The Metaport EOL Survival Guide

We've worked where you work so we reckon we know a thing or two about agencies and managing software EOL dates. In this guide, we offer some practical steps and boilerplate copy for use in your own customer conversations.

Get inventory 01

Popular opinion holds that inventorying a portfolio is the first step to understanding it. It's not terrible advice, but the huge range of tech managed by agencies means it's not always so easy.

Inventory: Document your portfolio using Metaport, wikis or spreadsheets: Use a gantt chart with years on the first row and rows beneath according to release, support, and EOL date. Annotate with key business dates such as contract renewals for added utility.

Prioritize: Projects with a larger "blast radius" go to the top of the pile:

- accredited customers (PCI-DSS, SOC 2, ISO 27001, etc)
- managing PII (Personally Identifying Information)
- large user volumes (think government app vs local builder's website)
- SLAs (think uptime, RPO & RTO)

Identify: Record EOL dates using Metaport, isitendoflife.com or via vendor websites.

Plan: Estimate upgrade effort for Technical, QA and Project Management time.

Communicate: Present effort, timelines, and budgets to your customers, well in advance.

Monitor: EOL dates are not one-offs, so keep inventories up-to-date.

Convince the inconvincible 02

An upgrade project reduces risk but doesn't add any new features which is often a problem for customers. But an Account Manager can explain the costs of action versus inaction in non-technical terms to clarify the value proposition.

Account Managers can share screenshots from the isitendoflife.com tool.

Example customer email 03

We've been supporting <App> since <Date> and like most modern applications, it relies on third-party software components which are managed by external maintainers.

From time to time, these maintainers reduce or end support for specific versions, it's something we actively monitor, including security updates and end-of-life (EOL) dates, on your behalf.

We've identified that <Number> component(s) used by <App> are reaching EOL this year. Our advice is always to keep production systems on supported versions which reduces security and operational risk. Our recommendation therefore is to upgrade these component(s) sooner rather than later, to stay ahead of those dates.

Once the upgrade work is complete, <App> should continue to operate as normal, with no visible change for users. Allowing EOL software to expire doesn't stop <App> from working, but it does increase risk over time as external conditions inevitably change.

<App> Upgrade Project					
Component	Installed Version	EOL Date	Supported Version	EOL Date	Cost
foo/bar	1.2.3	Dec '26	2.0.1	Dec '28	$From - $To

Say no to Groundhog Day 04

Repeating the same EOL effort reactively is costly and disruptive, but the following practices will help Project Managers minimise that cycle over time:

- **Living documentation:** Maintain up-to-date records and decision checkpoints so EOL considerations are addressed early.
- **Application visibility:** You can't secure or plan upgrades for systems you can't see. Clear asset visibility supports better forward planning.
- **Ongoing support models:** Support arrangements create space for proactive maintenance, rather than reactive upgrades.

Metaport is designed for agencies just like yours which struggle to keep ahead of EOL dates, security vulnerabilities, and SSL certificate expiries. We're always keen to learn more about your business, and how Metaport might help. Please drop us a message, we'd love to chat!