Debunking the Top Myths About Unsupported Linux for Embedded Development

Guide

www.suse.com

Guide

SUSE Embedded Solutions
SUSE Linux Enterprise

Debunking the Top Myths About Unsupported Linux for Embedded Development

Linux has changed the embedded systems landscape and has quickly become the preferred code base for development. Device manufacturers can elect to design, build, and support the Linux operating system themselves (roll your own—RYO) or partner with a supported Linux distribution.

Embedded solution developers are often attracted to the "free" aspect of Linux and choose the RYO development route. However, an unsupported distribution might leave developers vulnerable to a wide range of hidden costs, risks, and time-consuming activities. This paper explores the top questions and myths about unsupported Linux for embedded applications.

Trends in Embedded Development

The embedded development market has seen rapid changes over the past few years. There is a growing emphasis on IoT systems, reliable connectivity, enterprise-grade security, and extended device lifecycle management. Devices continue to become more complex, which makes them more difficult to manage. In turn, the ongoing management, patching, and updating of the software powering these devices creates long-term challenges for embedded solution developers.

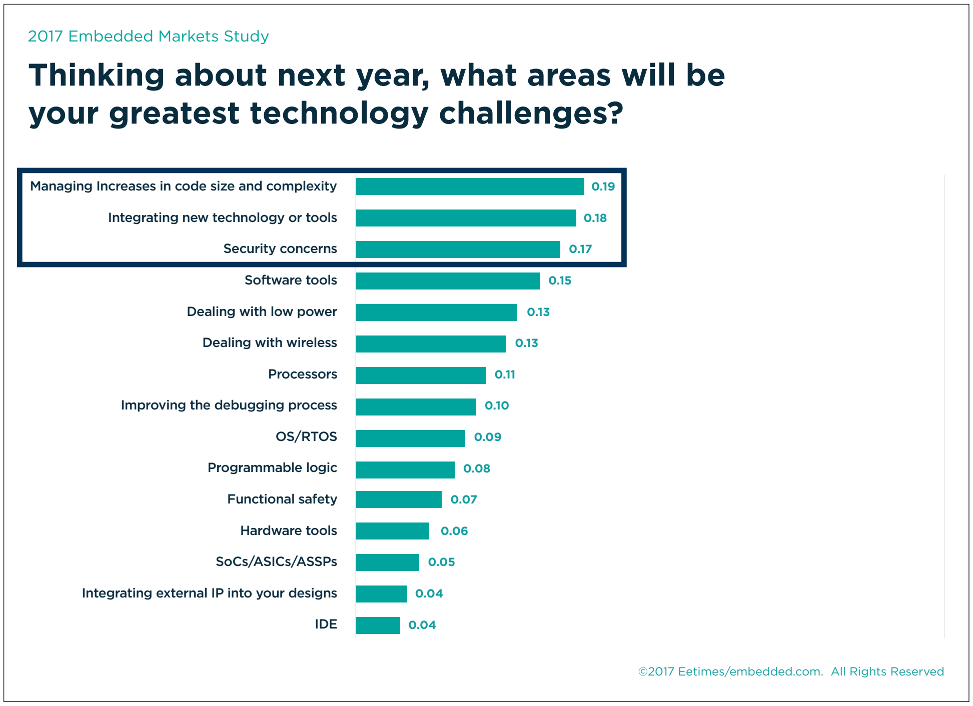
Benefits of SUSE® Linux for embedded applications:

- *Expert Linux support*
- *Comprehensive management tools*
- *Enterprise-grade security*
- *Configuration & distribution*
- *Automated patch management*

A recent EE Times survey *(see Figure 1 on the following page)* highlights the top challenges embedded system developers face, including:

- *Code complexity (19%)*
- *Integrating new technology (18%)*
- *Security (17%)*

Embedded applications by nature are very fragmented, which can lead to long development cycles and unpredictable maintenance schedules. New pressures continue to emerge for embedded software developers to bring products to market faster, make them more secure, and keep costs down.



©2017 Eetimes/embedded.com. All Rights Reserved

Figure 1. *EE Times 2017 Survey: Greatest Technology Challenges*

Debunking the Top Myths of Using a RYO Environment

To address and keep up with these top technology challenges, developers often choose to roll their own Linux as opposed to leveraging a supported Linux operating system. The decision to build or buy an operating system is a complex one that requires consideration of multiple factors that impact every aspect of the project—from development time and man-hours, to security updates and licensing costs.

As embedded development environments continue to become more complex, management of a RYO system can present numerous challenges for organizations. Consider the following top myths about RYO environments:

Myth #1: RYO Is Less Expensive Than a Commercial Operating System

As an attractive development platform, unsupported Linux distributions tend to capture the attention of organizations through the concept of reduced investment and no upfront cost. When evaluating the associated costs, businesses should strategically weigh the long-term vs short-terms costs and requirements of unsupported Linux.

Hidden RYO environment costs include:

- **High price of development/ management:** *The costs associated with developing and maintaining a custom Linux distribution or proprietary operating system can be prohibitive and pull development resources away from value-add tasks. Unfamiliarity with the open source community, lack of technical training and support, and the need to ensure compatibility with a range of hardware platforms all contribute to higher costs and longer development cycles for organizations that rely on unsupported Linux distributions.*
- **Complicated and expensive licensing:** *Organizations that rely on unsupported Linux distributions often end up with a mix of proprietary and open source code, which complicates licensing requirements. Expensive and overly complex licensing and subscription models can drive up development costs, as well as costs to end users, thereby eroding profits or diminishing competitive advantage.*

Myth #2: RYO Provides More Flexibility and Control

A RYO Linux environment promises to enable customized development and avoid vendor lock-in. With that flexibility, embedded system developers are often tasked with providing their own resources to manage day-to-day development needs and support the lifecycle. When evaluating operating systems, device manufacturers should factor in the additional administrative, technological, and management considerations in an unsupported RYO Linux environment.

Hidden resource and time considerations with a RYO environment include:

- **Ongoing management/patching/updating:** *Most embedded devices are in the field and in production for an average of 5–10 years*[1]*, often making in-house system management and maintenance overwhelming for development teams* (see Figure 2 on the following page). *Many organizations lack the resources and/or manpower to regularly monitor and update security, manage bug fixes, and patch the operating system throughout the product lifecycle.*
- **Decreases in developer productivity:** *Developers often take on more administration and systems management when they use a RYO Linux platform—stealing valuable time away from their core priority of device development. As the life of the device increases, the required development resources and skills on staff to manage the environment are often not sufficient.*

Myth #3: Managing Security Will Be Easy with Unsupported Linux

Prior to selecting an operating system, it is imperative that embedded solution developers ask targeted questions about the critical nature and manageability of security. Security needs to be a key ingredient in embedded solution development from the outset—not tacked on as an afterthought.

Hidden security concerns with a RYO environment include:

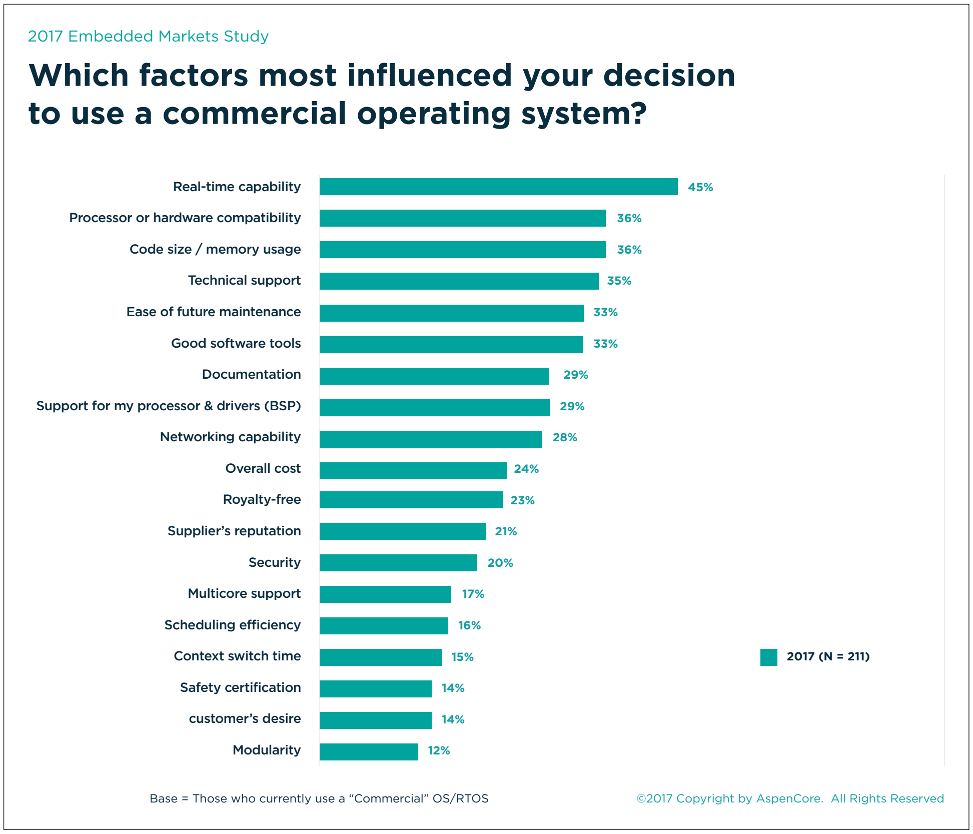
- **Difficulty administering patches and updates:** *Embedded solution development requires multiple updates and patches through the entire product lifecycle. If using a custom operating system or unsupported Linux distribution, embedded solution developers need to include these security updates themselves. The financial commitment and number of man hours associated with developing and maintaining a custom Linux distribution or proprietary operating system can be prohibitive. As more functionality is embedded into smaller device footprints, security concerns and the ongoing ability of device manufacturers to effectively manage these devices becomes increasingly complex.*
- **Security updates aren't seamless:** *To maintain secure devices and mitigate their overall risks, embedded solution developers require seamless patch and update capabilities. Effective updates address software bugs and flaws that can alleviate vulnerabilities. Without updates and patches, embedded systems might be more easily compromised. AuthO recently found that 85 percent of developers admitted that they had rushed applications to market despite having security concerns about the device*[2]*. Embedded solution developers need to be confident that security patches and updates are easily managed prior to launch and over the course of a product life cycle. With a supported distribution like SUSE, companies can rest assured that they will get immediate response times and quick resolution to potential security threats.*

The Business Case for Embedded Solution Development

After considering the top myths of RYO development and their associated hidden costs, a compelling business case can be made for commercial-grade operating systems such as those offered by SUSE.

Figure 2 on the following page highlights the top reasons why organizations moved to a commercial operating system, according to EE Times. Compatibility, technical support, ease of future maintenance, software tools, security, and documentation are all key factors on why recent survey participants chose a commercial Linux operating system for embedded projects.

[1] **www.inhand.com/designing-long-life-cycle-minimizing-cost-new-product-development/**
[2] **https://auth0.com/blog/surprised-turns-out-consumers-dont-trust-iot-security/**



Base = Those who currently use a "Commercial" OS/RTOS

©2017 Copyright by AspenCore. All Rights Reserved

Figure 2. *Which factors most influenced your decision to use a commercial operating system?*

A SUSE Embedded Operating System, based on the SUSE Linux Enterprise code base, enables developers to support their top technology challenges and also provide the technical capabilities required in a commercial operating system, with a meaningful choice of components up and down the software stack. With the inclusion of an commercial-grade Linux kernel, plus a core of available user-space packages, SUSE Linux Enterprise also provides a significant ecosystem of modules, extensions, and easy-to-use developer tools. SUSE Embedded Solutions are capable of solving the top three challenges shown in Figure 1: code complexity, integrating new technology, and security.

With a supported commercial embedded operating system, developers can dedicate their resources to value-added tasks: designing systems that the meet the specific needs of their target markets, rather than doing system builds and time-consuming maintenance. Over time, commercial operating systems provide maintenance and support that benefit lifecycle management and reduce the total cost of ownership.

Companies with a need for an embedded operating system often select a free Linux-based system without realizing that RYO and custom systems carry a variety of underestimated costs and diminishing returns in regard to the cost and man hours associated with security, patch management, configuration, scalability, and speed to market.

Companies with a need for an embedded operating system often select a free, unsupported Linux without understanding that unsupported, custom systems carry a range of hidden costs and technical challenges.

Comparison Attributes/Features	Description of Feature	Unsupported RYO Linux	SUSE Embedded
Initial cost of the operating system	There is no initial price associated with a RYO operating system, it is simply a free download. However, refer to the hidden costs (3) associated with a RYO operating system.	$	$$[3]
Long-term costs of the operating system	The costs associated with developing and maintaining a custom Linux distribution or proprietary operating system can be prohibitive and pull development resources away from value-add tasks. Unfamiliarity with the open source community, lack of technical training and support, and the need to ensure compatibility with a range of hardware platforms all contribute to higher costs and longer development cycles for organizations that rely on unsupported Linux distributions.	$$$[4]	$
Third-party support	Organizations often do not have access to adequate technical resources and expertise required to support rapid, Linux-related development enhancements and services. See Figure 2, which highlights technical support as a key reason why many developers turn to commercial operating systems for embedded solution development.	— Needs to be created and developed based on RYO environment	X Included
Documentation	Documentation is critical to the successful management of the embedded development environment.	— Needs to be created and developed based on RYO environment	X Included
Security	Multiple updates and patches are required throughout the product lifecycle. If using a custom operating system or unsupported Linux distribution, embedded solution developers need to include these security updates themselves. The financial commitment and number of man hours associated with developing and maintaining a custom Linux distribution or proprietary operating system can be prohibitive.	— Needs to be created and developed based on RYO environment	X Included in Support Services
Patch management	As more functionality is embedded into smaller device footprints, security concerns and the ongoing ability of device manufacturers to effectively manage these devices becomes more complex.	— Needs to be created and developed based on RYO environment	X Included in Support Services
Licensing management	Organizations that rely on unmanaged Linux distributions often end up with a mix of proprietary and open source code, complicating licensing requirements. Expensive and overly complex licensing and subscription models can drive up development costs, as well as costs to end users, eroding profits or diminishing competitive advantage.	—	X Flexible and requirements-based agreements
Development costs	When developers have the ability to develop feature-focused systems, they are able to reduce cost and required man-hours, while enjoying the flexibility and freedom to maximize build efforts. Developers and designers can focus on creating powerful business solutions and building innovative hardware, rather than spending time ineffectively on the management, maintenance, and scalability of an unmanaged Linux system.	$$$	$

X = Included — =Not included $ = Less Expensive $$$ - More Expensive

Learn more about SUSE Embedded Solutions at: **www.suse.com/embedded** and request your complimentary ROI consult on the potential savings when moving from a Roll Your Own (RYO) environment to a SUSE Embedded operating system.[5]

[3] Contact the SUSE Embedded team for details on Licensing and Pricing: **embedded@suse.com**
[4] See hidden costs in Myth #1 (short-term vs long-term)
[5] **www.suse.com/products/server/jeos/**

Additional contact information and office locations:
www.suse.com

www.suse.com