

White Paper:

Service Level Management: Creating a Structured Process to Improve Availability



Edward Giebel,
Director of Solutions Development
CDI IT Solutions
602-508-6456

Service Level Management

Imagine the following situation: A mission critical system went down at the worst possible time and now the business is going to miss their target. The executives are all angry because this outage negatively impacts their bonuses and the warehouse is mad because the overtime they had been counting on is not needed. This type of event would cause any IT manager to feel as if their job is on the line.

Many IT managers dread the day when a system fails at the most inopportune time. Never mind if they had been flawlessly operating for the past two years; everyone remembers the last big failure. The time the IT team with the projects no one really understands let the company down.

The first line of defense IT teams use to increase application availability and mitigate this problem is infrastructure. They approach this problem with the perspective of an engineer – if they plan for failure and engineer around it, then they can mitigate the problem. This approach solves for one element that can cause a failure; namely the components. At the same time, it adds complexity into the environment, requiring ever more attention to detail to make sure those complex components are all working in tandem. If the IT team misses a detail, the resulting problem could be more difficult to diagnose and lead to an even bigger outage.

Service Management as a Path to Higher Availability

As demand for high-value IT solutions continues to grow in organizations, a high level of availability is expected for all computer systems. The IT department is faced with the challenge of reducing or eliminating unplanned downtime.

Achieving a high level of availability is difficult for even the best IT organizations, often unplanned occurrences create issues that result in server downtime. Five 9's or 99.999% uptime is the holy grail of IT service availability. Based on 24 hours of operation 365 days of the year; it translates to slightly more than 5 minutes of unplanned downtime a year. IT executives must have a plan in place to manage the level of service they provide to their organization. Without service level management, application availability suffers, putting IT at odds with their customers.

In an environment built for high availability, application outages typically occur when communication breaks down, leaving the IT team scrambling to recover from an action someone consciously performed. By implementing process to manage service levels, IT executives build a level of trust with their business counterparts. Business executives look for reliable partners, they expect no less from their internal organizations.

“As demand for high-value IT solutions continues to grow in organizations, a high level of availability is expected for all computer systems.”

Growth of the IT Department

When an organization has just one IT person, that individual will have a clear understanding of what they do on a day-to-day basis. Once IT growth develops into double digit staffs and tasks are divided along functional areas, the ability to effectively communicate is often effected. The problem is not people talking to one another, but rather the challenge of those in the department realizing the impact of their actions on everyone else. Few people last long in an IT shop if they display a cavalier attitude about the impact of their actions. Indeed, outages are often the result of someone actively deciding their task does not impact someone else in the organization.

For instance, network administrators often need to get patches into place quickly to eliminate vulnerabilities in the network once they are identified in public. But what happens when the application developers, either purposefully or unintentionally, have exploited the vulnerability being patched in their code? With luck, an outage occurs and the team quickly correlates the event – a change to the network – to the result – an application outage. However, often a less desirable result occurs. For example, a degradation in service is experienced and it is difficult to quickly correlate the event with the outcome, resulting in a great deal of time being spent looking at other possibilities for the change in application performance. Or an intermittent outage occurs and an IT manager must first determine the conditions that

created the outage before they can begin the process of determining the root cause of the outage. It is therefore clear that the better an organization is with up-front communication through use of a structured process, the less time is wasted diagnosing unplanned outages.

So how is the right amount of process determined? The best approach is to start with enough process to address all IT needs without significantly impacting the output of the team. The goal is to improve the quality of the operation, eliminating unplanned downtime. This is met when the target for unplanned downtime occurs over a statistically long enough timeframe.

“The best approach is to start with enough process to address all IT needs without significantly impacting the output of the team.”

Determining the Right Amount of Process

In many cases, the lack of IT process discipline is not a planned event. It occurs because simple processes that worked well for the organization 5, 10 or 20 years ago no longer fit due to growth, a change in business, new reliance on IT as a business differentiator or even a change in technology. Managers do not set out with the goal of having little to no process; however, it is often the result of a single minded focus towards completing tasks. Lost in the day to day operations is the time and effort spent resolving issues that proper process would avoid.

In addition, many IT organizations abhor process. Common excuses for not creating process include that the change is too small to require a process, the job is too complicated to be understood by other IT managers or it slows down service to customers. This resistance will result in attempts to circumvent new process tasks. Developers will complain about their inability to troubleshoot because they do not have access to the production systems. IT staff will avoid closing trouble tickets with meaningful root-cause information. Changes will be classified incorrectly to speed up approval and minimize review, not with malicious purpose, but because the change request was not submitted on time.

In the end, the job of the IT manager is to meet the service level commitments to the business. Process is a tool to support the manager in meeting this objective. The specifics of that tool are also very different when moving from a large to a small organization or from a shop with a single mainframe to a shop with 500 servers of every make and model.

The Hidden Costs of Downtime

Often, the IT department will claim that the system went down overnight, but the team restored service by 7:00am so there was no impact to the customer. Unless the person making this statement had a 24-hour operations staff that was able to restore the operation without calling anyone, there was an impact. Maybe the impact wasn't in a lost order, it wasn't in a visibly upset customer, and it likely wasn't even in bad publicity. But if it woke up a member of the IT staff and pulled them away from their families, the result is a productivity hit in whatever work they were scheduled to accomplish the next day.

Such a hit is likely negligible when it happens once a month or less, but once it becomes a three-times-a-week occurrence, project schedules will slip as the midnight owls miss commitments. Humans need uninterrupted sleep to function at their best, and troubleshooting problems, even if it only takes a short time, frequently cuts into this important need. While they may log time on the daily task, chances are it was not completed to the same level as if they had not been disturbed.

With enough of these unplanned wake-up calls, staff turnover begins. People start out believing they are critical to the business success when they get woken up two or three nights a week, but it eventually grows old when they realize they do not control their own life. Turnover begets new problems, as fewer knowledgeable staff members struggle to keep up with the problems.

Where to Start

The first step on the road to improvement is recognizing the problem. Perhaps it has yet to reach the critical stage, but the first signs of trouble are developing. No matter how minor or major the problems, they never get better without specific action to solve them. Holding a meeting without taking action on the outcome won't solve a growing availability problem. An IT manager should clearly identify the issue before starting to treat symptoms. Even if a process is functioning well, it should be reviewed as thoroughly as problem areas. The proper processes must be put in place to prevent outages when possible and at least quickly diagnose and solve them when they do happen.

The next step is to conduct an objective assessment. Start by taking a specific function and really understanding the process. Map each of the tasks involved. Identify which tasks are value adding, and which are non-value adding. Within the non-value adding tasks, determine which are process tasks and which are compensating tasks. Process tasks may be necessary, but compensating tasks are never good. As an example, completing a change control request is a process task. It may be seen as a pain, but without a common set of

information on a change request, it is difficult to evaluate impact. On the other hand, holding a change meeting simply because people do not take the time to read and approve changes online is a compensating task that should be eliminated.

The IT Manager can then determine where compensating tasks exist in the current process and decide what should be automated. Organizational inertia should not prevent fresh thinking and development of new suggestions for change. In addition, just because the majority of the team considers that a process works fine does not mean that the perspective of the dissenters should be ignored.

Establishing Key Pain Points and Setting Realistic Expectations

The first problem to resolve should be the biggest pain point. If 6 out of 10 times a change to the production system causes an outage, that would be a good place to begin. If there is a continuous problem with the file systems filling up, then that should be addressed first. Organizations have evolved in different ways over time. In some cases the problem is the result of a process issue upstream from where it manifests itself. More often than not, the problem is seen where it began, and should be addressed accordingly.

Fixing or implementing service level process should be approached as a series of intermediate goals. Trying to go from ad hoc problem reporting with developers moving code into production to a fully Information Technology Infrastructure Library (ITIL®) compliant process in a single leap has a high chance of failure. Trying to implement ITIL compliant Incident and Problem Management processes, before moving onto to Change Management and then to Release Management, has a much higher probability of success.

However, it is possible to be successful starting with Change or Release Management. If the nature of the problems is understood, or the problems are mainly related to issues with new code put into production, then Change Management is a viable place to begin. Picking an area that is not the source of a problem is not likely going to show much in the way of results, and worse, could disillusion the IT staff on the very idea of better process management.

In addition, it is important to remember that software is just a tool. There are many good software packages available to help with service level management, and today most of them follow a recognized service management framework. But software, no matter how good, is just a tool. It is important to invest in the corresponding training as well, as most IT organizations can benefit from learning the proper methods from a professional.

“The organization that successfully tackles service level management is one that realizes this is not a “project” with a start date and an end date.”

Also, the software acquired should match current maturity levels. If there is only time to implement Change Management, it is important not to acquire a package that forces implementation of Incident and Problem Management to obtain Change Management. The efforts needed to compensate for the process imposed by the software should not be underestimated – this could lead to an implementation failure.

Continuous Process Improvement

The organization that successfully tackles service level management is one that realizes this is not a “project” with a start date and an end date. In many instances, it is a cultural revolution. The focus shifts from what drives many IT managers (how much can be delivered) to a business centric focus (what’s the error rate). With the right implementation, the second metric is positively impacted and, over time, so is the first. People aren’t spending as much time diagnosing and fixing the same issues (problem management), participating in those annoying “what changed last night” con-calls (change management), or analyzing the difference between Node A, where the application works, and Node B, where it does not (release management).

Once the process is established, success should be celebrated. Then a review should be scheduled for an appropriate time in the future. Just as a business model isn’t static, an IT model undergoes continuous change. Outsourcing, server and storage virtualization, SaaS, and SOA are just the start of the list of changes going on in the typical IT model. If a business is growing, moving to the web, expanding to multiple locations, adding work from home options or any number of other factors, then the IT processes need to adapt.

It is important to recognize the need to change with the business. The processes built should have the flexibility to assess differing business risks and treat them differently. A new system with few users running in its own environment is different than a change to the production MRP system. While it is imperative to understand any downstream affects a new system may have, the risks associated with changes to an existing production system are far greater.

At times in an organization, things may need to move faster, with fewer checks, to meet business goals. It could be due to an acquisition that needs to close, or the creation of a new product or line of business to meet a competitive threat, or even a new regulation that the company needs to meet. The reasons are many, but there are times the due date is the most important thing. Nonetheless, ensure the process

design does not allow people to abdicate responsibility simply because something has to happen by a certain date. They might not have two days to think about something, but it does not mean they do not have to think.

"If a business is growing, moving to the web, expanding to multiple locations, adding work from home options or any number of other factors, then the IT processes need to adapt."

Timely Problem Management

When a process begins to not work, it needs to be addressed quickly. Few things turn off employees more than management ignoring things that are obviously not working. Often, it is possible that it is not the process that is not working, but the people implementing it. The process should not be tweaked just to appease people's reluctance to follow it, but should be revised if non-value add compensating activities are occurring.

The end result should be considered when evaluating the success of an implementation. Even if the process is being followed and no one is complaining, are the planned results being obtained? If one of the goals was to reduce unplanned downtime, has it actually gone down? If not, is it a result of new factors, or has the process implementation not had the desired results? If the results are not what was expected, is it due to one of the factors discussed previously? Or have new documentation steps just been added to a process that was not working in the hopes of getting different results?

Progress Management

Progress can be measured in a number of ways. If implementation date against project plan is measured, a great deal of bad code will be pushed into production where it is then fixed. If the number of calls a help desk agent handles is measured, many calls will be taken, but only because people have to call back multiple times to get their problem solved.

One important gauge of progress on the road to implementing process is if tangible results are seen against the goal. Are system outages less frequent and more easily solved? Is a knowledge base in the ticketing system beginning to be built because IT staff are providing meaningful explanations in the resolution? Are there fewer problems from changes to the environment? There is no one gauge, because everyone starts from a different place. Progress will depend on the starting point as well as the maturity of the organization.

It is important to implement meaningful measurements that can be accurately monitored and reported. Metrics gathering should be a simple, logical output of the process. If it takes a day to compile the monthly report, either the wrong thing is being measured or it is being measured in the wrong

way. In addition, measurements that put groups within an organization at odds with one another should not be implemented. No one benefits from the effort it takes to push the problem from one group to another.

Also, compensating controls should be established for measurement. If a business has an issue with the amount of time it takes to close a problem report, close time should be measured, but a measurement of reopened tickets needs to be added to make sure tickets aren't closed before the problem is really solved. It is also possible to come close to eliminating unplanned downtime at the cost of being so risk adverse as to paralyze the operation. The stated goals need to be achieved while not creating a negative effect that completely offsets the achievement.

Reducing Unplanned Downtime

In any discussion on availability, unplanned downtime is a key component. With proper architecture and solid operations, the five 9's of availability can be achieved. This might not be needed, or the architecture to reduce or eliminate single points of failure can be too expensive.

In addition, to measure unplanned downtime properly, an agreement on planned uptime should be established. What are the normal hours of operations for the organization? What are the planned windows of downtime? Many IT organizations fall into the trap of failing to have routine downtime planned in the schedule. Unless an IT shop can run 24 x 7 x 365, rolling changes into production without taking down applications, an IT manager must plan for downtime on a periodic basis - weekly, monthly or quarterly.

Once there is agreement on the normal operating hours for each application, measuring downtime is much easier. At least there is little disagreement on when the application was experiencing "unplanned" downtime. In order to be credible with business users, the measurements should be clear so as not to be easily disputed. If three servers must be rebooted in sequence, the outage isn't over until the last server is back online and running the application code, not when the first server starts rebooting – a typical IT focused measurement regarding when the solution was established and action took place.

However, the many common causes of application downtime can still be eliminated. A plan to test applications after changes should be established to measure not only changes to the applications, but any change that has even the remotest possibility of impact to an application. And testing is not simply checking the login screen. The major functions of the application must be tested, including connectivity to other systems and both reading and writing the database. The problem with not doing these tests is in the heat of the battle,

when the application is down and everyone wants to know when it will be back up, people start typing before they start thinking. And the 30 minute outage that would have occurred for a system restart turns into an all day outage while a disk drive is rebuilt.

After a plan to test changes is established, it is best to determine what will happen in the event that the test fails. How much time is there to diagnose the problem before the rollback plan must be implemented? What is the rollback plan, if any, and is there confidence in how long it will take and if it will work? Who owns the decision on when to start the rollback?

The final piece of the plan is what to do in the event a problem is discovered after the system is back in production. A number of factors go into such a decision, including the impact of the problem – is 10% of the application functionality impacted or 50% - as well as the time it will take to fix the problem. Is it a minor problem to someone who tends to blow things out of proportion or a major problem to someone who tends to understate the magnitude of the problem? Considering the possibility of a problem happening during production in advance minimizes the chances of a knee-jerk reaction once a problem does occur.

Faster Restore Time for Improved Customer Satisfaction

The faster normal operations can be restored, the better for the organization. Not surprisingly, the more executives that are hovered over the system administrator's cubicle, the more compelled they are to start typing. It is wise to understand the problem before attempting to fix it, or often the system administrator will have to chase the problem around as they fix each symptom, popping up problems somewhere else. Customer satisfaction is a good measurement of how users perceive the job of the IT group. However, it is highly subjective and easier to negatively impact than positively impact. For this reason, customer satisfaction as a measurement must be taken at regular intervals that take into account typical business cycles. The previous period and year should be compared to gauge improvement. Did IT do a better job this season than last? Is the trend each quarter moving up?

In looking at customer satisfaction regarding downtime, there needs to be a balance between the two ends of the spectrum – many new features with little stability and a very stable system with little innovation or change. The right place on that line varies with each organization, and even within different business units within a single organization. Process around service management should accommodate both ends of that spectrum, and the points in between.

“With careful development, implementation and progress management, a service level program can be created that will ensure the highest level of availability possible, leading to greater long term productivity and better reliability for the consumers of IT systems.”

Conclusion

The road to implementing good service level management has numerous potholes and other hazards. To keep people motivated, it is important to realize when even moderate success is achieved. All successes should be celebrated and efforts published to the business customer. Tools that show ongoing success need to be established, such as trend lines, so even when a metric backslides on a month to month basis, there is a positive trend.

The demonstrated success is a great selling tool to customers. They will understand that these successes will improve their user experience. This is the ultimate goal of the service level management process. With careful development, implementation and progress management, a service level program can be created that will ensure the highest level of availability possible, leading to greater long term productivity and better reliability for the consumers of IT systems.