esper

TCO Considerations

for Android DevOps vs. Traditional MDM Solutions

esper.io

Content

Executive Summary

While many IT leaders believe that traditional MDM solutions can deliver lower total cost of ownership (TCO) for Android devices, the true cost output varies significantly according to hardware and use case. MDM can deliver a lower cost output than manual management methods for bring your own device (BYOD) use cases, but it may not have the same effect for corporate-owned, single-use (COPE) or single-purpose Android devices like kiosks, point-of-sale, ruggedized hardware, tablets, and interactive signage.

Esper's experience and multi-industry customer success stories show Android DevOps can lower OpEx by 60% or more compared to traditional MDM solutions in single-purpose device scenarios. By examining all the factors that lead to savings with deploying and managing single-purpose Android devices and apps, organizations can gain a deeper understanding of the TCO differences between traditional MDM and Android DevOps tools.

Overview

IT professionals often believe that single-purpose Android devices have a higher cost of ownership than Android smartphones in BYOD or corporate-owned, personally-enabled (COPE) use cases. Most total cost of ownership (TCO) analyses for Android mobility focus exclusively on workplace smartphone scenarios via BYOD or COPE models instead of the mission-critical, revenue-generating single-purpose Android devices that make up the majority of today's corporate fleets.

In actuality, the capital expenditure (CapEx) and operational expenditure (OpEx) costs associated with single-purpose Android are much different than CapEx and OpEx for workplace smartphones. Studies show the single highest cost factor involved in BYOD and COPE smartphone programs is data connectivity, which is rarely true for single-purpose Android devices such as kiosks, point-of-sale, or interactive signage. Mission-critical, single-purpose devices often have much higher operational and maintenance requirements than employee-enabled smartphones.

To build a comprehensive TCO analysis, IT pros need to look beyond monthly recurring connectivity costs to understand CapEx plus OpEx. In addition, it's wise to consider how the resource and monitoring requirements of secure single-purpose Android can impact long-term TCO. In this whitepaper, we've included a list of factors to consider when comparing Android DevOps solutions to traditional MDM.

Definitions

Traditional MDM

Mobile device management is a category of technology that manages policy, inventory, security, service, and apps for smartphones and tablets. MDM originated in the early 2000s, and it is generally specific to one or more mobile operating system (OS) platforms such as Android, iOS, or Windows.

A traditional MDM solution may be related to technologies such as mobile application management (MAM), unified endpoint management (UEM) or enterprise mobility management (EMM). For the purposes of this analysis, the traditional MDM category includes both cloud-based solutions and homegrown mobility management applications.

Android DevOps

Android DevOps is an emerging category of technology that describes an infrastructure for managing both traditional and non-traditional device deployment and app management, including support for kiosks, point-of-sale, telehealth devices, smart fitness equipment, and more.

Android DevOps creates a responsive connection between devices and cloud, allowing IT operations to monitor, update, and remediate the total state of device health. A DevOps solution encompasses device configurations, hardware, firmware, operating system, applications, and more.

An Android DevOps solution has capabilities that allow you to operate on a fleet (or subdivided fleet) holistically, so DevOps teams can worry less about device management and more about the product. It is also equipped with additional features to support a full-lifecycle approach to mobility - automation tools, pipelines, and a complete software development kit (SDK).

Factors to Consider in Building a TCO Analysis

IT Operation Staff Costs

For many organizations, investing in single-purpose Android devices can create a drastic shift in IT talent requirements and workloads. In order to successfully deploy nontraditional Android hardware using traditional MDM, you'll need to have personnel who are capable of each of the following functions:

- **Procurement:** All activities necessary to manage Android hardware, software and services, including device testing, business requirement documentation, and research.
- **OEM/ODM Management:** Organizational restructuring may be necessary to assign one or more persons to a role where they hold responsibility for maintaining relationships with OEM/ODM who provide off-the-shelf or pupose-built hardware.
- **Deployment and Upgrades:** In addition to initial MDM system deployment, your IT ops staff will need to continuously deploy Android apps, OS updates, and security patches. In many cases, this will require on premises maintenance activities or a unique update scheduling for each OEM relationship or device type.
- **Testing:** Individuals with Android or mobile QA expertise will be needed to measure various aspects of your product roadmap, including operational per formance and customer satisfaction.
- **Troubleshooting:** Your staff may be required to remain available to trouble shoot ongoing issues on a 24 x 7 basis, including on-site visits to fix Android hardware. This can require a large, distributed staff who are available to per form on-site visits at a moment's notice, or a relationship with a 3rd-party contractor for troubleshooting services.

Organizations should also consider talent availability when considering the impact of traditional MDM on an Android product roadmap. Nationwide, there is a noted talent shortage of individuals with Android device lab expertise, especially when it comes to nontraditional or purpose-built hardware. Depending on your region and total compensation packages, you may face significant delays in filling open positions.

In addition to talent-sourcing challenges, it’s important to take additional staffing costs into consideration. Internal costs related to recruitment, turnover, retention, ongoing training and other factors can all contribute to the costs of using traditional MDM to manage single-purpose Android fleets.

Device Deployment Timelines

Your organization's average time to deployment can impact both Android cost considerations and user satisfaction. Android deployment delays can carry both hard and soft cost factors due to a loss of potential revenue or damaged customer trust. Also, the potential ROI of single-purpose devices has evolved significantly as a result of the 2020 COVID-19 pandemic. A single unattended payment device likely has a higher potential ROI than ever before.

73% of customers now prefer self-service options such as kiosks, unattended point-of-sale, or interactive digital signage to face-to-face interactions with a human customer service representative, according to Aspect research.

While deployment timeline factors can vary significantly according to your traditional MDM and business requirements, it's important to consider the following factors when comparing traditional MDM to Android DevOps in terms of cost:

- Off-the-shelf or purpose-built device procurement time frames
- Shipping time from OEM to HQ
- Shipping time from HQ to deployment site
- Application and license procurement intervals
- Integration testing
- Application testing
- Remote site deployment

Depending on the deployment approach taken, sourcing and provisioning Android hardware via traditional MDM can take months or longer due to the challenges associated with testing devices for interoperability and on-site provisioning requirements.

In contrast, an Android DevOps solution that offers remote provisioning enables you to ship devices straight from the manufacturer to the intended point of use.

Maintenance and Remediation

The majority of MDM and Android DevOps solutions have similar pricing on paper - generally, organizations pay between $3-15 for each device monthly. However, the subscription fees associated with mobility management tools are not reflective of true TCO.

Monthly maintenance costs, or OpEx, for Android devices and apps can be around 15-20% for workplace smartphones. But, the maintenance costs for single-purpose devices can be much higher than the OpEx common to BYOD or COPE scenarios.

Single-purpose devices often have much more stringent security and privacy requirements to prevent end user misuse. Cutting corners on mobile security can double the risks of a security incident, according to Verizon Research. Sacrificing security for time savings can be incredibly costly, especially considering the global average cost of data breach recovery is $3.92 million.

Single-purpose devices such as kiosks or point-of-sale are generally operated at a separate location from the organization's headquarters and IT operations staff. Finally, single-purpose device scenarios are more likely to involve non-traditional or purpose-built hardware that's not necessarily built for remote maintenance or remediation, increasing the need for on-site repairs.

Using traditional MDM to manage single-purpose Android can impact both OpEx and CapEx significantly, depending on the following factors:

- Post-sales support subscriptions from device or MDM manufacturer
- On-site support to perform regular or ad hoc maintenance
- Testing application or peripheral compatibility prior to upgrades
- Revenue and reputational losses due to device downtime
- Productivity losses due to employee or customer device misuse
- Physical losses due to device tampering or theft
- Device shipping costs when applicable
- Time and resource commitment needed to track upgrade requirements

Today's Android fleets often represent a broad mixture of device models and device manufacturers. Each model and manufacturer can have a unique schedule for patching and upgrades, which can be challenging for IT operations staff to track.

Finally, non-traditional hardware upgrades can be surprisingly challenging, and in some cases, require technicians to physically unscrew a case or device components before security patches can even be applied. Trying to manually maintain a fleet using traditional MDM is often unpredictable, leading to remote device downtime and security risks.

New Application Adoption

Many single-purpose Android devices are locked to Android kiosk mode, a configuration state that limits end user access to download apps, make calls, or browse the internet. Locking users to one or more enterprise apps can protect productivity and privacy and mitigate the chances of misuse. But, secure and effective application deployment can be a challenge without sufficient tooling to support Android kiosk mode.

Traditional approaches to mobile application management are ineffective for modern Android mobility roadmaps. End users often don't need unfettered access to download social media apps via Play Store, especially since consumer apps can carry security and productivity risks. In addition, many traditional MDM solutions require that devices are powered on before app updates can be pushed to production, which is unreasonable for many global fleets.

IT operations pros need to shift their mindset from traditional mobile app management to edge content delivery. Content delivery networks create a powerful connection between the cloud and edge device by frequently caching updates to ensure end users have access to the latest app versions and device content.

While the OpEx and CapEx costs of traditional MDM vary by use case, they can include:

- On-site visits to physically update device application content
- Waiting for devices to come online before pushing app updates
- Minimal cloud lab testing tools to gauge real-world performance before deployment
- Failed or mixed deployment results leading to loss of uptime or satisfaction
- Poor visibility into app performance or success in production
- Minimal transparency between app development, IT ops, and customer success teams

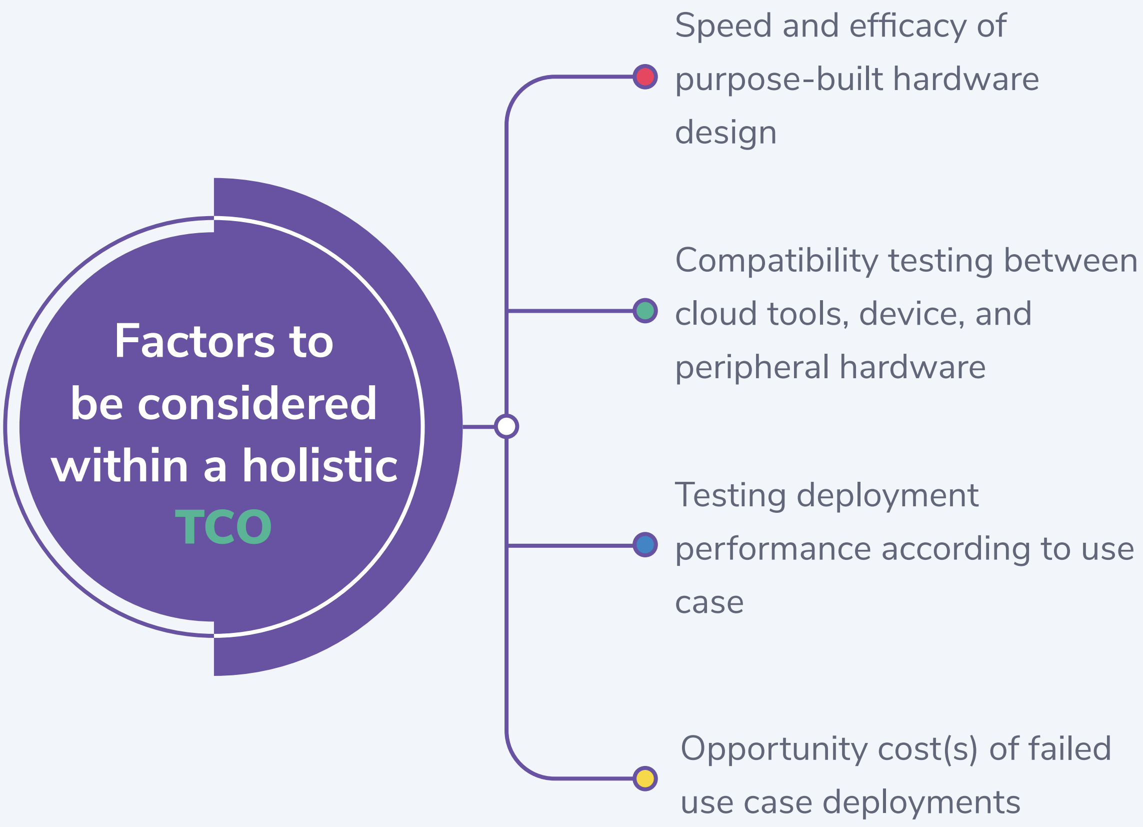
In contract, Android DevOps tools can push deployments to a single device or device groups, including provisioned devices which are powered down. They offer powerful features to easily manage multiple versions of a single enterprise application. Android DevOps solutions also include pipelines to automate deployment roll-outs and roll-backs based on custom success criteria.

New Use Case Adoption

The global pandemic has created a "new normal" for consumers worldwide. Consumer-facing organizations face new levels of customer demand for self-service and unattended mobile solutions. Business agility has always been critical, but it matters more than ever with the shift in customer expectations resulting from the pandemic.

Research by McKinsey confirms organizations with a faster operational response time to new use cases achieve 30-50% superior measures of efficiency. The most agile organizations also appreciate 10-30% better customer satisfaction scores. And finally, the same study shows that agile organizations have 10-20% better financial performance than their counterparts.

The ability to rapidly adopt new use cases, especially remotely, has not traditionally been a part of TCO analyses for Android mobility. But, the idea of use case adoption and opportunity loss has clear value in the post-pandemic world. Shekel Brainweigh research indicates that 87% of global consumers now prefer digital self-service.



Factors to be considered within a holistic TCO

Speed and efficacy of purpose-built hardware design

Compatibility testing between cloud tools, device, and peripheral hardware

Testing deployment performance according to use case

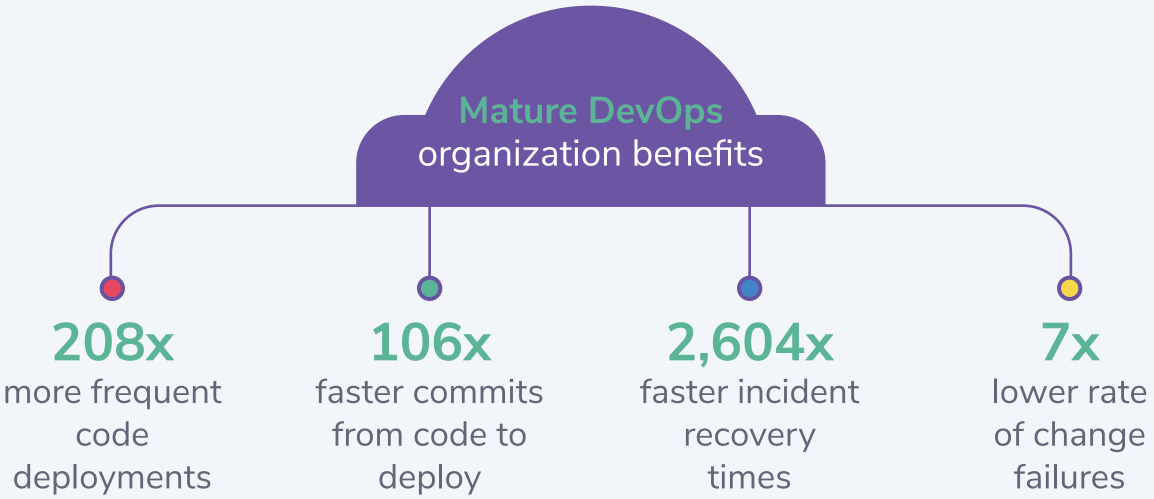
Opportunity cost(s) of failed use case deployments

With traditional MDM, organizations run the risk of waiting months or even years before being able to expand to new use cases. If a purpose-built hardware project fails or organizations order hardware that is not compatible with the traditional MDM, new components of the product roadmap can be significantly delayed.

In contrast, an Android DevOps solution can significantly streamline the deployment process by offering validated devices, validated peripherals, and cloud test tools that are backwards compatible with developer and QA environments.

DevOps Maturity

Android DevOps has many characteristics in common with other DevOps practices, including a focus on cloud maturity, automation, and operational excellence. Google’s annual State of DevOps report reveals that “elite performers,” or the most mature DevOps organizations, reap company-wide benefits:



Mature DevOps organization benefits

208x more frequent code deployments

106x faster commits from code to deploy

2,604x faster incident recovery times

7x lower rate of change failures

Essentially, a mature infrastructure for DevOps has close association with operational excellence and a customer-obsessed culture. The most sophisticated organizations deploy new code daily and have low deployment failure rates.

Some of this efficiency comes from DevOps technologies such as highly-scalable cloud infrastructure, and some is cultural - success metrics are shared among developers, operations, and customer success teams within a DevOps working model.

But, it can be challenging for organizations to adopt DevOps processes using a traditional MDM solution. Organizations can inadvertently absorb both hard and soft costs when using traditional MDM solutions for single-purpose deployments, which include:

Delayed device and app deployments

Extended app or device downtime

More failed deployments

Scalability

Historically, mobility had a 1-to-1 ratio for devices to employees. In the early 2000s, each member of the senior leadership team was equipped with a Blackberry. A decade ago, personal smartphones became a commonplace tool for both workplace and non-work usage scenarios, which led to the advent of the BYOD movement. Today, the ratio has shifted dramatically. Connected devices significantly outnumber contributors at most organizations. IDC predicts there will be 41.6 billion smart, connected edge devices by 2025.

Traditional MDM tools simply weren't architected to handle the demands of a large-scale fleet. And, trying to use a traditional MDM solution to manage a fleet of 10,000 or 100,000 devices can create serious efficiency problems. It's crucial to consider whether your current mobile device management solution can scale to meet your Android roadmap, or whether you'll absorb risks such as performance degradation or inefficiency. Cost factors associated with using traditional MDM while scaling can include:

- Limitations on reusing or repurposing device provisioning templates
- The inability to reuse a single provisioning template for multiple device types
- Limited or nonexistent ability to remotely provision devices using IMEI or other factors
- Minimal remote, over-the-air deployment and provisioning tools
- Nonexistent device group commands or nested device groups
- Poor ability to create programmatic actions, alerts, or reporting
- A lack of APIs for custom-built, cloud-connected solutions

In contrast, an Android DevOps tool is built to scale from 10,000 to 100,000 devices and beyond without adding to the IT operations burden. It offers full cloud GUI features as APIs for custom development scenarios and sophisticated, nested groups to support programmatic or streamlined actions at the single device, device group, or sub-group level.

Conclusion: Esper for Android DevOps

A thorough analysis of all cost components described above shows why Android DevOps can lower OpEx compared to traditional MDM. In order to fully understand the cost of MDM solutions built for smartphones, organizations must look beyond pure measurements of OpEx to understand the greater context and related factors, including the significant provisioning, maintenance, and remediation requirements needed to secure remote, single-purpose devices.

Our experience indicates that traditional MDM have value for select scenarios, including both corporate and employee-owned smartphones. But, MDM can create massive efficiency and orchestration challenges when organizations attempt to scale these tools to manage kiosks, point-of-sale, interactive signage, or other single-purpose fleets. In practice, many organizations find they are unable to integrate non-traditional hardware or use cases with their existing cloud tools or have little visibility into the health or performance of field devices.

Esper is the world's first Android DevOps solution built specifically for the challenges of a modern single-purpose Android fleet of both traditional and nontraditional hardware. We're compatible with GMS-certified and AOSP (non-GMS) hardware, Android OS versions 4.4 / 5.x / 6.x+ to provide comprehensive support for both off-the-shelf and purpose-built solutions that span retail, hospitality, education, healthcare, fitness, and more.

Traditional MDM has it's value, but Esper is the first tool built to simplify the challenges of scaling a single-purpose fleet. We offer approximately 700 enhanced or validated Android devices from 100+ OEM/ODM and manufacturing support to streamline deployments. Our user-friendly cloud GUI features are offered as Android management APIs, all of which is backed by best-of-class customer support. Visit esper.io/signup today to get started.

About the Authors

Billy Sheng

Director of Cloud Operations, Americas and EMEA at Esper. He is an AWS Certified Cloud Practitioner, Microsoft Certified in Azure Fundamentals, and holds a BS in Engineering from University of Washington. Billy is a champion for remote debugging and effective CI/CD practices for single-purpose Android device fleets.

Jasmine Henry

Director of Cybersecurity at Esper. She has a MS in Informatics & Analytics and a Graduate Certificate in Healthcare Informatics from Lipscomb University in Nashville, Tennessee. Jasmine has written about endpoint security for Time, Forbes, Reuters, and many other publications.