

Maximizing Web Services Project ROI with Actional SOAPstation™



Actional

Table of Contents

Introduction	3
Anatomy of a Web Service Project	3
The Challenges of Web Services Management	5
Managing Service Lifecycles (Minimizing the Impact of Change)	5
Defining and Enforcing Service Level Agreements	5
Ensuring that Services are Secure	6
Provisioning Services to Meet Specific Consumer Needs	6
Applying Global Business Policy	7
Modeling Web Services Project Costs	7
Lowering Management Costs	8
Conclusion.....	10
About Actional Corporation	11
Appendix A: GENERAL PRODUCTS PROJECT ASSUMPTIONS	11

INTRODUCTION

Over the past several years, the emergence of Web services has become one of the most talked-about advances in the technology industry. Throughout corporate IT, there is growing interest – and a bit of skepticism – about the promise and potential of Web services for corporate computing. “Will Web services change the way I do business?” “Web services are the holy grail of computing.” “Is anyone really doing Web services?”

Indeed, Web services do represent tremendous business opportunity. As leading companies begin to develop and use service-based applications and development tools, application development costs are dramatically reduced and business agility is accelerated. However, while the development costs are often planned and understood, little attention is paid to the cost of service infrastructure, service management and service maintenance.

Without taking these additional costs into account, project ROIs can be greatly overstated and miscalculated. In fact, the cost of management can easily dwarf the cost of development, which can lead to insufficient funds for project completion. Fortunately, products such as Actional SOAPstation can dramatically and predictably lower these costs and allow the organizations to fully reap the benefits that Web services can deliver.

ANATOMY OF A WEB SERVICE PROJECT

To illustrate the critical need for Web services management, and to illustrate its value, let's take a look at a typical Web service project.

General Products is a F1000 multi-division durable goods manufacturer. The Vice President of Customer Operations has recognized for quite a while that his customer service operation is fundamentally inefficient due to the lack of integration of corporate and inter-divisional information. In fact, to enter an order for a customer, the customer service rep must go through four distinct processes, each of which is tied to a different application system:

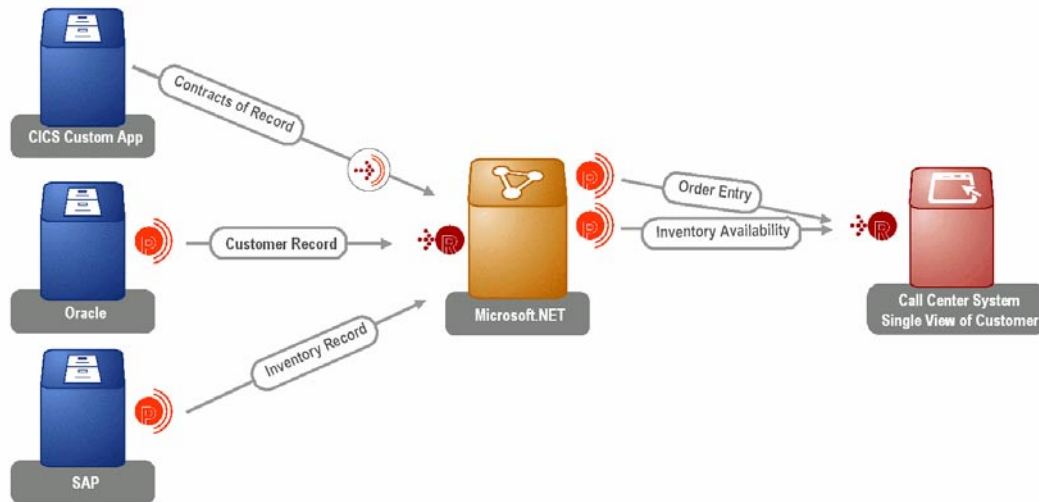
1. Gather customer information from the Siebel system
2. Check credit status and terms in a mainframe application
3. Check inventory availability in each division's SAP implementation
4. OK the order in the Siebel system

This process is complex, time consuming, and actually requires the service rep to hand write information or get a fax from the customer, process that information and then call the customer back.

General Products has assigned an IT team to solve this problem, with the goal of building an application that can complete this process in a single step, leveraging the existing applications. They have chosen to use Web services to attack this problem and decide to create two services: Enter Order and Inventory Check.

Using one of many popular products such as Microsoft.NET or BEA Weblogic Workshop, the development team implements this plan, first reaching back into their applications for the business logic and data from which they create these new services. The services are then integrated into an existing customer service portal that can natively consume them (see **Figure 1**). Once implemented, these services will allow for greater throughput of orders and improved customer service, enabling them to reduce the headcount required to process orders while improving customer service.

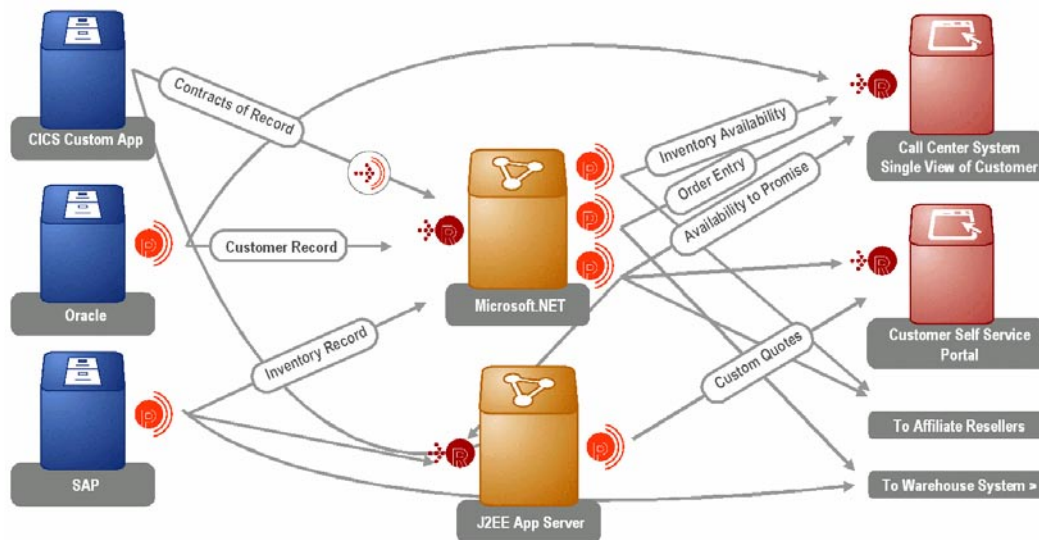
Figure 1



Because Enter Order and Inventory Check are now Web services, General Products can reuse these services to drive incremental business benefits. General Products now leverages the Enter Order service in new and exciting ways such as allowing key customers to directly enter orders via a self-service portal. They can also make these services available programmatically to their affiliate resellers and outsource warehouse providers.

Seeing the success of this service-oriented approach, the team creates an Available-to-Promise service, building on the other available services. Another development team creates a Custom Quote service. With this explosion of services and consumers, the simple project has now blossomed into a complex yet valuable set of business interactions as depicted in **Figure 2**.

Figure 2



If the story ended here, Web services would represent the ultimate free lunch, a perpetual motion machine of business benefits. However, it is not that simple. In reality, while attempting to scale and operate the project described above, General Products will face a growing set of daunting challenges. Without proactively addressing these issues, the General Products project is doomed to fail.

THE CHALLENGES OF WEB SERVICES MANAGEMENT

The specific challenges facing companies deploying projects such as General Products' can be classified into five main categories:

1. Managing service lifecycles (minimizing the impact of change)
2. Defining and enforcing service level agreements (SLAs)
3. Ensuring the security of services
4. Provisioning services to meet specific consumer needs
5. Applying global business policy

While every project will not necessarily need to address every one of these issues, even the simplest of projects will face many of them. Let's take a look at each of these problem categories to understand its potential impact on the General Products project.

MANAGING SERVICE LIFECYCLES (MINIMIZING THE IMPACT OF CHANGE)

Services, like applications, have lifecycles. The lifecycle of a service typically consists of service creation, service deployment, service upgrades (sometimes including rollbacks) and service retirement. We've seen how easily services can be created and deployed when Web services are used; however, there are significant issues in upgrading, rolling back and retiring services from usage once they are deployed. Managing these lifecycle events and their impact on operations is a complex and potentially costly operation precisely because Web services consumers, by their very nature, are loosely coupled to the provided services. Consequently, when a service changes, organizations must first identify all of the consumers of that service and coordinate the simultaneous upgrading of all consumer applications. This process can be extremely costly, time consuming and complex.

For example, let's say that General Products needs to upgrade its SAP system and add new functionality to the Inventory Record Web service. The .NET platform, the J2EE application server, and the warehouse system all consume this service. Many downstream services also depend on this service. Beyond the cost of upgrading the service logic itself, the General Products project team must coordinate upgrading the three consuming applications, recode those applications, and schedule downtime and upgrade processes between the applications and consumers of downstream services. Even in this simple example, the ability to roll out a logic upgrade is both complex and expensive.

DEFINING AND ENFORCING SERVICE LEVEL AGREEMENTS

By definition, building applications using Web services entails the composition and assembly of services that are loosely coupled and provided by other systems. It is critical to understand the performance characteristics of the services that will comprise any given application.

In addition the developer of a given application may be held to provide specific levels of performance on services that he delivers to others. Therefore, Web service projects must include the ability to define, measure and monitor specific performance characteristics of the services consumed and provided by the project. These sets of performance characteristics are often referred to as service-level agreements (SLAs) and can be either explicitly or implicitly defined and managed.

Further, project administrators must be able to quickly spot variations in performance so they can act to prevent failure and minimize the impact of failure. Overall, the costs of failing to actively manage performance can range from lost revenue and productivity to actual hard dollar penalties from failing to meet explicit SLA targets.

In the General Products example, the Web services team has promised certain levels of response time to the team that manages the customer service and self-service portals, specifically around the response time and availability of the Inventory Availability, Available-to-Promise, and Enter Order services. Failure to meet these targets results in an escalating set of charge backs to the IT group, based on estimated productivity and revenue loss. The mission-critical nature of these SLAs requires a comprehensive system of alerting, monitoring and reporting. Developing this capability adds time and cost to the project, delaying its rollout and time to value. In addition, because this capability is custom built, it is expensive to maintain, change and deploy as the needs of the project change, adding still more life cycle costs to the project.

ENSURING THAT SERVICES ARE SECURE

Because Web services often move important and valuable business data, security is paramount. The key in securing services is to verify that the users of such services are both authenticated (they are who they say they are) and authorized (they have permission, based on business rules) to access the service they are requesting.

Most organizations have already invested in user directories such as LDAP and identity management systems, as well as single sign-on systems such as Netegrity. However, integrating the Web services into these infrastructures requires custom coding for each service. Moreover, different services may need to authenticate and authorize users based on different user directories and or credential types. Lastly, there are many situations where access to services must be gated by business logic – for example, in many regulated security markets it is actually illegal to take orders during non-market hours.

General Products is faced with this exact problem. While they have implemented Netegrity Siteminder internally for single sign on to all applications, their customers and affiliate resellers are tracked in an LDAP directory dedicated to external users. While internal users have access to the SAP, Oracle and mainframe systems, external users do not. The extensive security tasks that the development team must tackle include:

- Delegating authentication and authorization of internal service requests to the Netegrity system
- Authenticating via a look-up in the external LDAP and then authorizing external users based on their organization and the specific services and operations they wish to access.
- Securing the same service for both internal and external users, using the methods above
- Ensuring that customers that order directly do not exceed their credit limits
- Being prepared to take advantage of emerging Web services security standards as they become available in commercial products

In effect, securing Web services into existing security systems requires creating and maintaining another layer of infrastructure, further increasing General Products' total project cost.

PROVISIONING SERVICES TO MEET SPECIFIC CONSUMER NEEDS

Web services represent standard interfaces to business logic. When any service is created, it specifies the set of operations, data types and message formats to be exchanged between a provider of a service and its consumers. For example, an order entry service may have an operation that allows the consumer to enter multiple line item orders. This operation may be called `m_enterorder`.

The `m_enterorder` operation then specifies what data elements are required, the data type for each element, and how to structure the request document. It then specifies what the requester of the document should expect as a reply. All of this is accomplished through the publishing of a WSDL file associated with this service.

However, it is likely that a service consumer may already have a standard interface to other systems, typically via a pre-defined XML document. In many cases, these consumers have the business power to demand adherence to their standard, not the one the provider has defined. For example, large customers often force smaller suppliers to conform to their standards or parent companies require conformance from recently acquired subsidiaries. Beyond that, two applications that have been service enabled may define data elements differently altogether. Clearly, provisioning services to match the needs of specific consumers is another cost driver in Web services projects.

As General Products rolls out its services, they prepare to make the Enter Order service available to their largest reseller. This reseller accounts for over 30 percent of General Products' domestic volume, and therefore holds a great deal of power over them. As part of the latest round of negotiations, the reseller stipulated that General Products must accept the reseller's XML documents for all order-related transactions. These documents vary significantly from those defined in General Products Web services definitions. General Products is now contractually bound to support these documents within their Web services project, and must figure out how to do so without scrapping all the work they have invested in their currently deployed services. Once again, project costs increase incrementally.

APPLYING GLOBAL BUSINESS POLICY

Often, when building Web services, there is a need to apply global business policy. Examples include controlling access to services based on credit limit, the appending of billing codes to all messages, or the routing of orders to different systems based on part number, where certain types of orders are handled by different back end systems. Global applications of this type of logic can significantly lower the cost of rolling out and maintaining services.

MODELING WEB SERVICES PROJECT COSTS

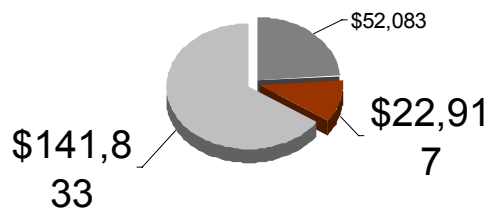
Given the costs identified in the five sections above, what is needed is a model for defining and calculating them. A simple summary of these costs is presented in **Figure 3**, divided into initial and ongoing project costs.

Development Costs	<ul style="list-style-type: none"> • Service Logic Development • Maintenance and Upgrade Development
Infrastructure Development/Integration	<ul style="list-style-type: none"> • Service Reporting, Alerting and Monitoring Development Costs • Service Security Integration with existing systems • Maintenance and Upgrade Development
Ongoing Management Costs	<ul style="list-style-type: none"> • Maintenance and Upgrade Roll-out and Administrative Costs • Performance Monitoring • Cost to Provision to specialized needs • Cost Associated with Missing Service Level Agreements • Cost of managing and maintaining Global business policy

Figure 3

In fact, applying these costs and the assumptions listed in Appendix 1 to the General Products project, in which four services are distributed to 10 consumers, yields the results displayed in **Figure 4**.

General Products Web Service Project, Year 1 Total \$ Costs



General Products Web Service Project, Year 1 Total % Costs

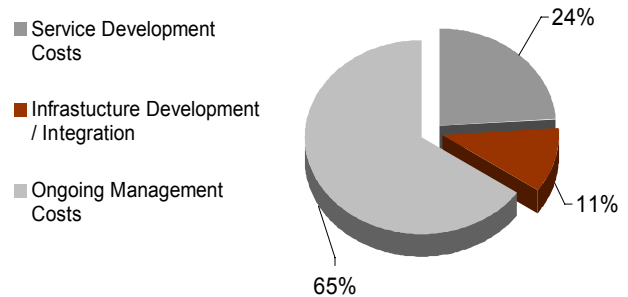


Figure 4

These charts show in this relatively modest project, more than 75 percent of the first year costs are related to the infrastructure and ongoing management of the services. Of the \$217,000 spent on the project, more than \$164,000 was spent on the management related costs. This is dramatic proof that management costs must be accounted for in understanding the cost of any Web services project.

LOWERING MANAGEMENT COSTS

Without doubt, the feasibility and ultimate success of any Web services initiative depends on dramatically reducing the management costs. Actional Corporation is focused on this very issue and has introduced the Actional SOAPstation Web Services Management Broker to lower the costs of deploying and managing Web services projects.

Actional SOAPstation is an active Web Services broker that simplifies the secure distribution of Web services to broad networks of consumers - extending the reach and enhancing the business value of those managed services. SOAPstation acts as a Web services proxy, brokering the interactions between applications providing Web services and the systems that build on them.

Figure 5 and **Figure 6** illustrate specifically how the capabilities of Actional SOAPstation bring Web services management costs under control.

The following table (**Figure 5**) revisits the five cost categories with their associated cost drivers, and describes how the capabilities, features and benefits of Actional SOAPstation directly contribute to dramatic cost savings in managing and maintaining Web services projects.

Figure 5

Cost Category	Actional SOAPstation		
	Capability	Features	Benefits
Managing Service Lifecycles	<ul style="list-style-type: none"> Provides point-and-click facility to allow for service upgrades, roll-backs and changes without requiring consuming applications to recode Supports rolling, cut-over, and load balanced upgrade strategies 	<ul style="list-style-type: none"> In-flight transformation of incoming and outgoing messages Content- and context-based message routing Ability to reuse the same service multiple times Revision capability to track and maintain deployment changes 	<ul style="list-style-type: none"> Reduces the administrative effort and time to deploy service upgrades Increases satisfaction of downstream service consumers Improves time to market for new service changes
Defining and Enforcing Service Level Agreements	<ul style="list-style-type: none"> Point-and-click ability to set performance parameters for auditing Point-and-click ability to define and implement warning and error levels for real-time alerting Automatic reporting on service-level performance 	<ul style="list-style-type: none"> Automatically captures broad range of statistics on message traffic Generates standard statistics and reporting logs Out-of-the-box reporting interface Alerting and auditing can be triggered on both performance and business data Data logged to standard databases for access by reporting tool of choice 	<ul style="list-style-type: none"> Virtually eliminates the time and effort to integrate reporting, alerting and monitoring into services Dramatically reduces penalties and business costs of missing performance expectations Provides far more robust capability than typical custom built interfaces
Ensuring That Services are Secure	<ul style="list-style-type: none"> Point-and-click definition of each service's authentication and authorization strategy Allows the same service to be provisioned with multiple security strategies and user directories Seamlessly integrates service security with existing infrastructure such as LDAP, and single-sign-on systems Intermediates mismatched security systems, allowing the mapping from one system to another 	<ul style="list-style-type: none"> Supports all common credential types such as username and password and client certificates, as well as emerging standards such as SAML and WS-Security Supports any number of user directories Supports all major LDAP and single-sign-on implementations Provides security interceptor for custom implementation Provides authorization based on roles, organization or a combination of both Can provided mapped credentials between incompatible systems 	<ul style="list-style-type: none"> Virtually eliminates the time and effort to integrate new services with existing security infrastructure Provides ultimate flexibility to tailor security implementation to specific needs on a service-by-service basis Future proofs deployment against changes and implementation of new standards Provides for more robust and flexible capability than typical custom built infrastructure
Provisioning Services to Meet Specific Consumer Needs	<ul style="list-style-type: none"> Allows the same service to be provisioned to specific consumer needs by creating a tailored access point to that service 	<ul style="list-style-type: none"> Point-and-click creation of new service variants Services can be provisioned to support varying message structure, data format, transport protocol, and security needs Graphical transformation builder and high-performance runtime transformation engine 	<ul style="list-style-type: none"> Eliminates the need to rewrite services to meet special consumer needs Increases the business value of services by increasing the potential audience for those services Lowers the cost of service maintenance over time
Applying Global Business Policy	<ul style="list-style-type: none"> Allows for the application of business logic to all message flows, requests, replies, and faults Point-and-click deployment of global business logic such as credit checking, billing code appending, or business rule-based access control 	<ul style="list-style-type: none"> Global intercept capability support code-based, transformation, and Web service-based intercepts Business rule-based access control Business rule-based firing of interceptor and access control policy decisions 	<ul style="list-style-type: none"> Lowers the costs of applying global policy to Web service deployments Lowers the cost of maintaining deployed business policy

Further, **Figure 6** shows the costs calculations of General Products' project with and without Actional SOAPstation deployed (See Appendix 1 for cost assumptions). As you can see, when applied to the General Products modest Web services project, Actional SOAPstation lowered the total project costs by over 73 percent and reduced the cost associated with Web services management by over 97 percent!

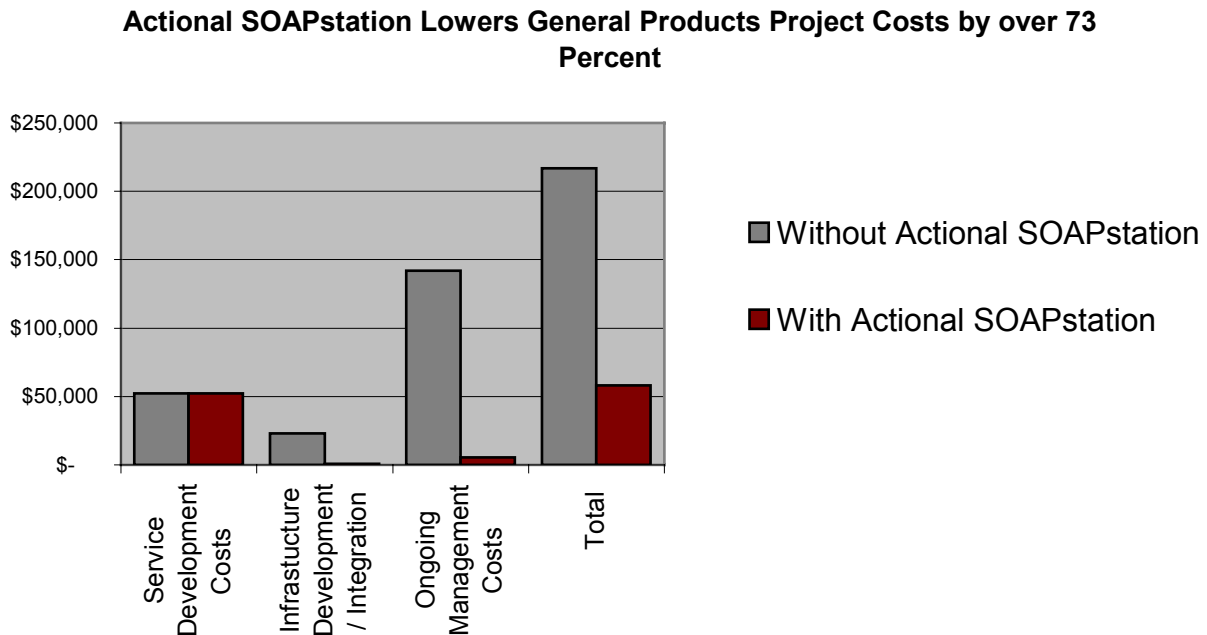


Figure 6

CONCLUSION

Web services offer great promise for dramatically lowering the cost of application development and substantially increasing IT's ability to respond more quickly and with greater agility to changing business requirements. However, as illustrated in this paper, the dynamic and changing nature of Web services creates many management challenges that can quickly overwhelm even a modest project with skyrocketing costs and complexity.

Every technology innovation brings with it a need for management. Actional SOAPstation is precisely the right product at the right time to meet this need. With its ability to reduce costs as shown in this paper, Actional SOAPstation is critical to the infrastructure of those who are serious about deploying Web services at any scale.

Leading companies recognize that a Web services management solution, such as Actional SOAPstation, is a required – not optional – component in their Web services architecture. Only by deploying this type of can organizations rein in the management costs of Web services and be in a position to capitalize on their potential

For more information about how Actional SOAPstation can significantly improve the ROI of your Web services deployments, contact Actional at 1-650-210-0700 or email info@actional.com.

About Actional Corporation

As the leading provider of SOA enablement solutions, Actional Corporation addresses the specific challenges of securing, deploying and managing service-oriented environments, from early Web services projects to enterprisewide SOA initiatives. The company's sole mission is to keep customers' SOAs secure and operational 24/7, making Actional the choice of leading organizations around the world including Danish Immigration Service, Partners Healthcare, Thomson Prometric, MCI, Telstra, Travelers, the U.S. government, and others. With Actional's SOA Command and Control Platform, customers can reduce costs and ease the complexity of SOA deployments, thereby increasing the responsiveness of IT, accelerating time-to-market of business-critical applications and capitalizing on the business value of their SOA. Actional, which recently merged with Westbridge Technology, is a privately held company based in Mountain View, Calif.

APPENDIX A: GENERAL PRODUCTS PROJECT ASSUMPTIONS

Salary

Annual Burdened Salary of Developer	\$ 125,000
Annual Burdened Salary of Service/Network Administrator	\$ 90,000

Services

# of Services developed	4
Time in Developer days to create services	20
Time to Developer days to integrate each service with security code	5
Time in Developer days to build reporting into each service	5
Time in Developer days to build billing integration into each service	1
# of Services Upgraded (%)	50%
Time in Developer days to upgrade services	10
Avg Number of Consumers per service	10
% of consumers requiring custom provisioning	25%

Management

Maximum # of Services monitored per administrator	10
Days to rollout service upgrades per service/ per consumer	3
Development days to upgrade consumer clients	5
Days to custom provision a consumer	3

Service Level Agreements

Avg of Service Requests/service/year	12500
SLA Misses	4%
Cost/miss (Penalty/Missed Revenue...)	\$10



800 W. El Camino Real, Suite 120
Mountain View, CA 94040
Phone: 650-210-0700
Email: info@actional.com
Web: www.actional.com