

# PHYTEC

Agile Lösungen  
für hochdynamische Prozesse.

# Software Life-Cycle-Management



Smarter. Faster. Easier.

# SOFTWARE LIFE-CYCLE-MANAGEMENT

Nachhaltige und verbindliche Pflege von Software über den Produktlebenszyklus gewinnt immer mehr an Bedeutung. Systeme, die in irgendeiner Form mit dem Internet Berührung haben, müssen sich stets verändernden Sicherheitsbedrohungen stellen.

Phytec bietet eine klare Strategie zur Pflege der BSPs für Standardprodukte und für kundenspezifische Varianten. Mehr noch: Phytec bietet eine enge und sehr direkte Abstimmungsmöglichkeit mit den Kunden über sogenannte „Approved Projects“.

**Um unsere Vorgehensweise möglichst nachvollziehbar zu machen, ist es wichtig, einige Ausgangsüberlegungen und Abwägungen zu kennen:**

*Phytec bietet zu jedem Modul eine Kit-Variante an. Hierbei handelt es sich um eine Standardvariante mit meist maximalem Ausbau. Darüber hinaus definieren wir eine möglichst geringe Anzahl weiterer Standard-Versionen. Die Pflege unserer Software umfasst in erster Linie die Standardversionen. Das Bestreben nach einer geringen Zahl von Varianten optimiert deren Verfügbarkeit und fokussiert unsere Aufwände.*

*Unsere Module sind typischerweise viele Jahre für Neudesigns von Interesse. Über einen solchen Zeitraum kristallisieren sich gewisse Vorzugstypen aus den Anwendungen unserer Kunden heraus. Größere Speicher und schnellere Taktraten werden zum Standard und lösen frühere Versionen ab. Um dem geänderten Bedarf und der Weiterentwicklung der Komponenten Rechnung zu tragen, passen wir in geeigneten Zeiträumen unsere Standardversionen an. Neue Konfigurationen werden zum Standard und unsere Softwarepflege konzentriert sich dann auf diese neuen Standardversionen. Damit steht Ihnen für Ihre Neudesigns immer das Optimum zur jeweiligen Zeit zur Verfügung.*

*Aber keine Sorge: Wir nehmen keine Varianten vom Markt, für die es Bedarf gibt. Und Sie können sich die Langzeitpflege einer bestimmten Standardvariante, für die Sie sich einmal entschieden haben, sichern.*

## Approved Projects

Sie wollen Standardversionen unserer Module einsetzen und sicher sein, dass wir für genau diese Variante die Software mindestens solange weiterentwickeln, bis Sie stabil in der Serie sind – eben bis Sie Ihre Design-in-Phase abgeschlossen haben? Dann melden Sie Ihr Projekt bei uns an. Wir haben so eine Chance, Sie auf möglicherweise attraktivere Varianten oder anstehende Änderungen zu informieren, bevor Sie starten. Wir kennen dann auch Ihre Zeitplanung und können uns über die Produktpflege optimal mit Ihnen abstimmen. Wenn wir diese gemeinsame Basis miteinander etabliert haben, sagen wir Ihnen unsere Unterstützung zu (selbst wenn die gewählte Variante nicht mehr zu unseren Vorzugsvarianten zählen sollte). Sie haben dann ein „Approved Project“ mit uns.

*Die BSP-Pflege findet in mehreren Phasen statt:*

### Design-in

- Zu allen Standardmodulen existiert ein serientauglicher Bootloader.
- Bei Auslieferung von Standardmodulen wird standardmäßig nicht notwendigerweise die neueste Version des Bootloaders aufgespielt.
- Sind Bootloader-Anpassungen erforderlich, um die einwandfreie (serientaugliche) Funktion des Moduls zu gewährleisten, so wird der Bootloader bei Standardmodulen angepasst (z.B. neue Timing-Werte bei RAM-Änderungen).

*Kunden werden über unterschiedliche Informationskanäle informiert, sind jedoch gehalten, die korrekte Funktion im Kundenkontext zu verifizieren. Derartige Änderungen sind üblicherweise die Folge von Bauteiländerungen und gehen mit einer Versionserhöhung einher. Beim Einsatz eigener Bootloader sind die gemachten Anpassungen durch die Kunden nachzuziehen.*

- Zu allen als Vorzugsvarianten gekennzeichneten Standardmodulen stehen BSP-Implementierungen gemäß unseren hohen Qualitätsansprüchen zur Verfügung. Der Funktionsumfang ist jeweils beschrieben.

*Dasselbe tritt (möglicherweise auf das jeweilige Projekt angepasst) für Standardmodule zu, die in „Approved Projects“ eingesetzt werden.*

- Jährlich gibt es mindestens ein Release; wir streben zwei Releases an. Hierbei bevorzugen wir LTS-Kernelversionen (long-term-support).

*Gerade bei neuen Controllern bietet ein Hersteller-Kernel (nicht unbedingt mit LTS-Kernelversion) jedoch die bessere Feature-Unterstützung und wird für den Design-in Prozess von uns üblicherweise bevorzugt.*

- Das Release beinhaltet auch eine möglichst aktuelle Version von Yocto. Aufgrund des Versatzes von Verfügbarkeiten von Kernel- und Yocto-Neuver-sionen kann es hier im worst Case zu einem Versatz von einem Jahr kommen.
- Wir streben an, möglichst nahe an Mainline zu kommen. Dies hilft uns bei der Weiterentwicklung und vor allem bei der späteren Pflege der BSPs für alle Projekte, die dann in Serie sind.

### Übergang Design-in zu Serienphase

Ein LTS-Kernel hat eine typische Supportzeit von zwei bis maximal drei Jahren. Der Supportzeitraum einer Yocto-Version beträgt typisch zwei Jahre. Hier sprechen wir vom Supportzeitraum durch die Community. Wie wir mit Updates innerhalb dieser Supportperiode umgehen, wird weiter unten beschrieben.

- Wird ein Produkt/Plattform nicht mehr für Neudesigns empfohlen, so wird dennoch das BSP noch für einen Zeitraum von zwei Jahren weiterentwickelt und entsprechende neue Versionen werden veröffentlicht.

*Siehe hierzu die oben gemachten Ausführungen zu Vorzugsvarianten der Standardmodule und zu „Approved Projects“.*

- Alle BSPs unserer Standardentwicklung werden intensiven und weitgehend automatisierten Tests unterzogen.

### Pflege älterer LTS-Versionen

- Jedes Jahr gibt es zwei Pflegezyklen, in denen Updates von älteren, aber noch im Support befindliche (LTS)-BSP-Stände veröffentlicht werden. Hierbei unterscheiden wir zwei Fälle:

- Die entsprechenden Fixes der Community bauen sich fehlerfrei und das Ergebnis durchläuft unsere Tests ohne Mängel. -> Direkte Veröffentlichung
- Beim Bauen oder bei den Tests ergeben sich Fehler. -> Hier bemühen wir uns, die Fehlerursache zu finden und zu beheben. Jedoch ist der hierfür von uns erbrachte Aufwand sehr begrenzt (max. 3 Manntage Aufwand).

*Die Ursache für auftretende Fehler kann im Zusammenspiel zwischen den offiziellen Fix-Releases und verwendeten Patches oder in der Verwendung spezieller Treiber liegen. Bei den Fix-Releases muss in Abwägung des Aufwandes und der damit verbundenen Produkt-Zusatzkosten hier ein klarer Schnitt gemacht werden. \**

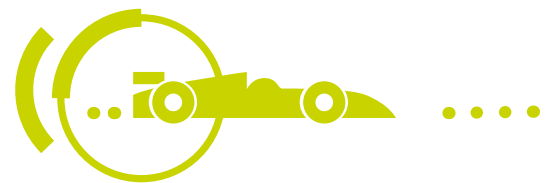
### Vorgehensweise bei kundenspezifischen Varianten / kundenspezifischer Hardware

Alle oben genannten Dienste werden für kundenspezifische Varianten gegen Entgelt ebenfalls angeboten. Unsere Kunden profitieren hier von unseren Vorleistungen bei der Pflege unserer Standardprodukte und von unseren Testverfahren und -Einrichtungen.

Entsprechende Pakete werden derzeit neu definiert, inkl. Preisgestaltung. Sprechen Sie uns hier bitte aktiv an und verfolgen Sie entsprechende Informationen auf unseren Webseiten.

### Ein wichtiges Zusatzangebot wird wie folgt aussehen:

- Wir bieten die Aufnahme kundenspezifischer Versionen in unsere automatisierte Testumgebung (und passen diese auch ggf. an). Ein (kostenpflichtiger) Service besteht darin, neue Releases von uns auf das Testsystem des Kunden anzuwenden. Als einfachste Stufe würden wir den Kunden über auftretende Probleme informieren und die weitere Vorgehensweise mit ihm abstimmen.



#### \* Individuelle Lösungen:

-> Wir veröffentlichen das Ergebnis und suchen dann im Einzelfall mit betroffenen Kunden eine gemeinsame und individuelle Lösung, die dann auch separat abgerechnet wird. Wir sind überzeugt, dass dies die faireste Lösung darstellt: Sind viele Kunden betroffen, verteilen sich die Aufwände; ist mit hohen Aufwänden (pro Kunde) zu rechnen; so sind sehr wahrscheinlich andere Strategien zielführender (gerade im Kundeninteresse). Naheliegender ist eine Umstellung auf eine aktuelle Version. Dieses Abwägen ist im Kundeninteresse, da zu erwarten ist, dass sich die fehlerverursachende Grundlage in die Zukunft fortpflanzt und zu immer höheren Aufwänden führen kann.

# Bootstrategie, Updatekonzepte

Phytec setzt Barebox als Bootloader ein. Unseres Erachtens soll ein Bootloader i.W. nur eine zentrale Funktion haben: Das sichere Laden eines Betriebssystems. Dies soll von den Medien möglich sein, die vom jeweiligen Controller unmittelbar unterstützt werden. Zum ordnungsgemäßen Laden eines Betriebssystems sind auch einige Initialisierungsarbeiten (z.B. RAM Parameter Einstellungen) erforderlich.

Wir empfehlen, höhere Funktionalitäten, wie z.B. Anzeigen eines Splash-Screens oder die Implementierung von Update-Strategien, bereits unter der Regie eines Betriebssystems durchzuführen. Hierzu bieten wir aus unserem „Werkzeugkasten“ fallbezogene Lösungen, die wir auf Ihre Erfordernisse anpassen können. Um sehr schnell nach dem Einschalten eines embedded Systems reagieren zu können, empfehlen wir das Starten eines Minimalsystems (z.B. Linux Kernel mit einem sehr kleinen Root-File-System; genau zugeschnitten für anstehende Verwaltungsaufgaben wie eben die genannten Updates). Im Gegensatz zur Realisierung der Arbeiten im Bootloader stehen unter Linux höhere Protokolle zur Verfügung; im Bootloader selbst müssten komplexe Funktionalitäten sonst mühsam programmiert und gepflegt werden.

Signierte Updates von unterschiedlichen Medien – auch solchen, die als Bootquelle vom Controller nicht unterstützt werden – können ebenso realisiert werden wie komplexe Updateszenarien aus der Cloud. Redundanzen zum Recovery bei auftretenden Fehlern im Updateprozess selbst sind einfach realisierbar. Möglicherweise gewünschte Benutzerführung für diese Prozesse lässt sich ebenfalls deutlich komfortabler realisieren, da – wie schon erwähnt – die ganzen Tools eines Betriebssystems genutzt werden können und entsprechende Peripherie (z.B. Display, Touch) einfach ansprechbar ist. Doppelte Anpassungsarbeiten entfallen.

Es gibt durchaus auch Szenarien, bei denen etwas mehr Intelligenz in den Bootloader verlagert werden muss: Ein Beispiel wäre der durch Redundanz gesicherte Ersatz auch des „kleinen Starthilfe-Linux“. Auch hier empfehlen wir, die Funktionalität im Bootloader auf das notwendige Minimum zu reduzieren. Entsprechende Lösungen können wir – zugeschnitten auf Ihren spezifischen Bedarf – ebenfalls erstellen.

# Vorleistungen & individuelle Softwareerstellung im betriebssystem- nahen Umfeld

Je größer die allgemeine Verwendbarkeit einer Funktionalität ist, desto höher ist unser Vorleistungsgrad. Die Unterstützung der Modul-Eigenschaften ist ein zentrales Element der Betriebssystemimplementierung und steht im Vordergrund unserer BSP-Arbeiten. Auch im Bereich Middleware ist die Frage der breiten und möglichst stets identischen Nutzung die Leitlinie für die Entscheidung zur breiten Unterstützung durch unsere Software-Entwicklungsmannschaft.

In vielen Fällen ist eine Technologie nur in angepasster Form geeignet, Ihre Aufgabenstellung zu lösen. Wir bieten einen großen Werkzeugkasten derartiger Technologien auf Betriebssystemebene oder im betriebssystemnahen Umfeld. In solchen Szenarien besteht unsere Vorleistung darin, effizient und kostengünstig Ihre individuelle Anforderung unter Verwendung eines möglichst hohen Anteils von Vorleistung realisieren zu können. In neue Basistechnologien sind wir bereit, mit zu investieren, wenn uns diese unseren „Werkzeugkasten“ sinnvoll erweitert.

Ihre Vorteile aus dieser Strategie einer geschickten Mischung von Vorleistung und individueller Realisierung liegt in der Kosteneffizienz Ihres Gesamtprozesses. Wir halten die hardware-bezogenen, allgemeinen Softwareentwicklungskosten niedrig und können Sie dennoch auf Basis von Grundsatzarbeiten und jeder Menge Projekterfahrung ganz gezielt unterstützen. Hierbei kommt Ihnen die enge Verzahnung unseres Wissens aus Hardware und Software zugute.

