Problem management for reliable online services

April 2013

Contents

Overview

More and more organizations are using problem management to help them achieve greater reliability, agility, and efficiency in their operations. With the increasing use of the cloud and the complexity of cloud and online services, problem management is even more important. Organizations that provide online and cloud services face a key challenge to put in place functions and processes that can detect, diagnose, and resolve existing or potential problems that threaten their customers' ability to do business. Doing just that, though, is key to maintaining online services that meet the marketplace's expectations for performance, availability, and quality of service.

Cloud service providers often use a combination of virtualized and online service environments that run high-density workloads on multi-tenant architectures. Large-scale service environments are inherently complex because of their size. Such environments have a plethora of disparate components and dependencies, which can increase the number of service related failures. In addition, as organizations (whether they are the cloud provider or the customer) scale up their operations, they can come to employ hundreds or even thousands of on-premise and public physical and virtual servers and their corresponding configuration items. In such circumstances, tackling failure in a strictly reactive way is insufficient. To deliver a competitive service that can please customers, service providers must examine possible failure conditions proactively and holistically before those failures happen.

To manage their efforts, service providers committed to live site excellence should strive to put in place a problem management function and process that not only determines the root cause of incidents to prevent their recurrence, but continually monitors configuration, capacity, availability, and performance to forestall service problems from arising in the first place. Such problem management processes can lead to predictable and repeatable levels of operational performance.

Problem management can be used by a wide range of organizations to help improve the reliability of their online services – from a cloud provider providing infrastructure, platform or software solutions, though to an independent software vendor (ISV), or the end customer. This paper introduces problem management and the benefits organizations may derive from implementing a robust problem management framework. It frames problem management by comparing it to incident management and describes the difference between the two. It also provides several foundational concepts of effective problem management processes, and discusses examples of how these types of organizations might implement problem management, including root cause analysis, to help improve the reliability of their online services.

The benefits of problem management

Organizations that provide online services can greatly benefit from establishing a problem management function or problem management processes. Particularly, those organizations may be better able to meet their customers' service expectations, more ably establish and meet their service level agreements (SLAs), and keep their engineering resources focused on providing features that delight their customers. Problem management supports these high-level benefits by:

- Eradicating underlying problems that can cause the same issues to occur repeatedly.
- Reducing the number of incidents that cause service outages or compromise service performance.
- Reducing the costs related to incident response.
- Enabling organizations to resolve outages more quickly.
- Enabling higher productivity for both the business and IT.

When a service provider better understands the benefits of problem management, they can better analyze whether their processes address their business needs and their customers' best interests.

Problem management versus incident management

A common problem for many service providers is that they confuse incident management with problem management. They conflate their robust incident management processes with implementing problem management functions and processes. This confusion can hinder an organization's agility and limit its ability to optimize engineering resources to address the underlying causes of common incident types.

To clear up this confusion, the essential characteristics of incident management are to respond to warnings and restore services as quickly as possible. Primarily a reactive process, incident management is typically focused on a specific component.

Problem management is a proactive process that takes a more holistic view of the entire service, its components and dependencies. When an organization implements problem management, the end-goal is to investigate why each outage incident occurred and correlate that information with data gathered about other incidents and activities. The primary goal of problem management is to reduce service outages and, therefore, help to mitigate their impacts on the business. To that end, problem management works to help:

- Prevent problems and similar classes of problems from causing incidents in the future.
- Management teams to allocate engineering and IT resources based on problem impact and cost.

Because incidents that affect an online service can occur frequently and because service restoration from incidents that threaten service availability are urgent, staff who perform incident management typically do not have the time required to perform effective problem management. This is why it is important to have a team and processes that support problem management that are separate from the incident management teams.

As an analogy, fire fighters have the immediate goal of putting out the fire (incident) as quickly as possible, rather than determining the cause of the fire (problem). Fire inspectors investigate why a fire occurred—whether, for example, the root cause was human error, faulty wiring, or malfunctioning heating equipment. Similarly, online services problem management staff investigate the root cause of incidents, whether human error, faulty incident response procedures, or hardware failure.

Moreover, the fire inspector can help prevent fires from occurring by examining buildings for potential hazards and recommending corrective actions. The fire inspector could determine that a certain building material is more vulnerable to catching fire and that these incidents often happen with buildings located near the explosion of fireworks during New Year's Eve celebrations. As a result, they might recommend either the application of fireproofing to the building material, the banning of fireworks near such buildings, and extra vigilance during New Year's Eve celebrations.

Fire inspectors also have access to data from many fire departments and this gives them the perspective to understand the causes of fires more broadly. Similarly, online services problem management staff has access to incident and monitoring data from all components and features that make up the service and this provides them with the knowledge to better understand the impact of types of problems across the service.

As a further example, compare the following scenarios.

Scenario 1
On a Monday afternoon the service monitoring system in a datacenter operations center raises a problem alert against component 1 on server 1. The incident management staff opens an incident ticket and consults the standard operating procedures (SOP). The SOP states that when this condition occurs the proper corrective action is to reboot the server, so staff reboots server 1, the problem goes away, and staff closes the ticket. The next week, again on a Monday afternoon, the monitoring system raises the same alert against component 2 on server 2. Again, staff consults the SOP and this time reboots server 2, and the problem goes away. These incidents continue to occur across different components and servers in the datacenter but always on Monday afternoons and always with the same alert. The incident management staff always follows the SOP and reboots the errant server and the issue goes away. Each time this event happens a subset of customers experience a service disruption. Additionally, the incident negatively impacts the services quality metrics that the team monitors and is trying to improve. Both of these will continue to occur if the incident management team continues to follow SOP.

Scenario 2
For three weeks, the service monitoring system triggers the same alerts as in Scenario 1, and incident management staff reboots servers as prescribed in the SOP. However, the problem management staff notices that the system has raised the same alert across the datacenter on Monday afternoons for the past three weeks. They enable the collection of targeted data about the incident. Over the next few weeks, the problem management team researches the alert to determine why it is being triggered on Monday afternoons. The team uses telemetry from the live site to determine what factors may be causing the recurring incident and correlates it with other similar incidents in the service. After it determines the cause of the incident and other incidents that are associated with it, the problem management team proposes proactive steps to optimize service performance, availability, and Quality of Service (QoS) in the long term. After these proposals are implemented, the recurring service interruptions stop and the service quality metric improves to provide a better customer experience.

Comparing these two scenarios prompts the question, what do problem management teams do specifically to correlate incidents across an online service environment? First, they mine forensic incident data for high-impact incidents and look for ways to reduce the mean time to resolve (MTTR) for those incidents and ways to manage and reduce unplanned events. Problem management provides leadership with data about where to allocate limited engineering resources for the best return on investments in reliability. It also collects trend data by ongoing monitoring of datacenter health, such as service level quality and capacity usage. Proactive data collection across production helps to identify and manage configuration drift between as-is and released standards for operating systems, database platforms, and virtual infrastructure, storage, and network.

Organizations that implement effective problem management can make incident management and the processes it uses more effective and efficient as well. This is because problem management's goal is to reduce the underlying causes of incidents through workarounds or by suggesting services fixes to the engineering team. Doing this can also reduce the incidents themselves, lessening the workload of the incident management team.

Implementing problem management

Implementing formal problem management that is separate from incident management becomes more and more critical as online service providers and ISVs scale up their operations by deploying and maintaining an online services infrastructure and platform. For example, a company that plans to offer a suite of services begins by offering one. While they are only offering the single service, there are only a small number of critical incidents across the service. In these early stages, incident management alone may meet the company's needs to maintain the service's availability, performance, and QoS. However, as the company scales to offer more services and increase the features of the existing one, the number of major incidents increases due to the proliferation of dependencies, components and configuration items, and the increasing complexities of the service environment. Along with this, the negative impact of service disruptions grows in parallel with the increased scale. It no longer makes sense from a business perspective to expect an incident management team to be able to fix more incidents, while also providing a proactive role in identifying systemic root causes of those incidents.

As mentioned previously, many organizations believe they perform effective problem management when they actually perform incident management. This section details some of the prerequisites for implementing fully functional problem management and provides insights into how to address problem management for hybrid cloud service environments.

Ingredients for effective problem management

Problem management is a specialized function and in most organizations requires sponsorship from senior business leadership and an organizational recognition of its business value. The problem management team may sit in the operations or engineering teams, or in a separate team altogether but regardless of their placement in the organization team members must have neutrality and impartiality to be successful. Additionally, each sub-group of the engineering team should be key participants in solving problems identified by the Problem Management team. This includes planning for upcoming versions of service offerings. Additionally, the problem management team must provide input to and gather input from the operations team. Problem management teams should consult and partner with the incident management team so that both groups understand their respective roles. Finally, problem management must report to business leadership, operations management, and service delivery process improvement to ensure that those stakeholders understand current issues and incorporate problem management when adding new service offerings or scaling up existing services.

Mature organizations that want to perform effective problem management should have automated systems in place that can capture sufficient metadata and can accurately and rigorously collect the details on each incident in the service environment. Less mature organizations that wish to provide large-scale online services should plan to build those systems, if they do not already have them in place. The details captured by these systems can include incident severity, priority, when the incident started, when it was detected, how it was detected, time to mitigation, time to recovery, customer impact, and many others. This type of rich data collection and the ability to query, and analyze the data is imperative to establish incident patterns. Beyond the forensic information provided by incident data, an organization must also use historical data about the service to compare against the forensic data.

It’s important for service providers and ISVs to have a good understanding of all components and dependencies that make up their service environment and ensure they capture the right data to effectively perform problem management. In addition to collecting the data, these organizations must also agree upon which tools and methodologies they will use to perform problem management, which establishes a common taxonomy and set of metrics for analysis. Organizations that share common dependencies may choose to align on tools and methodologies, as well, so that their platform investment is more efficient and so that they have a common vocabulary for classifying problems. This enables them to perform data analysis and problem correlations smoothly across service boundaries.

Problem management and public online services

Understanding all components of the service environment is particularly important when an organization implements a hybrid cloud model to provide services to its customers. In this model, internal online applications or services coordinate with those delivered from public online service providers. Such an environment frequently has layers of complex dependencies between the private online services and the public online services.

Any organization, be it a cloud service provider or a cloud service provider's customer, which has an effective problem management must ensure that any provider they partner with extends their problem management capabilities rather than hinder them. The additional provider should surface any and all information that will help its clients maintain good problem management. Additionally, it is important for the client organization to understand the additional provider's processes and how it augments and enhances the organizations own. Some questions to consider can include:

- What best practices that support problem management does the additional service provider suggest to customers?
- Does the additional service provider have processes in place to diagnose problems and identify root causes?
- Does the additional service provider actively correlate trend and historical data to find relationships between outages?
- What systems does the additional service provider have in place to gather information about incidents and the health of its service environment?
- What processes does the additional service provider have in place to analyze incidents and correlate them with other incidents and activities in the service environment?
- What metrics does the additional service provider use to measure the effectiveness of its problem management process?
- How does the additional service provider monitor the health of its IT operations to preempt outages and ensure services are delivered according to SLAs?

The problem management process

Along with having a system in place that is capable of capturing all significant data for a service or services, it's important to establish a process that enables the organization to respond to problems reactively and proactively. The process and its methodology can vary from one organization to another, but each organization must define the steps that they will include.

The following figure captures such a problem management process, as defined by the Information Technology Infrastructure Library (ITIL).

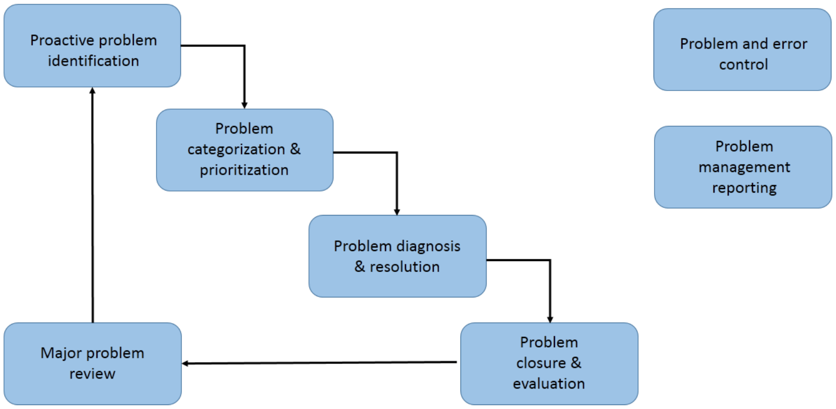
Proactive problem identification
Problem categorization & prioritization
Problem diagnosis & resolution
Problem closure & evaluation
Major problem review
Problem and error control
Problem management reporting

Figure 1. A problem management process, as defined by the Information Technology Infrastructure Library (ITIL).

Proactive problem identification

The first step in problem management is to identify existing or potential problems. Part of this process can be to define what constitutes a defect in the online service environment and then to determine what issues and defects could potentially cause an incident that could affect the availability of one or more services. These defects can be said to pose a risk to the environment, and those with the highest risk would be given the highest priority to address. Having a process such as this in place can enable an organization to improve the availability of its services by proactively solving problems or providing suitable workarounds before incidents occur.

Problems can be identified in a number of ways. The most common way to identify problems is to correlate multiple incidents, or even outages, into a single, underlying problem. Some problems are identified by using alerts in monitoring software, using instrumentation within a service, and comparing current conditions in the service infrastructure against historical data. Frequently, the data provided by these techniques across the service environment are how problem management teams correlate seemingly unrelated incidents into a problem.

Problem categorization and prioritization

In this stage of the problem management process, the organization creates a problem record, which captures all details of the problem. This record can include an identifier for the problem, when the problem was detected, who owns the problem, a description of the problem, the users or business units affected by the problem, problem status and change history, and many other potential entries. The key is to include enough information to help the problem management team decide how to address the problem. After the record has been created, the team prioritizes the problem based on its severity and its impact on customers, among other factors.

This is also the stage in which the organization makes decisions about which engineering and IT resources to allocate to resolve the problem.

Problem and error control

The task of problem and error control recurs throughout the lifetime of the problem management process. In this stage, the problem management team constantly monitors outstanding problems with regards to their processing status. This enables the team to determine when it needs to apply fixes or workarounds to the problem and the errors it may cause before a workaround or full fix can be applied. For example, if a problem has been discovered in which a server hosting a particular component of a service uses excessive memory when under load, the team monitors the problem to make sure it doesn’t threaten availability or performance while the team is putting together short-term and long-term solutions.

Problem diagnosis and resolution

After the team has prioritized the problem and, the team works to identify the underlying root cause of the problem and initiate the most appropriate solution. Diagnosis includes root cause analysis.

Root cause analysis is a structured process for determining the actual cause of a problem as opposed to a perceived cause. For example, to arrive at the actual root-cause, problem management applies a systematic analysis to incident data using well – known methods such as fishbone diagrams or 5 whys analysis—a recursive questioning technique to identify the root causes.

The goal of this analysis is not only to identify root causes of problems, but to create information that is actionable to the resolution of the problems. To do this, the problem management team should classify problems into taxonomies. Doing so is helpful because it allows the team to create a standard vocabulary and framework for describing and sorting problems, revealing patterns and avenues for possible solutions. Establishing a taxonomy and applying it to recorded problems also allows the team to find patterns in the data at a later date. For example, an organization could classify root causes into three categories: people, processes, and technology, and subcategories within each of these. For people, was the cause inadequate training or human error? For process, did the problem result from a faulty standard operating procedures, Live Site guidance, or troubleshooting guide instructions? For technology, was the failure because of design or architecture, a bug, or configuration error?

Table 1 shows an example taxonomy with primary and secondary categories.

Primary category	Secondary category	Possible resource allocation
People	Human error – fat finger	Automate process, improve review process, provide training,
	Human error – customer asked for wrong thing	Improve documentation, provide training.
	Human error – malicious	Restrict and audit access to process to certified personnel or automation system.
	Human error – mis-configuration	Invest in automated configuration management system, monitoring/alerting on configuration errors; rollback to previously known good configuration.
Process	Communication – deployed wrong version	Improve review process, automation
Technology	Software – code failed to deploy or rollback	Improve deployment process, rollback capability
	Software – code caused failure (bug)	Improve testing, pre-production
	Software – maintenance failover failed	Regular testing of failover
	Hardware – datacenter power, cooling, etc.	Application fault tolerance and resiliency to infrastructure failures.
	Hardware – disk, CPU, memory, fan, etc.	Application fault tolerance and resiliency to infrastructure failures.

Table 1 Example taxonomy with primary and secondary categories

Ideally, when the problem management team determines the root cause of the problem and finds a solution, that solution should be applied to resolve the problem.

Problem closure and evaluation

The goal of problem closure and evaluation is to ensure that the records associated with a problem contain a full historical description of the problem, including steps taken to resolve it. Known error records associated with the problem should be updated as well to reflect that the problem has been closed.

However, there are some follow up tasks to perform after the problem is closed. It is important to update required activities to detect incidents that may be associated with the closed problem to make sure that the problem does not recur. Organizations can use key performance indicators (KPIs) to track the effectiveness of the process against service-level agreements (SLAs). These metrics could measure, for example, trends in the number of problems resolved by known errors, costs for problem resolution, percentage of problems with identified root causes, time to bring problems to closure, and the reduction of incidents.

A known error record is closed after the resolution plan or change request has been implemented successfully. Any correlated problem or incident records related to the known error are also closed.

Major problem review

The goals of major problem review is to determine how to prevent recurrence of the resolved problem and gather the lessons learned in resolution of the problem so that they can be incorporated into subsequent engineering and operations investments. Additionally, an organization can use this stage of the process to verify whether the problems marked as closed have actually been eliminated.

The review process should focus on what went well, what went badly, what could have been done differently, and what needs to be in place to avoid recurrence of the problem.

Additionally, major problem review provides an opportunity to improve the problem management process itself. When an organization reviews its problem management procedures it should assess:

- The effectiveness of problem management in reducing incidents.
- The relationship between increasing proactive interventions and reduction of reactive problem management.
- Any service quality improvements developed by the engineering team intended to help reduce risk of introducing new problems to the production environment.
- Proposals to improve problem management processes.

Problem management reporting

Problem management reporting is a recurring stage in the problem management process that an organization can use to keep stakeholders apprised of all outstanding problems, their current processing status and existing workarounds. Problem management teams should provide reports to the business management teams as appropriate as well as IT management teams. The team could also capture the lessons learned and share them with partners and dependent teams so that they too can learn from the resolved issues.

Approaches to problem management: two examples from Microsoft

With a better understanding of the general problem management process, it is important to understand how organizations that have successfully implemented problem management process have implemented them. This section looks at how two organizations within Microsoft, each with distinctively different customer needs and demands on their infrastructure— implement problem management. They are Microsoft IT and Bing.

Problem management as Microsoft IT

Microsoft IT operates and delivers shared infrastructure services for line-of-business application production and deploys and supports these infrastructure services. Their enterprise-level infrastructure has the following characteristics:

- Includes nine data centers.
- Provides services to 468 site locations across 112 countries.
- Supports 1,068 line-of-business applications.
- Includes 30,000 managed servers.
- Virtualizes 60% of the workloads they support.
- Includes 26 petabytes of storage capacity.

In this IT support model, the Problem Management function serves two key missions. The first is to identify, manage, and eradicate infrastructure risks and defects in the production datacenter and field locations. Second, it provides problem resolution and root cause services for the Global Service Desk, specifically the worst case high impact incidents termed as major incidents.

Within the Microsoft IT on-premise and public cloud environment, real-time and historical analysis of events, alerts, and incidents is a vital activity in understanding the "hot spots" within the environment. In this high density virtualized platform, recognizing the event trend deviations as quickly as possible to control a flurry of incidents all sourcing back to a common root-cause is critical to rapid and accurate resolution.

Two roles are defined for the team members. The Problem Analyst diagnoses problems and analyzes the root cause. The Problem Managers assess risk and work to eradicate the known error and ensure continued compliance.

Problem Managers hold weekly incident and proactive problem review meetings with the shared infrastructure support and hosting service owners. Issues discovered are those issues that incurred high impact or that pose a high risk of recurrence, high time to recovery, or have a potentially high revenue impact. These repair items are assigned to a designated service manager, engineering manager, or project managers.

Workflow management tools include Microsoft® Team Foundation Server and internal ticketing systems. Major Incident cases and remediation are tracked weekly and monthly with service operations and engineering teams leading into General Manager (GM) and Chief Information Officer (CIO) level visibility in a "Corrective Action Review" (CAR) process. This is to ensure the highest impact events have senior management visibility, priority and support.

At Microsoft IT, then, infrastructure problem management is a centralized function driven by a single team, with accountable shared service owners in the collective effort to enhance the user experience for application owners and end users. These service teams along with the leadership level CAR process create an effective model for Microsoft IT to solve problems in their infrastructure.

Problem management at Bing

Bing is Microsoft's public web search engine. Within the Bing organization, the Live Site Engineering (LSE) team is dedicated to understanding issues and risks that affect the Live Site—the production systems supporting the online services that Bing users interact with. LSE address those issues using problem management to:

- Conduct deep analysis of data from the live site.
- Identify opportunities for live site improvement.
- Work closely with feature teams to harden their services.
- Create product features and tools that improve live site stability.

LSE holds weekly incident and problem review meetings. A live site incident is any event in production that results in impact to customers. They are tracked in a custom incident tracking and response (ITR) tool. The majority of incidents are reviewed by both LSE and the feature teams to identify the root causes and next steps, which are also called actions. Key for the team is to separate the incident trigger from the root cause. To do this the team uses the iterative question-asking technique called the 5 Whys (referred to in the Problem diagnosis and resolution section on page nine of this paper). The '5' in the name derives from an empirical observation on the number of iterations typically required to identify the true root cause(s) of the incident.

Any root cause or vulnerability that is likely to occur in the future is assigned an action and the responsible feature team works on that action. Actions result in changes which are nearly always realized through software as opposed to human process. Making the changes part of the core code prevents mistakes, enables greater agility and results in a more reliable service.

Actions that can be completed quickly – typically within weeks – are tracked as Microsoft Visual Studio Team Foundation Server 2012 (TFS) work items. Actions that require significant investment are tracked as longer term problems. Examples include problems that require changes in service architecture, that require significant new features to better mitigate the problem, or that require adding capacity incrementally to the service.

After LSE identifies a problem, LSE creates a tracking item for the problem in TFS, a Microsoft application lifecycle management solution. Upon creation, LSE assigns the TFS item to a designated responsible individual (DRI), who is a member of the feature team responsible for addressing the problem. The DRIs are then the leaders of their team's efforts to resolve the problem.

Once assigned, LSE's role then becomes one of tracking the age of the problem in TFS and leading weekly and monthly meetings to discuss progress toward a decision on whether to make the required fixes, or after that, progress on those fixes.

While the problem management process is owned and centrally managed by LSE, the implementation of the problem fixes resides with the Bing feature teams. This distributed DRI model leaves the fixes with the teams who best know the systems that they work on. Each Bing feature team participates in the problem management process, and is accountable for addressing issues raised in that process. This leaves the LSE team free to take the larger view of the entire suite of Bing services, do the work of correlating incidents across services, and manage the problem management process.

Summary

Problem management has the dual goals of minimizing the adverse impact of errors within the IT infrastructure, and preventing the recurrence of incidents related to those errors and classes of those errors. To achieve these objectives, problem management analyzes the root cause of incidents and looks broadly across incidents and correlated business, engineering and operational events to correct or improve the production environment.

If implemented and managed effectively, problem management can provide numerous benefits, such as optimizing how an organization assigns engineering resources to address problems, improving delivery against service level agreements, and reducing operating expenses.

However, many organizations believe they have implemented problem management when in reality they may have only become better at managing incidents. These organizations do not fully execute or manage all facets of problem management and therefore do not realize its full benefits.

Perhaps the most valuable benefit of problem management lies in shifting an organization's focus from a solely reactive mode revolving around ticket volume and resolution times, to a more proactive role that analyzes patterns of incidents enables an organization to recognize more cohesive and strategic approaches to solving problems. This is especially crucial as an organization scales its online service offerings. As its production environment grow more and more complex with multiple dependencies and a growing number of components, the frequency of failure will dramatically increase. In such cases, organizations need problem management to help them manage for efficiency, agility, and reliability.

Additional resources

- Foundations of Trustworthy Computing: Reliability

  http://www.microsoft.com/reliability
- Microsoft Trustworthy Computing

  http://www.microsoft.com/twc
- Problem Management Service Management Function

  http://technet.microsoft.com/en-us/library/cc543264.aspx
- ITIL Website

  http://www.itil-officialsite.com/home/home.aspx
- IT Process Wiki: The ITIL Wiki

  http://wiki.en.it-processmaps.com/index.php/Main_Page

Authors and contributors

PETE APPLE – Microsoft IT

DAVID BILLS – Microsoft Trustworthy Computing

SEAN FOY – Microsoft Trustworthy Computing

MARGARET LI – Microsoft Trustworthy Computing

CHRIS McKULKA - Bing

SIAN SUTHERS – Microsoft Trustworthy Computing

JASON WESCOTT – Microsoft Trustworthy Computing

DONALD WILLIS – Global Foundation Services

SHAHBAZ YUSUF – Microsoft IT

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.