Using Our Understanding of z/OS Integrity to Enhance Audits

David Hayes – Auditor, Federal Government

21 February 2017

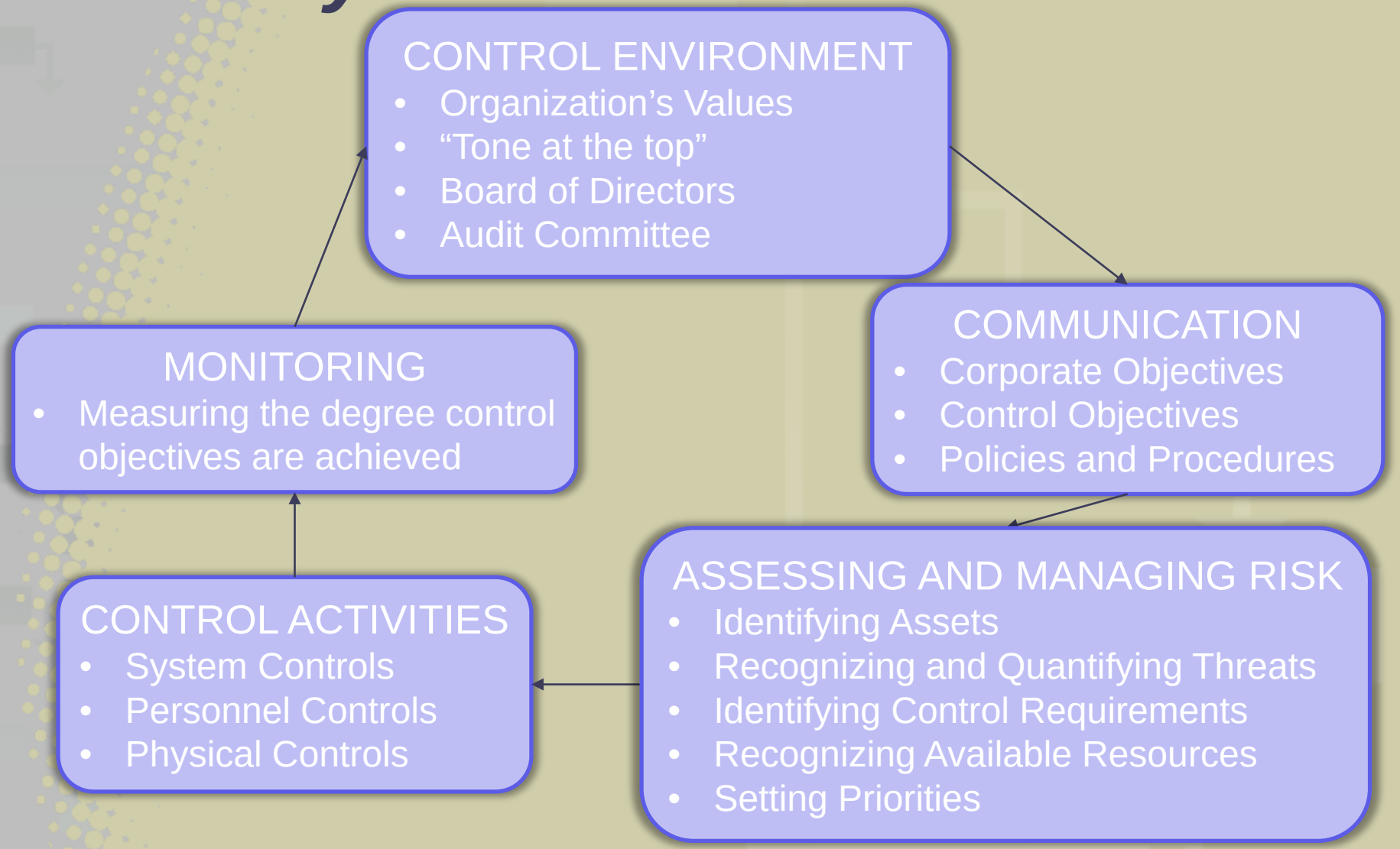
Introduction

- Objective
  - Use and expand on the topic and theme of Paul Robichaux's session on z/OS integrity to present audit considerations built on how z/OS integrity can be achieved (and compromised).
- Desired Outcomes
  - Move past a compliance mindset so control efforts (often thought of as audit preparation) can contribute to informing decision making
  - Help management and operations personnel understand the auditors' mindset in order to more effectively interact with audits
  - Identify where we are making assumptions

Agenda

- Understand the *organic* system of internal control
- Develop a better understanding of where and how audits (should) fit into an organization's structure
- Recognize the realities of how controls over z/OS platforms are perceived and the control assumptions often in place
- Leverage an understanding of the integrity achievable in z/OS environments to support or perform audits

The *System* of Internal Control



Achieving compliance with standards = Getting to a minimum level of performance

Audit Authority and Role

- All audits are conducted in accordance with the standards and requirements of an auditing standards body.
- Audits may be entirely based on evaluating compliance with one or more sets of standards or requirements.
- Most organizations are subject to independent audits of their financial reporting.
- Auditors have a responsibility to communicate their results to management and relevant stakeholders.

Audit Authority and Role (more)

- While communication between operational staff and auditors must occur to conduct audits (and get accurate results), caution should be exercised – too often operational staff perceive that auditors *require* things or changes *must* be made based on communications from auditors.
- Organizations should *never* rely on external audits to identify areas needing attention – external audits ARE NOT part of an organization's system of internal control.

z/OS Controls: Perceptions

- Applications, data and processing are all secure due to the external security managers
- Different workloads and functions are isolated from each other because they are in separate databases or LPARs or CICS regions…
- The control requirements other platforms are subject to don’t apply because mainframes are inherently secure

z/OS Controls: Realities

- z/OS integrity – at rest and in flight – is prerequisite for any representation or conclusions that controls *can* be reliable.
- The same level of control *can* be achieved with all three external security managers, but extra care is needed with two of them.
- Controls are almost always expressed in the context of boundaries. So, achieving a trusted computing base is predicated by sound boundary definitions.

z/OS Integrity and Audits

- z/OS integrity (at rest and in flight) results in having a trusted computing base:
  - A hardware and operating system architecture consistent with sound control objectives – approved by management
  - Real time ability to demonstrate to management and auditors that the running systems are (and stay) consistent with the approved architecture
- Apply the same discipline to boundaries that define the *controlled* environment (external security manager, databases, on-line systems, network, storage)

The *Inclusive* Trusted Computing Base

- External Security Managers:
  - Definitions of software components are appropriate (and STAY that way)
  - Privileged access is always closely restricted and real-time accountability exists
- Databases:
  - Configurations are known and access to them is tightly controlled
  - Control over data and processing (such as the ability to bind) is fully consistent with the data owners' specifications (including full knowledge of controls *not* implemented in z/OS)

The *Inclusive* Trusted Computing Base (more)

- On-line Systems:
  - Administrative access (including command lines) is always closely restricted and real-time accountability exists
  - System parameters are controlled at rest and in flight
- Networks:
  - Configurations are known and access to them is tightly controlled
  - Management knows which control points potentially exist and which ones are (and are not) operationally relied on.

The *Inclusive* Trusted Computing Base (and more)

- Storage:
  - The use of aspects of storage as system boundaries (or not) is known and approved by management (the architecture is consistent with the organization's control objectives)
  - At rest controls, inclusive of the IODF and SMS definitions are controlled
  - In flight controls are equally controlled – VARYDEV and SMS rules

z/OS Integrity & Audits - Thoughts

- Does your organization know and understand their responsibilities and what must happen on a daily basis to achieve integrity in their z/OS platforms?
- Is it realistic to expect z/OS integrity to be continuously maintained without using tools?
- Can your organization effectively communicate how it does not assume z/OS integrity to auditors?
- How much of your control environment relies upon naming conventions?