



Patton Electronics, Co.
7622 Rickenbacker Drive
Gaithersburg, MD 20879
Tel: +1 301.975.1000
Fax: +1 301.869.9293

SmartWare Release Strategy

White Paper

Abstract

This white paper describes the strategy of SmartWare software releases. Multiple software releases are maintained (available) at the same time: One *technology release* to provide periodic, fast access to new software functionality, and several *maintenance releases* to regularly publish problem fixes with the least possible side-effects. Lastly, the *customer release* is a maintenance release for a specific customer. Clearly defined software life cycles, release states and documentation deliverables allow customers to choose between a maintained, stable and backwards-compatible software with known functionality and a software containing the latest features. This strategy allows quality, consistency and emerging technology at the same time for all products running SmartWare software.

Contents

1	Introduction	3
2	Terminology	4
2.1	Overview	4
2.2	Releases	4
2.2.1	Major Release	4
2.2.2	Technology Release	4
2.2.3	Maintenance Release	5
2.2.4	Customer Release	5
2.3	Builds.....	5
2.3.1	Technology Build	5
2.3.2	Maintenance Build	5
2.3.3	Customer Build	6
3	Release Strategy	7
3.1	Overview	7
3.2	Software Change Classification and Synchronization	7
3.3	Maintenance Release Life Cycle	8
3.4	Overlapping of Maintenance Releases	9
4	Deliverables to Customers.....	11
4.1	Overview	11
4.2	Software Images	11
4.3	Software Configuration Guide (SCG).....	11
4.4	Release Notes (RN)	11
5	Conclusion	12

1 Introduction

SmartWare is a software suite that runs on multiple hardware platforms manufactured and distributed by Patton Electronics. Software is subject to changes over time due to the addition of features and the resolution of software bugs. Thus, software upgrades have to be delivered from time to time.

Overall, the goal of delivering new software is customer satisfaction. Within that goal it is desirable to provide various categories of software to fit the different needs of customers. Customers may have different reasons to upgrade software; some of them are even orthogonal to each other. Consider a comparison between an Internet telephony service provider (ITSP) and an enterprise integrator.

The internet telephony service provider (ITSP) uses the software on its VoIP gateways and is interested in stability with backwards compatibility – each software upgrade requires thorough testing in his environment. He will upgrade only if there are critical issues that force him to upgrade.

On the other hand, an enterprise integrator often wants to employ the latest technical developments and will choose new software upgrades. This gives him the opportunity to offer additional services to his customer base.

For both of these examples, a software upgrade is required – while one customer prefers maximum software stability, the other is searching for the latest technology to give him the competitive edge. Thus, our software release strategy has two goals in releasing new versions of software:

- Make new software features accessible to customers within a short timeframe
- Maintain stability, backwards compatibility and quality to customers requiring lowest risk upgrade.

Patton Electronics' policy is to provide new features as well as bug fixes free of charge to all customers. This document describes the strategy of SmartWare releases, the release life cycle and associated documentation deliverables which are designed to meet the above goals.

2 Terminology

2.1 Overview

The terms Release and Build are often used throughout this document. They come in different flavors and are described in the following paragraphs. Figure 1 provides an overview of all terms and how they relate. It shows a Major Release, how it contains one Technology Release and multiple Maintenance Releases, and how these are instantiated by Builds on a regular basis.

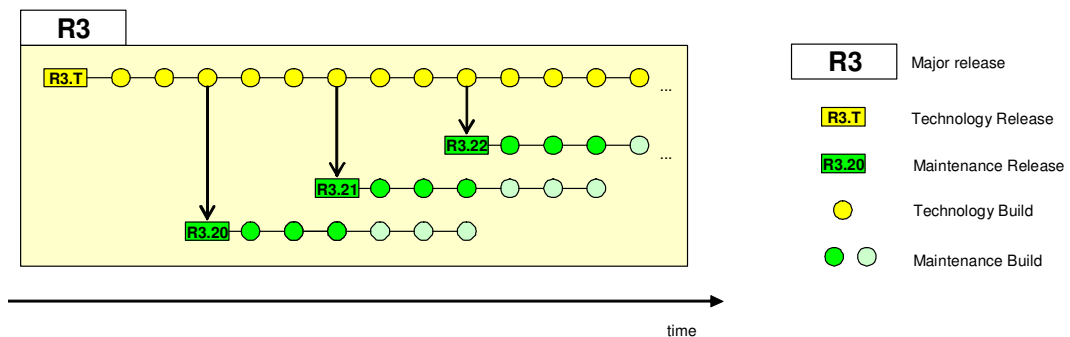


Figure 1: Releases and Builds

2.2 Releases

A Release is an abstraction of a certain set of software functionality and hardware platform support. Within this document, it is represented with a rectangle containing the Release name. There is one Major Release that contains three types of sub-Releases: A Technology Release, Maintenance Releases, and Customer Releases.

2.2.1 Major Release

A Major Release denotes a bigger step within the software development process. This step is usually R&D driven and defined by a new or fundamentally revised software component, enabling support for new functionality or new hardware, but increasing risks for migrating from older Releases. It serves as container for a Technology Release and several Maintenance Releases. It is represented with a white rectangle containing the bold faced name of the Release, e.g., **R3**.

2.2.2 Technology Release

The Technology Release provides continuing feature development within a Major Release. Its key goal is to publish new features within the scope of the Major Release over time. Thus all new features, hardware support and bug fixes are integrated. There is one Technology Release within each Major Release, and it serves as a container for all Technology Builds. It is represented with a yellow rectangle containing the name of the Release, e.g., **R3.T**.

2.2.3 Maintenance Release

A Maintenance Release is a Release spawned from the Technology Release. It freezes the set of software functionality and behavior of the Technology Release at a certain point in time and maintains it. Its key goal is to provide software stability. Bugs are fixed but no new features are added, and the behavior changes are limited to the minimum necessary to fix bugs. Backwards compatibility is maintained. A Maintenance Release is represented with a green rectangle containing the name of the Release, e.g., *R3.20*, and *R3.21*.

2.2.4 Customer Release

A Customer Release is a Maintenance Release created for a specific customer. It is an abstraction of a set of software functionality which is requested and controlled by that customer. The Customer Release can be spawned from either a Technology Release or a Maintenance Release upon special arrangement with Patton Electronics. Setup and maintenance fees as well as special terms and conditions apply.

2.3 Builds

A Build is the instantiation of a Release at a certain time of development. It consists of the *set of software images* that are used in the various products supported by the Release.

2.3.1 Technology Build

A Technology Build is the instantiation of the Technology Release at a certain time of development. It publishes new software features and bug fixes for all current products and may add support for new and upcoming products. Figure 2 shows a Technology Release and the Builds as they occur over time with the nomenclature of Technology Builds. Each Build consists of a set of software images that cover all supported products. Technology Builds are represented by yellow circles.

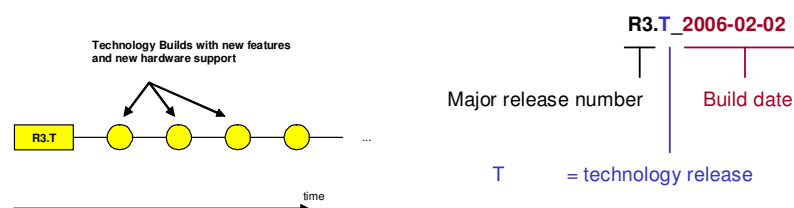


Figure 2: Technology Builds

2.3.2 Maintenance Build

A Maintenance Build is the instantiation of a Maintenance Release at a certain point in time. It publishes bug fixes for all products supported by the Maintenance Release. Figure 3 shows a Maintenance Release and the Builds that instantiate it over time with the nomenclature of Maintenance Builds. Each Build consists of a set of software images that cover all supported products and is backwards compatible with respect to all previous Builds of the same Maintenance Release. Maintenance Builds are represented either by bright green or light green circles.

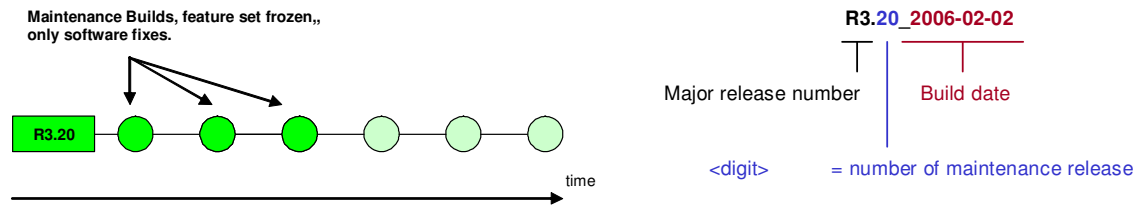


Figure 3: Maintenance Builds

2.3.3 Customer Build

A Customer Build is the instantiation of a Customer Release at a certain point in time. It may contain new software features and/or bug fixes at the discretion of the customer. As build lifecycle and nomenclature may vary from customer to customer, this document does not further discuss Customer Builds and Releases.

3 Release Strategy

3.1 Overview

As section 2 already suggests, the Technology Release within a Major Release is used to make new software features accessible quickly and on a regular basis, while Maintenance Releases provide stability and backwards compatibility.

For example, the Major Release *R3* evolves into one Technology Release (*R3.T*) and multiple Maintenance Releases (*R3.20*, *R3.21*, etc.). Figure 4 shows this dependency graphically with the example of Major Release *R3*. The Maintenance Release *R3.20* for example is a consolidation of all features and hardware support of the Technology Release *R3.T* at time *T1*. At a later point in time, another Maintenance Release is spawned, again freezing the current state of features and hardware support of the Technology Release, e.g., Release *R3.21*.

Builds from all Releases are created periodically, every 2 months (more on the life cycle of Maintenance Releases in the next section). A new Maintenance Release is spawned from the Technology Release every 6 months. These timing parameters allow rapid and regular access to both new features in the Technology Release as well as bug fixes within Maintenance Releases. They also allow for the consolidation of new software features in the form of a Maintenance Release after a reasonable amount of time.

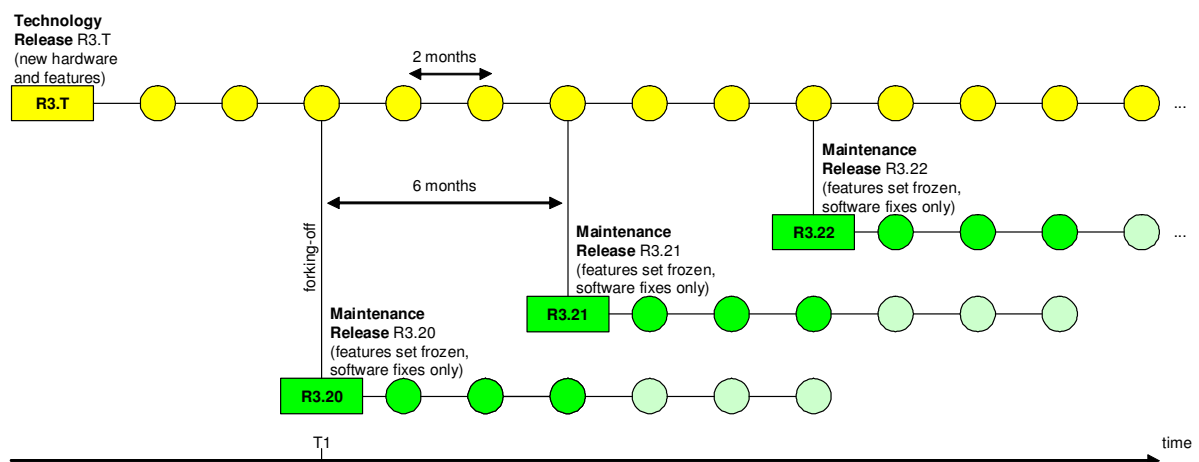


Figure 4: Major Release R3 evolution

3.2 Software Change Classification and Synchronization

A bug fix that is integrated into a Maintenance Build *R3.20* will, if applicable, at the same time be incorporated into the Builds for the newer Maintenance Releases *R3.21* and the Technology Release *R3.T*. Bug fixes are synchronized bottom-up – they are first fixed on the oldest Maintenance Release and then propagated into all applicable newer Maintenance Releases and the Technology Release. New software features are synchronized top-down. New features are only incorporated into the Technology Release. Figure 5 shows this relation graphically.

A decision must be made for each software change: Is it a *bug fix* (to be integrated in all applicable Releases) or is it a *new feature* (only to be integrated in the Technology Release)? All software change requests are reviewed for potential integration according to the classification matrix shown in Figure 6.

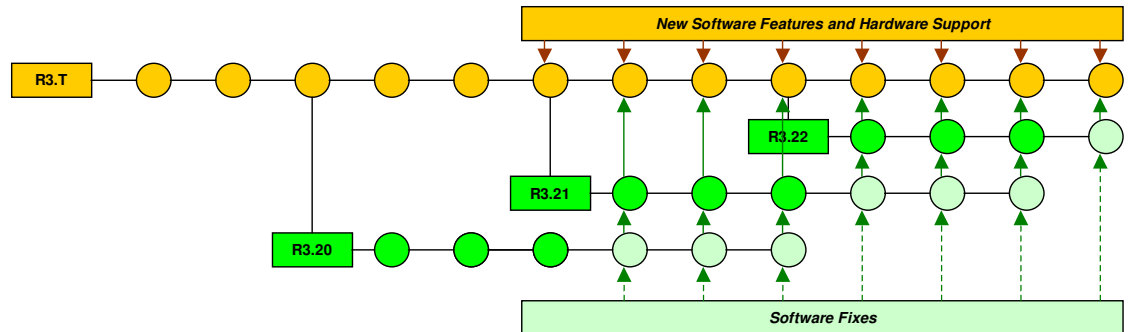


Figure 5: Software change synchronization

Severity Depth	Critical	Minor
New Feature - Hardware Support	all T-Builds	all T-Builds
Bug Fix	all T-Builds - all Maint-Builds	all T-Builds - commercial Maint-Builds

Figure 6: Software change classification

3.3 Maintenance Release Life Cycle

Each Maintenance Release follows the same life cycle. After being spawned from the current state of the Technology Release, it is first tested and subsequently categorized from *Commercial* to *Stable* and finally to *Retired* state. The state of the Release defines the availability and support provided for it.

Table 1 shows the states during the life cycle of a Maintenance Release. A graphical illustration is given in Figure 7.

Time Line [Months]	Milestone	State after Milestone	State Description
-2	SPAWN	Test	The Release is not yet available to customers. The Technology Build from which this Maintenance Release was spawned is tested in the field.
0	FCS First Commercial Shipment	Commercial	The Release is accessible to customers. Builds are published every two months. Support is provided, all bug fixes are integrated.
6	EOS End of Sale	Stable	The Release is still available for download to all customers, but only critical bug fixes are integrated. Builds are only published if a critical bug fix requires it.
12	EOE End of Engineering	Retired	No engineering changes are made on the Release. Support is still provided, and customers are helped to migrate to the next Maintenance Release. No Builds are published anymore for this Release.
18	EOL End of Life	(none)	The Release is not publicly accessible anymore, no support is provided.

Table 1: Maintenance Release Milestones and States

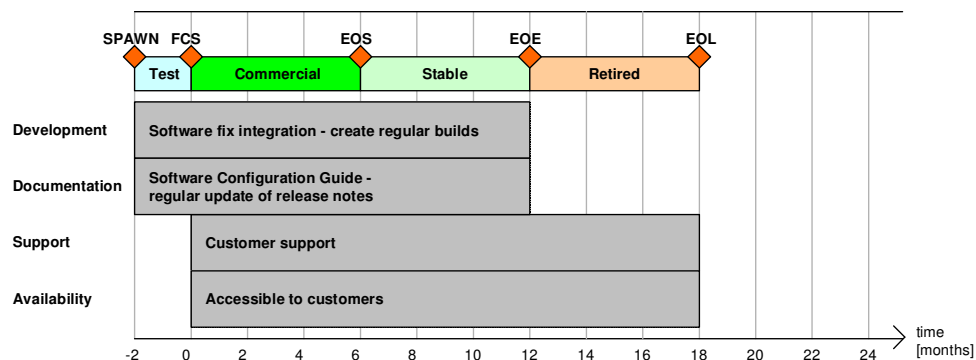


Figure 7: Maintenance Release life cycle

3.4 Overlapping of Maintenance Releases

Each Maintenance Release has a lifetime duration of 18 months. Since a new Maintenance Release is spawned every 6 months, the Maintenance Releases are overlapping. A maximum of three Maintenance Releases are publicly available (from different Maintenance Releases) – one in Commercial, one in Stable, one in Retired state.

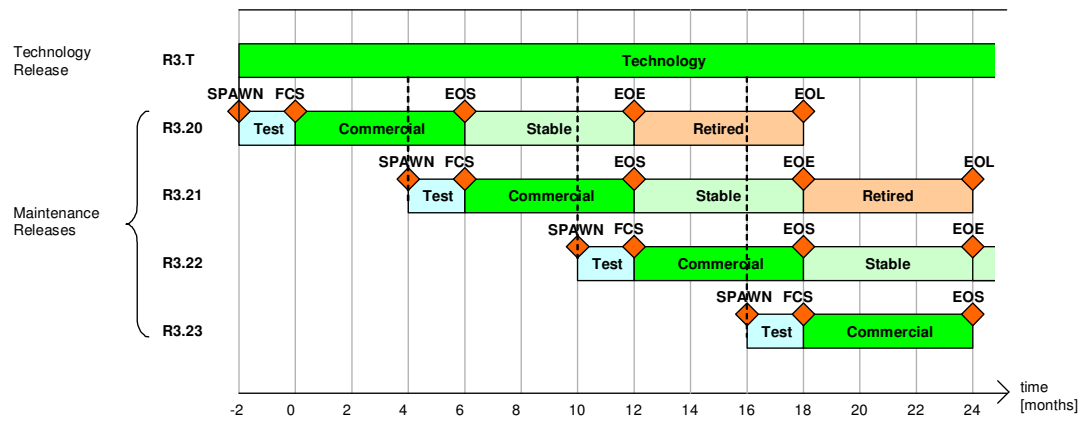


Figure 8: Overlapping Maintenance Releases

4 Deliverables to Customers

4.1 Overview

During the life cycle of a Maintenance Release, the following deliverables are provided to customers:

- Software Images
- Software Configuration Guide (SCG)
- Release Notes (RN)

These deliverables are accessible to all customers on the Patton internet site upgrades.patton.com. Figure 9 shows the deliverables in relation to the Maintenance Release life cycle.

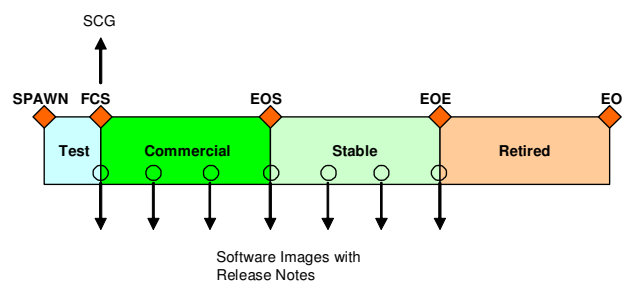


Figure 9: Deliverables overview

4.2 Software Images

As defined in section 2.3, a software image is part of a Build. For each product, software images of the following Builds are accessible:

- A few of the recent Technology Builds are available as *Early Access*
- The latest Maintenance Build of each Release in commercial or stable state is available as *Commercial*

All older software images are available on request from the Patton support team.

4.3 Software Configuration Guide (SCG)

The Software Configuration Guide describes how to use and configure the software (it's the software user manual) of a specific Maintenance Release. Hence, there is one Software Configuration Guide for each Maintenance Release. It is published at FCS.

4.4 Release Notes (RN)

Release Notes list the software changes integrated since the previous Build of the same Release. They are published together with each software image of a new Build:

- Release Notes of Maintenance Builds describe the history of accumulated changes since FCS
- Release Notes of Maintenance Builds also contain information on how to migrate from one Maintenance Release to the other.
- Release Notes of Technology Builds describe the history of software changes since the last Maintenance Release was spawned.

5 Conclusion

The strategy of software upgrade releases for SmartWare has been described. It illustrates that by distinguishing *Technology Releases* and *Maintenance Releases*, the goals set forth in section 1 – making new software features available quickly and maintaining stability and backwards compatibility – are satisfied. In review, the Technology Release contains the latest new software features, while Maintenance Releases keep a well-defined set of software functionality, only contain fixes, and are fully backwards compatible. For each Major Release, one Technology Release and several Maintenance Releases are engineered and supported at the same time. For each of the engineered releases, updates are published regularly in the form of Builds, available free of charge and for all hardware platforms supported by the Release. The advantages of this strategy are:

- **Stability and Quality**
 - Software upgrades are free of charge
 - Software upgrades are published on a regular basis
 - Features are frozen within a Maintenance Release
 - Only minor changes, mostly bug fixes, are done within a Maintenance Build
 - All Builds within a Maintenance Release are fully backwards compatible
- **Faster time-to-market for new features**
 - New features are integrated into the Technology Release which is regularly published
 - More room for adaptations and customer acceptance
- **Accurate documentation**
 - The associated Software Configuration Guide completely describes the software functionality contained in a Maintenance Release.
 - Release Notes are used to document the history of changes between updates