IT Service Provider Model



Microsoft Enterprise Architecture Roundtable, 2014

Section IV – Recommendations

Ultimately, the proliferation of Shadow IT comes down to the inability of the IT organization to provide solutions needed by the business. Unfortunately, whether we like it or not, IT organizations have limited resources. They need to allocate people and materials to the projects that are going to provide the greatest benefits to the company. So it becomes clear that that the only way to completely eliminate Shadow IT would be to have an IT organization with unlimited resources. Since that isn't possible- the next best option is to figure out how we can leverage Shadow IT to our advantage. There are actually two ways we can do that.

Shadow IT development arises out of a need - If someone goes through time and effort to port their application to a new phone, or to sneak a new operating system into the organization, chances are that those are technologies that IT should probably start evaluating for wider deployment. In other words- *Shadow IT Development projects can serve as an indication of where IT should allocate resources*.

Shadow IT Development can provide direct value to the company - both by freeing up IT resources for other projects, and by producing solutions directly. The challenge is to make sure that the work being done by the Shadow IT practitioners is a *net gain for the company*. A well designed, well documented project that fits into the overall business strategy is a definite win. A poorly documented project that is difficult and costly to support is not.

In many cases, the best thing to do is accept that Shadow IT exists and embrace it. Our goal should be to bring these projects out of the shadows and to encourage greater transparency. We need to give rogue developers the freedom to innovate- while at the same time providing guidance and resources whenever feasible. The more open this process is- the faster we can detect problems, and the better the end result will be. In fact, encouraging "rogue" efforts properly would result in "Citizen-Developers", or "Subject Matter Developers", rather than the more negative implications of "rogues".

However, it would be remiss not to point out that there are costs and risks associated with Shadow IT, so while we can learn from these practices, a modulation of our approach seems in order.

Learn From the Dark Side – Shadow IT in all of its guises is indicative of a delta between the services provided and those desired by non-IT professionals. Ignoring this delta can be tempting from a command-and-control perspective, particularly when resources are scarce. But the fact that funding is occurring in adequate amounts to fund IT functions within non-IT organizations suggests otherwise. Organizational realignment and even outright organizational change may be needed to affect significant positive outcomes. For years now, journals and blogs have spoken of the Business-IT divide, with the business side often complaining that IT was too slow, too unresponsive, and the IT side complaining that they had no seat at the strategy table, yet were essential to the ongoing operations of the business.

Consolidate or Commiserate? – if multiple groups outside of IT are engaged in duplicative efforts , spawning a wide range of software stacks and infrastructures, it may make sense to attempt consolidation of these groups. However, this is often a complex and dangerous task, despite the obvious benefits of decreased costs as systems are moved to a common platform. Underestimating the risk is the primary danger here. If however, the 'shadow' elements are practice driven rather than purpose driven, it may make more sense to commiserate, and determine what services IT can provide that all the shadow groups could leverage. Additionally, making sure that privacy, security and data governance policies are uniformly applied may be the real value IT can add to this scenario.

It will not always be possible for IT to enable and/or control every form of development in the enterprise. In fact, this is not desirable. There are wholly legitimate reasons for independent development groups to exist, particularly in the practice-driven development model. While IT may not provide direct services in many cases, IT can insist that corporate standards for handling data in terms of privacy, security and retention policies are followed. Additionally, IT can provide those services that are invested with a more public interest in the enterprise, such as document management, email, hosting and relational database services. Managing these centrally has great benefits, and helps control costs and manage server patching, upgrading, and security. Taken a step further, IT can consolidate only those services held in common by separate practice areas. For example, if each practice area has separate Database Administrators (DBAs) this function could and should be consolidated into a service each practice area can use. This can help IT reduce redundancies and the associated costs, without having to usurp complete control of the practice-driven effort.

Enable the Dark Side – ultimately, enterprises will always have a need for something IT doesn't currently provide, and expanding the existing IT department to provide every possible service is clearly untenable. So what should a rational CIO do? The answer is to be an enabler, while protecting the transactional and relational integrity of core data. Concepts such as transactional integrity are key to effectively doing business. Such concepts are supported via software and database tools that coordinate complex transactions and ensure they can be rolled back if any component of the transaction fails. This kind of functionality must be protected, and this falls solidly in the purview of IT. So how does a 'rogue' development effort, outside the auspices of traditional IT interact with the systems of record? Should they? As an enabler, the answer is 'yes'. The key to enabling is to expose functionality safely via an integration hub. Integration hubs allow authenticated applications safe access to the data and functionality of the SOR. Hubs generally host web services, and more recently API's that can be called using common web methods of the HTTP protocol. Such services and APIs are suitable for consumption by non-SOR users because the interface is tightly controlled, and the SOR maintains all of the transactional logic; it is only the interface that is exposed. Integration hubs also act as security points, allowing only authenticated applications to interact with the hub's services. IT's purview extends to the integration hubs APIs and services, but no further. The 'contract' between the consumer (which could be a business-driven mobile application) and IT in this case is the web service and/or API exposed on the service hub. As long as the consumer uses the mechanism properly, they will be guaranteed a valid result. If the consumer does not call the mechanism as specified, failure results, with no expectation of

support from IT. In this way, IT can provide secure, scalable and reliable access to SOR data and functionality to ‘citizen developers’ without having to control that community.

Conclusion

In this paper we've attempted to define 'Shadow IT' and identify the forms such efforts may take within the enterprise. We've attempted to shed light on the drivers for such efforts, and offer approaches for analyzing their impact to the organization. Perhaps the most significant conclusion that can be made about 'Shadow IT' is that it must be acknowledged because it tells us so much about how the organization *wants* to use technology, versus how IT tells the organization how they *should* use it. Such gaps are dangerous if ignored. Treating 'Shadow IT' efforts as in invasive species only leads to fruitless eradication and control efforts. Learning the drivers of this phenomenon can lead to understanding, and ultimately a coherent change in policies that can bring IT and the business together in a collaborative fashion as never before.