SHORING UP THE SAFETY NET

How Splunk Software Improved Application Delivery at a State Government Agency

Use Cases

• Improving Application Delivery
• Streamlining Internal Processes
Executive Summary

State and regional government human services agencies are faced with a difficult task. With ever-changing, sometimes threatened budgets, governments need to meet their missions to distribute funds and support efficiently, cost-effectively and proactively. When crises occur, agencies can be overwhelmed by demand for services. But unlike other organizations, when mistakes happen the cost is measured in more than dollars. A disabled elder can end up in the street due to a check that never arrived; an underemployed family can face a choice between dinner and heat. Poorly designed systems can lead to mistakes that cause delays in necessary benefits. These delays ripple out to those who also benefit from these funds indirectly — landlords, small businesses and others whose income depends on serving people at all ends of the economic spectrum. Economic stability of an entire state can be affected if a problem occurs and ripples to hundreds of thousands of recipients. What’s more, the federal government assesses steep fines based on performance levels and user service.

One U.S. state government health agency had recently upgraded from a legacy platform to a new system with a new mission: to provide extraordinary customer service to the citizens of the state, coupled with transparent accountability. To do this, the agency had to coordinate access to a multitude of federal and state programs and benefits. Staff had created an eye-pleasing, seemingly user-friendly portal to simplify access to benefits—but the volume of support calls didn’t go down, and they had no way to understand why.

Employees and the support staff were deluged with high-impact calls, with no way to gain insight into the design decisions or code that might be triggering errors. They searched for a way to see exactly which interface elements were tripping up users so they could tweak them and reduce confusion and calls—but the portal led to many types of systems, with little true coordination behind the scenes. This lack of coordination and visibility into problems magnified to create endless phone calls with highly emotional clients, and an easy target for fraudsters who figured out inroads into the system. While the portal looked better, nothing had actually changed for constituents or for the teams responsible for how the portal components worked. Resources to fix each problem added up quickly because no one knew what the issue was nor who was the best person to solve it. Due to poor access statistics and deficient reports, the state faced millions of dollars in federal fines.

Then, at a government conference, the agency discovered Splunk software, which changed everything. The software gave developers and IT Operations teams the insight needed to redesign and revitalize the system, quickly reducing errors prior to production, customer support calls and constituent frustration. Splunk software makes it possible to produce meaningful reports quickly, exposing procedural issues and logic flaws that had caused delays and endless loops for constituents. Splunk software enables management to optimize developer time based on platform knowledge, gaining understanding of what is important to the end user, enhancing and ensuring customer satisfaction. What had been a tedious and manual process was replaced with a “rapid-fire insight machine.”
Suddenly, with Splunk dashboards, the agency experienced:

- **Reduced errors leading to vastly improved call wait time.** The mission statement for this government agency includes superior customer service to those taxpayers who suddenly find themselves in need of services. Not meeting this goal can result in multimillion-dollar federal fines. When outages or system issues appeared, clients would find themselves on hold for hours while customer support scrambled to solve the problems. Now, with vastly reduced MTTR, frustration levels have plummeted, and downtime situations are remedied before they affect vast numbers of constituents.

- **As-it-happens view of user experience for support and development.** Once the customer could see exactly which parts of the website design were causing the most confusion, it could focus developer efforts on fixing them. Simply shifting a button from one side of the page to another or changing the text on a label can reduce errors significantly. With Splunk, developers have a virtual real-time laboratory where they can validate changes that make significant improvements before production.

- **Quick answers to questions about constituents.** Once the fires were put out, it was time to understand who was using the system and how they used it. Splunk software mines dozens of logs to provide

### Business Benefits at a Glance

<table>
<thead>
<tr>
<th>Challenges</th>
<th>How Value Is Measured</th>
<th>Business Impact</th>
</tr>
</thead>
</table>
| • Inability to correlate data from multiple systems | • Insight into processes  
• Ability to create meaningful reports | • Met federal metrics and avoided fines  
• Reduced Mean Time to Resolution (MTTR)  
• Improved uptime  
• Staff can focus on solution rather than diagnosis |
| • Lacked insight into user interaction | • Optimize developer time based on platform usage | • Developers spend time improving system rather than trying to collect data  
• Appropriate developers identified and allocated according to user pain |
| • Needed alerts to identify anomalies | • Quickly determine when issues arise with a system | • Proactive rather than reactive solutions avoid expensive support calls |
| • Lacked access to data from disparate systems | • Experts from different groups can troubleshoot across systems | • Dashboards and reports maximize the time and expertise of analysts, replacing tedious processes that yielded little impact |
visibility into what constituents need the most, and where developer, IT operations and business efforts should focus. For example, developers discovered that constituents who apply for food stamps and state health care support use different access methods and they can fine-tune the different sections of the portal.

- **Visibility into the big picture.** Before the agency deployed Splunk, questions from the governor’s office and state legislature were difficult to answer quickly, if at all. In the aftermath of natural disasters, it could take months to figure out basic facts, such as where most of the new claims originated. With Splunk dashboards, it is possible to look at a map of the state and spot outliers for further investigation.

**Too Many Systems Caused a State of Distress**

Like most American state agencies, this agency was responsible for a multitude of programs, each with its own sets of data and systems. To make it easier for citizens and management, the agency had created a portal in an attempt to coordinate benefits to the extent possible, as some of these systems interfaced with larger federal programs. Not only are all these programs managed from one portal, but the system also is required to incorporate eligibility determination, policy development and rule making, fraud and abuse protections, and ombudsman services.

While many companies deal with sophisticated end users, government agencies serve the widest range of users imaginable: the elderly, those who have limited access to computers and devices, people who might be less educated and computer-savvy, and those who are extremely stressed in the face of a personal or natural disaster. What’s more, government customers often access the computer systems from a library or public space, limiting techniques that would make repeat access less complex.

The Information Technology team, along with partners from major consulting firms, had created this portal so constituents could see a big picture view of the services. But while the portal appeared to be an oasis of benefits for the end user, it was a mirage when it was first deployed. The backend was totally disjointed and difficult for the team to implement, manage and troubleshoot. Various groups and departments were responsible for managing these tasks and the underlying systems that made them possible, spanning a wide variety of applications and data. It would never be possible to fully streamline all systems or redesign them into one platform because they interacted with national networks and regional systems over which the state IT group had no control. The system also needed to accommodate a range of user experience levels and platforms, ranging from a smartphone to an ancient library system. Despite all this, the system was required to meet substantial reporting requirements as well as high performance metrics. Otherwise, the agency would face significant federal fines and a media feeding frenzy that may result in political damage across the board.

There were many things that could go wrong, yet the team had little insight into which things had failed. When systems went down, or citizens got stuck, a meeting would be called with five groups in a room with a white board, going around and around in endless speculation as they tried to find out what happened, randomly trying fixes, rebooting machines, losing sleep and pointing fingers while the clock ticked and clients couldn’t get online. Something had to change, but no one knew where to begin.

**Enter Splunk**

Several key team members attended a government conference where they learned about the Splunk platform. They tried Splunk software to see what it could do for them, and within four hours of deployment they got their first meaningful insight, solving a problem that otherwise would have required the multi-group meeting. At that point, the agency couldn’t wait to purchase Splunk software and apply it to other problems.

With Splunk, the agency can ingest multiple types of data without needing to re-architect how it fits together with other systems. IT now can manage and monitor a wide range of data sets, searching through web, middleware, network, operating system and other types of data and events with one lens. They can see how everything fits together.
Here are some examples of the log types that the agency ingests into Splunk software:

- Specialized application logs – 20+ different types
- Physical server (Linux, Solaris) logs
- Proprietary system logs
- Middleware (IBM WebSphere) logs
- Apache Web Server logs

Data is gathered from the system level (CPU, network, etc.) to application level (WebSphere) to end user experience (Web Server logs) for a true end-to-end view. The Splunk architecture for the initial deployment is shown below (See Figures 1 and 2).

When the state agency started using Splunk software, it indexed 100GB a day. Within a year, Splunk software proved to be so critical that it began ingesting an additional 500GB. At the time of this writing, the customer owns a 2TB Splunk Enterprise license.

**Stating the True Problem**

The teams within the portal group include specialists in WebSphere, network, user interface management, application development and Oracle. Before the group deployed Splunk software, employees would attend long troubleshooting meetings to determine what needed to be fixed and who needed to do it. Since Splunk software was deployed, it is far easier to see what issues are occurring and to determine whom to pull in. Even better, Splunk software provides insight into specific user interface problems.

Before Splunk, tracking a specific incident was impossible—logs from different systems took days, and with more than one million users, it simply wasn’t feasible. But since Splunk software was deployed, a user can report a problem and an analyst can track what happens across the systems by using his or her ID (see Figure 3). Figure 3 points to a navigation issue, which, upon further drill down, shows a discrepancy in the application based on how the user accessed it.

When the application developers learned about this error, they did further research and discovered that it happened to users accessing the portal from a specific platform and operating system. They also found out that this had happened since a small code change the day before. They knew which team member had written this code, and she was able to fix this issue within minutes. This avoided hundreds of calls per day and many hours of frustration for both users and support staff.

Because of Splunk software, the team responds agilely to the first harbinger of difficulty. What’s more, team members who have nothing to do with a problem do not waste their time in fruitless meetings.
Splunk software also enables the team to troubleshoot errors according to the host and the time accessed. This dashboard suggests peak times might be the cause for higher error rates. But what if higher error rates are simply a result of more people accessing at a specific time? Using Splunk Enterprise pivots and search tools, the team discovered the rate of errors could be greatly reduced by shifting the timing of maintenance and processes to times when systems were sparsely used (see Figure 4).

How were errors distributed according to time of day and host? Figure 4 shows this visually; figure 5 shows a drill down at a specific time by error type.

After learning about all the errors that were occurring, the team created other searches to get more insight. Two dashboards capture what happened after a design modification (See Figures 5 and 6). The support staff was getting overloaded with frustrated phone calls, but until they could learn more about the errors, the IT team was in the dark. In the case shown, there were many 404 errors clustered at specific times. Further drill down revealed that the end users had clicked a link on the website but the link did not go anywhere. Continuing to drill down revealed the page where this occurred, and even the button that did not work. By fixing a typo in the link that was called by the button, the 404 errors caused by the broken button no longer occurred.
In order to assign information technology staff appropriately, management wanted to know which devices clients used most often (See Figure 8). Also, knowing the platforms used helps to illustrate the complexity of the user environment when asking for more budget for specific features. Moreover, business analysts wanted to know which platforms users tend to log in with, according to the benefit type they require. The answer to this question also helped management allocate specialists to the appropriate teams.

Using Splunk Enterprise to Plan Maintenance and Allocate Resources

This state agency employed more than 5,000 employees and many additional contractors, with a significant number working in information technology. They also had hundreds of machines to maintain. By using the Splunk platform, the agency is able to determine the best time to do maintenance and the best developers to hire or allocate to specific projects. The dashboard reveals the time of day when users are most likely to log in. Unsurprisingly, this is lunchtime. Still, many users log on into the late hours of the evening. The dashboard below, along with a similar dashboard that tracks weekend access, helps the team plan maintenance (See Figure 7).

However, 404 errors can be caused by multiple factors. Other dashboards showed 404 errors organized by city. This dashboard is very useful when calls flood the system. In the past, this type of problem would have caused staff to lose a great deal of time trying to repair issues that had nothing to do with the system. But now, the dashboard enables the staff to determine if there is a preponderance of outages in a specific area, rather than the state benefits system. The system also contains a mapped dashboard that can be used for geographical troubleshooting.

Answering User Experience Questions, One Dashboard at a Time

Once the team members saw how Splunk software provided the visibility needed to troubleshoot effectively, they decided to look at what else they could learn from Splunk dashboards. They put their attention toward studying how users interacted with the system, what they were trying to accomplish, and where they dropped off. For example, they built a dashboard to see which documents users accessed within the previous hour (See Figure 9). This dashboard can be analyzed to see if there are confusing issues that need to be elucidated, help docs that are more frequently accessed, and docs that haven’t been touched in years and can be archived from the website.
Pinpointing Non-Technical Causes of User Failure

Sometimes it seems as if the cause of user failure is a technical issue, but later research determines it is something else. As the pie chart shown in Figure 10 illustrates, during the application process for one of the benefits administered by the state, almost one-fourth of users gave up and left the site after trying to submit their applications. This was a serious issue that not only indicated that some citizens were not receiving needed benefits, but also could invoke federal fines. Troubleshooters wondered why?

They developed a Splunk search to find out the point at which users tended to leave the system; most users who dropped the application process abandoned the program in the first page (See Figure 11). However, there were no technical errors discovered on the first page. A bit of targeted user testing determined the reason: users were not aware of all of the documentation they needed before starting the process. This was a user interface issue, not a programming issue, and it was easy to solve by providing appropriate content. Detailed directions were created with a checklist to help users gather the appropriate information before starting this process. This resulted in several benefits:

- Less user frustration and less time filling out forms
- Lower computing resources needed as the process sped up significantly
- Fewer applications saved mid-session
- Fewer fraudulent applications
- Reports to federal regulators showed much higher rates of success, preventing fines

Figure 9. Docs accessed within previous hour.

Figure 10. User failure rates.

Figure 11. User drop off rate.
These are just a few examples of the types of insights that the Splunk platform revealed about end users. Here are examples of other reports and dashboards the teams created:

1. Track user activity by region: logins onsite in regional office correlated with similar accounts logging in offsite.

2. Report on statewide benchmarks, discovering the average amount recovered and saved per completed Medicaid provider investigation.

3. Report the number of state citizens using call centers versus the internet to apply for Medicaid, SNAP benefits and other state services.

4. Compute the average daily caseload for child or adult protective services.

5. Optimize the prevention, detection and correction of various fraud, waste and abuse focusing on high risk areas.


7. What keywords led users to the benefits portal? (Charted by domain, search phrase, keyword).

With Splunk Software, Needs Met
This use case illustrates how Splunk software ties together a complex set of systems and functions, providing insight into processes and users. The citizens of this state can finally manage their benefits with a reliable, web-based portal. Developers identify performance bottlenecks before rolling iterations of the portal into production. Production portal staff can spot end user performance and availability issues before they affect a majority of users, and use Splunk to identify the root cause. With Splunk, the agency realized:

• **Collect and Index Data.** Because Splunk software indexes all types of data, regardless of format or location, the customer does not have to manually pore through data to find out where a client’s request fell through the cracks, and customer support has end-to-end visibility on a customer’s processes.

• **Correlate and analyze.** Because Splunk correlates different types of data, the customer is able to see patterns in how clients use services, revealing opportunities for streamlining and improving service.

• **Visualize and report.** Because Splunk enables customers to visualize trends and characteristics in custom dashboards suited to any business, operational or security need, developers were able to come up with new approaches for unique situations.

<table>
<thead>
<tr>
<th>Industry</th>
<th>Splunk Use Cases</th>
<th>Splunk Products</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Public Sector</td>
<td>• Application Delivery</td>
<td>Splunk Enterprise</td>
<td>• Specialized application logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Physical server (Linux, Solaris) logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Proprietary system logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middleware (WebSphere) logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Apache Web Server logs</td>
</tr>
</tbody>
</table>

**About Splunk**
Splunk Inc. (NASDAQ: SPLK) is the pioneer in analyzing machine data to deliver Operational Intelligence for security, IT and the business. Splunk provides the enterprise machine data fabric that drives digital transformation. More than 12,000 customers in over 110 countries use Splunk in the cloud and on-premises. Join millions of passionate users by trying a free trial of Splunk.