



www.thalesgroup.com/germany

Erfahrungen mit agilen Entwicklungsmethoden

Dipl.-Phys. Stefan Thiele

Universität der Bundeswehr München, 16. Januar 2012

Sicherheit und Mobilität in einer vernetzten Welt.

THALES

- 1 Verteidigungsbereich von Thales in Deutschland
- 2 Systemübersicht
- 3 Unser Weg zur agilen Software-Entwicklung
- 4 Entwicklungsumgebung und Umfeld
- 5 Vorgehensmodelle
- 6 Fazit

1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit



Pforzheim

417 Beschäftigte

- ◆ Bodenüberwachungsradare
- ◆ Norumat
- ◆ SDR
- ◆ Taktische Funksysteme
- ◆ SOA
- ◆ Verschlüsselung
- ◆ Galileo Bodensegment



Wilhelmshaven

99 Beschäftigte

- ◆ Combat-Management-Systeme
- ◆ Integrated Logistic Support
- ◆ Advanced Naval Training Systems



Koblenz

156 Beschäftigte

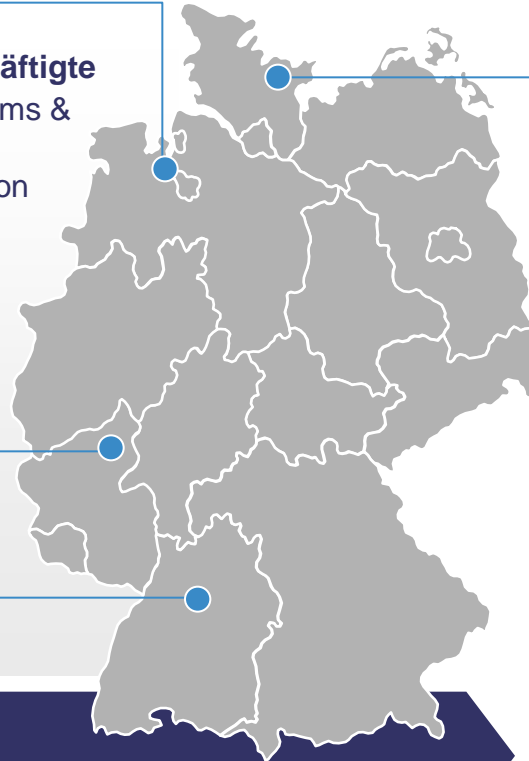
- ◆ Small Arms & Tactics Simulation



Kiel

150 Beschäftigte

- ◆ Integrated Navigation Systems
- ◆ Integrated Communication Systems
- ◆ Torpedo Control Systems
- ◆ Optronic Systems



~ 850 Mitarbeiter an 4 Standorten

1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

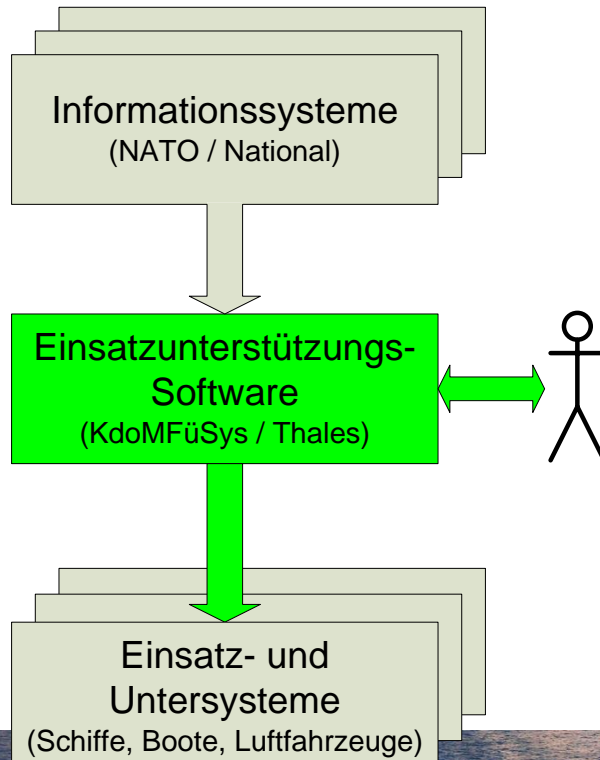
3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit

Einsatzunterstützung der Deutschen Marine



- ◆ Verarbeitung vielfältiger Informationen über proprietäre Eingangsschnittstellen aus unterschiedlichen Kontexten
- ◆ Spezifischer Workflow und Verarbeitungskontexte innerhalb des Systems
- ◆ Versorgung der Einsatz- und Untersysteme über proprietäre, herstellerspezifische Schnittstellen

Einsatzunterstützung der Deutschen Marine

◆ Systemumfang

- 30 Arbeitsplätze
- 4 Eingangsschnittstellen (binär, ASCII, MS-Access)
- 11 Ausgangsschnittstellen (binär, ASCII)
- 1 abgesetztes System
- 8 spezielle Reports
- 1 digitales Produkt mit eigener Präsentationslösung

◆ Funktionalitäten

- Verarbeiten grundlegender Informationen
- Planen von Einsätzen, Manövern, etc.
- Erstellen von einsatzspezifischen Produkten

1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit

Evolution statt Revolution (1/3)

◆ Einsatz des V-Modells 92 und 97

- Verpflichtender Einsatz des V-Modells 9x in den Projekten mit der Bundeswehr
- Zeitaufwändige Spezifikationsphase während der Projektlaufzeit
- Komplizierter Umgang mit Änderungen nach der Spezifikationsphase
- Ergebnis: Kunden- oder Nutzererwartungen wurden teilweise nicht erfüllt
- Folge: Komplizierte oder konfliktgefährdete Systemabnahmen am Ende des Projektes

◆ Spezielle Problemlage

- Unklare und/oder wechselnde Anforderungen
- Fehlende oder falsche Schnittstellendokumente

◆ Lösungsversuche

- Vorschalten von Studienprojekten um die Anforderungen des Nutzers zu klären und zu dokumentieren
- Aufteilen des gesamten, geplanten Leistungsumfangs auf Teilprojekte, um Korrekturmöglichkeiten und neue Bewertungsspielräume zu gewinnen

Evolution statt Revolution (2/3)

◆ Einführung des V-Modells XT

- Neben dem klassischen Wasserfall-Modell wird eine iterative Vorgehensweise ermöglicht
- Weiterhin aber sehr dokumentationslastig
- Extremes Tailoring in Abstimmung mit dem öAG / Nutzer möglich
- Ergebnis: Kunden- oder Nutzererwartungen lassen sich besser erfüllen, wenn auch teilweise mehrere Anläufe notwendig sind
- Folge: Systemabnahme am Ende des Projektes meist problemlos

◆ Angepasste Prozesse in der Entwicklung

- Vermeidung von monolithischer Dokumentation
- Frühzeitige Fokussierung auf kritische Systemelemente
- Feedback-Prozess mit dem Nutzer durch inkrementelle Einführung des Systems oder neuer Funktionalität
- Größtmögliche Automatisierung des Entwicklungsprozesses

Evolution statt Revolution (3/3)

◆ Problemlage

- Beschaffungsauftrag von teilweise fragwürdig spezifizierter Leistung bei gleichzeitiger Budgetbegrenzung
 - Führt zu Konflikten bei zu niedrigem Budget
 - a) AN überschreitet geplantes Budget und hat ein wirtschaftliches Problem
 - b) AN erstellt System wie spezifiziert, d.h. das System ist für den Nutzer nicht oder nur in Teilen verwendbar
- Iterative Vorgehensweise steht im Widerspruch zu konkreter Vertragslage bei Beschaffungsaufträgen des öAG
- Nutzer ist nicht Auftraggeber

◆ Erfolgsfaktoren

- Frühzeitige Nutzer-Einbindung
- Vertrauensvoller Umgang aller Beteiligten
- Wille zum Erreichen eines gemeinsamen Ziels

Leitmotiv / Motivation

„Da die Fachseite erst weiß,
was sie will, wenn sie sieht, was
sie kriegt, bekommt sie was sie
verdient.“

(unbekannter Autor)

1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit

Agile Vorgehensweise im Projektteam

◆ Spezifikation

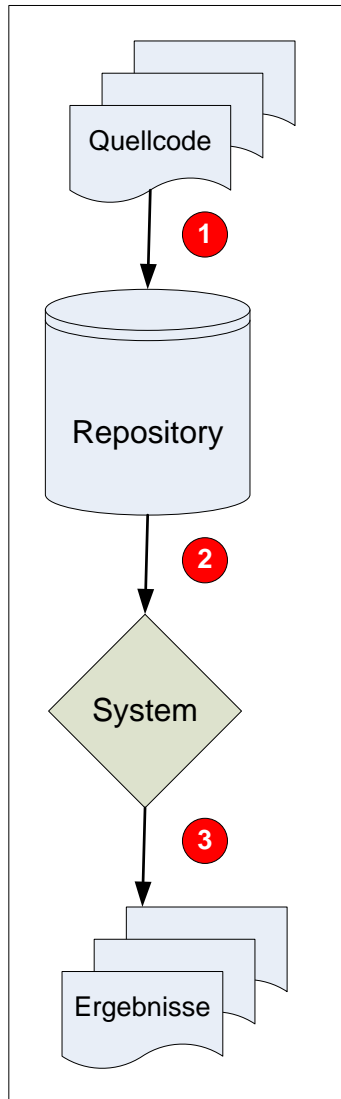
- Vertraglich werden die geforderten Funktionalitäten (Features) auf verlässlicher Basis vereinbart
- Priorisierung der Features gemeinsam mit dem Nutzer
- Grobplanung der Iterationen über die gesamte Vertragslaufzeit
- Für die aktuelle und mindestens die folgende Iteration
 - Detaillierte Spezifikation der geplanten Features
 - Abstimmung mit Nutzer und Entwicklern
 - Dokumentation im Projekt-Wiki



◆ Arbeitspakete

- Planung von Arbeitspaketen und Erstellung im Bug-/Task-Tracking Werkzeug
 - Keine konkrete Zuweisung von Arbeitspaketen an Entwickler
 - Regelmäßige Fortschrittskontrolle





Entwicklung und Systemerstellung

◆ Continuous Integration

- Neue oder geänderte Quellen führen zu einem kompletten Systembau und Test (1)
- Automatisierte Erstellung (2)
- Automatisiertes Deployment
- Mehrstufige automatisierte Tests
- Automatische Ergebnisdokumentation (3)
 - Feedback für Entwickler, Qualitätssicherung und Projektleitung

◆ Bisherige Erfahrungen

- Sehr gute Transparenz des EW-Prozesses
- Ständige Lieferfähigkeit
- Fehler werden frühzeitig erkannt



Jenkins

maven



QF-TEST



1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit

Agile Vorgehensweise auf der Zeitachse

◆ Modell 1

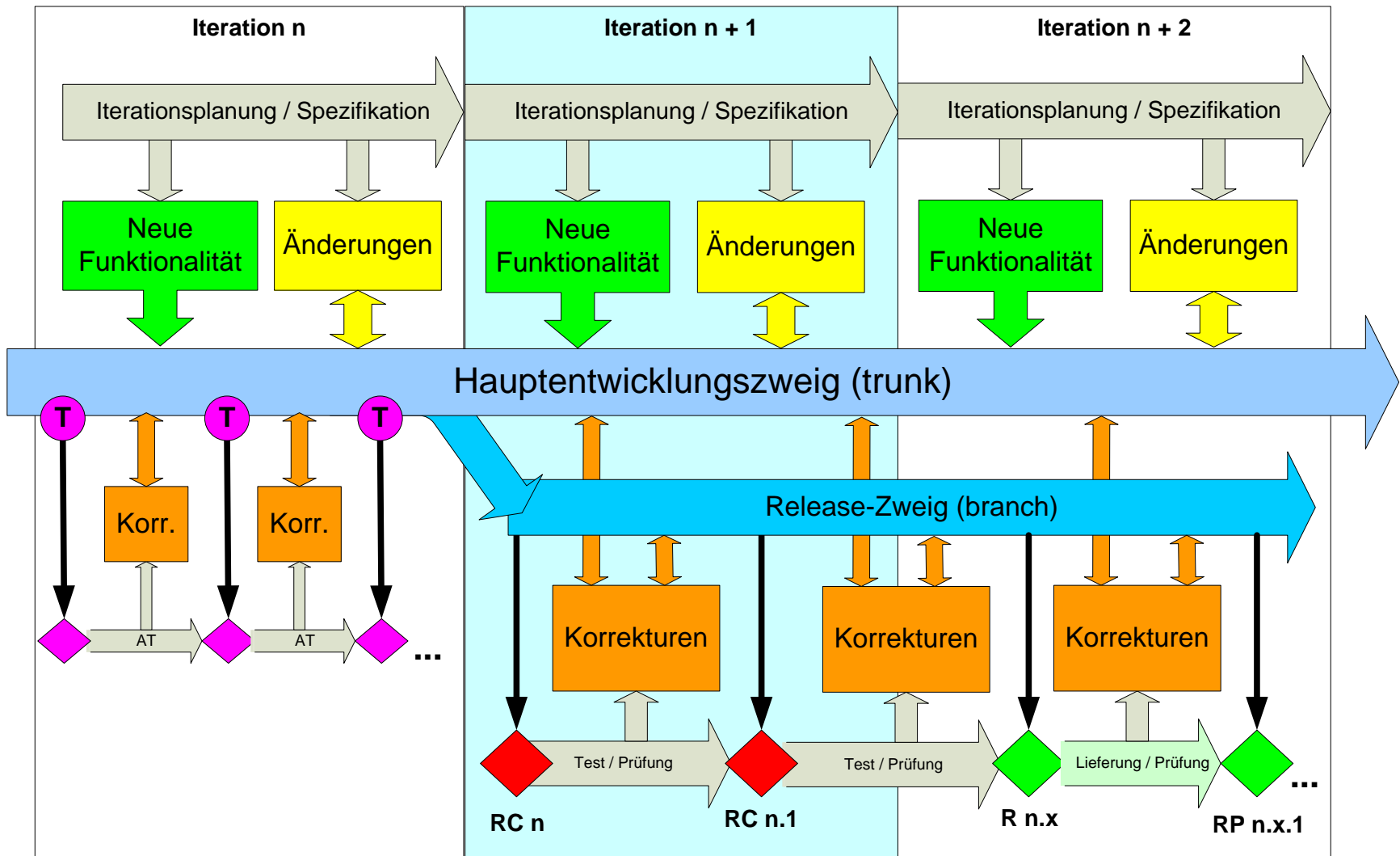
- Alle Entwickler arbeiten ständig in ein gemeinsames Quellcode-Repository
- Zum geplanten Zeitpunkt (Ende der Iteration) wird ein Release-Branch erzeugt, aus dem ein Kunden-Release generiert wird

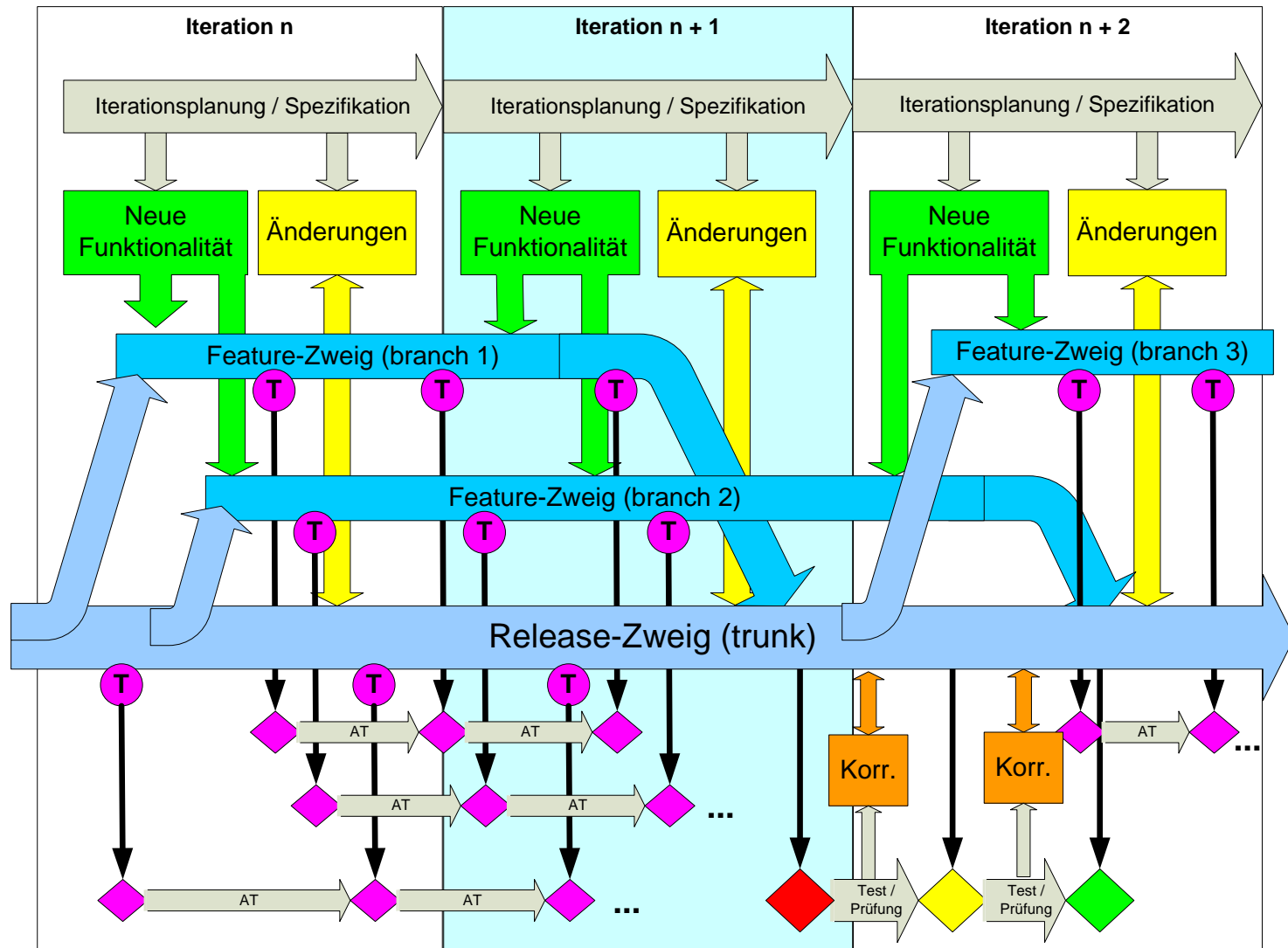
◆ Modell 2

- Entwickler arbeiten in einem oder mehreren dedizierten Feature-Branch(es)
- Das Kunden-Release wird immer aus dem Hauptzweig des Quellcode-Repository erstellt

◆ Gemeinsamkeiten der Modelle

- Nach manueller Prüfung und Fehlerbehebung wird das erstellte System beim Nutzer installiert, geprüft und in die Nutzung überführt
- Ständige automatisierte Test sorgen für eine frühzeitige Fehlererkennung
- Automatisierte Oberflächen-Tests sorgen für eine frühzeitige Erkennung von unerwünschten Nebenwirkungen durch neue oder geänderte Funktionalität





Erfahrungen mit den Modellen

◆ Modell 1 (Release-Branch)

- ✓ Wenig Konfigurations- und Administrationsaufwand
- ✗ Bei größeren Änderungen ist die ständige Lieferfähigkeit nicht gegeben
- ✗ Refactoring des Systems behindert die Erstellung neuer Funktionalität
- ✗ Vom Nutzer gemeldete Fehler müssen parallel in zwei Branches korrigiert werden

◆ Modell 2 (Feature-Branches)

- ✓ Ständige Lieferfähigkeit aus dem Hauptzweig möglich
- ✓ Vom Nutzer gemeldete Fehler müssen nur im Hauptzweig korrigiert werden
- ✓ Größere Änderungen können unabhängig voneinander ohne gegenseitige Störungen implementiert und getestet werden
- ✗ Je nach Anzahl der Feature-Branches entsteht extrem hoher Konfigurations- und Administrationsaufwand
- ✗ Beim Zusammenführen der Branches entsteht manueller Aufwand und es können leicht Fehler passieren

◆ Trotz der größeren Anzahl negativer Aspekte arbeiten wir vorwiegend nach dem erstem Modell

1 Verteidigungsbereich von Thales in Deutschland

2 Systemübersicht

3 Unser Weg zur agilen Software-Entwicklung

4 Entwicklungsumgebung und Umfeld

5 Vorgehensmodelle

6 Fazit

Agile Softwareentwicklung ...

- ◆ bringt für beide Vertragsparteien Sicherheit für die spätere Systemabnahme
- ◆ erfordert vertrauensvolle Zusammenarbeit der Vertragsparteien
- ◆ erfordert eine angepasste Entwicklungsumgebung mit „kontinuierlicher automatischer Systemintegration“ und weitestgehend automatisierter Testumgebung

Unsere Erfahrungen und Empfehlungen

- ◆ Investitionen in teure Werkzeuge sind nicht erforderlich, die frei verfügbaren sind auch für große Systeme ausreichend
- ◆ Standard-Vorgehen oder Lehrbuch-Arbeitsweisen müssen an die eigenen Verhältnisse angepasst werden
- ◆ Das Vorgehensmodell muss ständig überprüft und korrigiert werden
- ◆ Der initiale Aufwand darf nicht unterschätzt werden
- ◆ Agile Vorgehensweise ersetzt nicht das Verstehen der Anwenderdomäne

Fragen?

Vielen Dank für Ihre Aufmerksamkeit

Kontakt

Dipl.-Phys. Stefan Thiele
Thales Defence & Security Systems GmbH
Edisonstraße 3
24145 Kiel
stefan.thiele@thalsgroup.com
www.thalesgroup.com/germany