

# ESE Kongress 2017

Vortragsskript:

## Anforderungen – Eine Checkliste zur Reife Requirements Engineering und Management – Was kann ich tun?

Thomas Batt, MicroConsult GmbH

### Vorwort

Anforderungen zu erfassen und zu verwalten ist ein wesentlicher Schlüssel zu erfolgreichen Projekten. Egal ob im klassischen oder agilen Prozessumfeld - professionelles Requirements Engineering und Management für Embedded- und Echtzeitsysteme verkürzt Ihre Projektlaufzeiten und spart Entwicklungs- sowie Wartungskosten ein. Stellen Sie sich dieser Herausforderung!

Vor dem Hintergrund meiner beruflichen Erfahrungen sowie der Durchführung vieler Seminare und Coachings zum Thema Requirements Engineering und Management ist die dem Beitrag zugrunde liegende Checkliste entstanden.

Requirements Engineering und Management

Checksite	Requirements Engineering und Management für Embedded- und Echtzeitsysteme			
	Themen	Themenpunkte	Bearbeitungsstatus	
			JA NEIN NA	
Requirements Engineering (Anforderungen identifizieren, dokumentieren und überprüfen)	Anforderungen (WAS) von Umsetzung (WIE) trennen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungskontext identifizieren und darstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen textuell erfassen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungstemplates anwenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Natürlichsprachliche Methode anwenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungsqualitätsmerkmale definieren und berücksichtigen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Produkt-, Projekt- und Plattformanforderungen differenzieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Funktional, nichtfunktional und Zusicherung differenzieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Qualitätsanforderungen definieren und erfassen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	(Abstrakte) Qualitätsmerkmale quantisieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	(Abstrakte) Qualitätsmerkmale testbar / nachweisbar gestalten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Aus Standards gültige Requirements ableiten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungsglossar als Teil des Projektglossars pflegen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prozess, Vorgehen, Methode	Anforderungen auf allen Entwicklungsebenen etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Workflows, Aktivitäten, Artefakte und Rollen definieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Verständnis und Commitment zu Anforderungen schaffen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Ist-Zustand der bisherigen Produkte analysieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Stakeholder identifizieren und analysieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Stakeholder auswählen und dokumentieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Stakeholder-Interviews durchführen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen aus Interview-Ergebnissen ableiten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen mit Auftraggeber / Stakeholder abstimmen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen analysieren, modellieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen simulieren, ausführen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungsmachbarkeit prüfen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Prototyp (Wegwerf- oder evolutionär) erstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	QA: Reviews durchführen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	QA: Funktionalszenarien und Qualitätsszenarien ausführen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Konsistenz zwischen Anforderungen und Umsetzung gewährleisten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Abnahmekriterium / Test-Case für jede Anforderung entwickeln	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Kontinuierliches, proaktives Risikomanagement etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Änderungsprozess für Anforderungen etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aus Fehlern lernen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Prozess, Vorgehen, Methode u. Tooling kontinuierlich verbessern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requirements Management (Anforderungen verwalten, verfolgen und verlinken)	Anforderungen wiederverwenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungen parametrieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Anforderungseigenschaften (Properties / Attribute) definieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Versionierung von Anforderungen etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Baselining / Releases für Anforderungen etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Bidirektionale Traceability über alle Entwicklungsschritte etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requirements Management Tools	Templates für Artefakte bereitstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Text vers. Application/Product Life Cycle (A/PLM) Tools evaluieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Tool anforderungsbasierend auswählen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Tool integrieren, testen und schulen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Tool in täglicher Arbeit etablieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bild 1: Checkliste - Requirements Engineering und Management

Die Checkliste zeigt Ihnen, was Sie zum Thema Anforderungen in Ihrem Unternehmen einführen, durchführen und optimieren können. Sie ist gleichermaßen eine Fundgrube wertvoller Tipps aus der Praxis.

Der folgende Beitrag greift alle Themenpunkte der Checkliste der Reihe nach auf und beleuchtet den Hintergrund dazu. Wichtig für das Verständnis: Wo der Begriff „System“ verwendet ist, kann dieser durch andere Domänen, wie beispielsweise Embedded-Software oder -Hardware, ersetzt werden.

Die Requirements Engineering und Management Checkliste zusammen mit der dazugehörigen Dokumentation dazu haben wir unter [1] für Sie zum Download bereitgestellt.

## Requirements Engineering

(Anforderungen identifizieren, dokumentieren und verwalten)

### Anforderungen (WAS) von Umsetzung (WIE) trennen

Grenzen Sie gedanklich, schriftlich und organisatorisch Anforderungen von deren Umsetzung ab. Anforderungen meinen das WAS und die Umsetzung das WIE. Nur in wenigen begründbaren Fällen enthalten Anforderungen kein WIE.

### Anforderungskontext identifizieren und darstellen

Grenzen Sie das anfordernde System gegen dessen Kontext ab, um die externen Schnittstellen und damit die zugehörigen Anforderungen zu identifizieren.

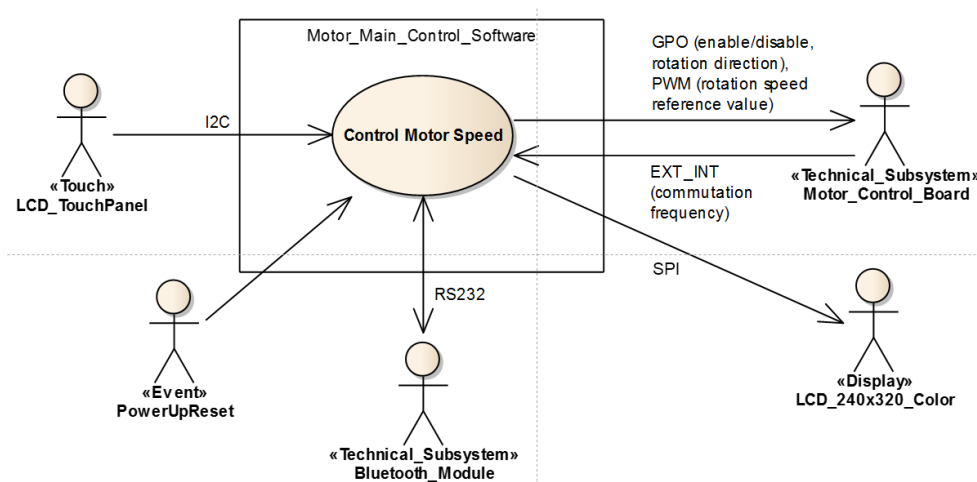


Bild 2: Kontextsicht mit dem Use-Case Diagramm

Als grafische Notation dazu eignet sich für eine Kontextsicht das Use-Case Diagramm aus der UML (Unified Modeling Language) bzw. aus der SysML (Systems Modeling Language) [5].

### Anforderungen textuell erfassen

Die Basis für alles weitere in der Entwicklung und im Test sind die Anforderungen. Sie sollten diese textuell erfassen und ggf. mit anderen Darstellungsformen wie Grafiken und Diagrammen ergänzen. Eine alternative Notationsmöglichkeit, mit der Sie die textuelle Anforderungserfassung komplett ersetzen können, ist bis heute (2017) nicht erfunden.

## Anforderungstemplates anwenden

Verwenden Sie Anforderungstemplates [8], um bereits beim Schreiben der Anforderungen einen sehr hohen Qualitätslevel der Anforderung zu erreichen.

Anforderungstemplate für selbstständige Systemaktivität:

*[Wann?] [Unter welcher Bedingung?] MUSS | SOLLTE DAS SYSTEM  
<Objekt und Ergänzung des Objektes> <Prozesswort im Infinitiv>.*

Anforderungstemplate für interaktive Systemaktivität:

*[Wann?] [Unter welcher Bedingung?] MUSS | SOLLTE DAS SYSTEM <wem?> DIE MÖGLICHKEIT  
BIETEN, <Objekt und Ergänzung des Objektes> <Prozesswort im Infinitiv>.*

Anforderungstemplate für potentielle Systemaktivität:

*[Wann?] [Unter welcher Bedingung?] MUSS | SOLLTE DAS SYSTEM FÄHIG SEIN,  
<Objekt und Ergänzung des Objektes> <Prozesswort im Infinitiv>.*

## Natürlichsprachliche Methode anwenden

Um die Qualität der Anforderungen nochmals zu steigern, wenden Sie ergänzend zu den Anforderungstemplates die natürlichsprachlichen Methode [8] an. Dadurch erreichen Sie eine Rücktransformation der beim Wissensausdruck entstehenden Tilgungen, Generalisierungen und Verzerrung.

## Anforderungsqualitätsmerkmale definieren und berücksichtigen

Jede Anforderung muss die folgenden Qualitätsmerkmale erfüllen: vollständig, korrekt, verstehbar, notwendig, konsistent, verfolgbar, machbar, Verbindlichkeit klassifiziert, testbar und [optional] atomar.

## Produkt-, Projekt- und Plattformanforderungen differenzieren

Machen Sie sich dieser Anforderungskategorien bewusst:

*Produktanforderungen:*

Anforderungen, die Leistungsmerkmale, Fähigkeiten und Eigenschaften des aktuell zu entwickelnden Produktes beschreiben

*Projektanforderungen:*

Von der Projektleitung kommende Anforderungen, beispielsweise Zeiten, Kosten, Lieferumfang und Logistik

*Plattformanforderungen:*

Anforderungen, die für mehrere Produkte bzw. für diese Produktgruppe immer gültig sind → Wiederverwendbarkeit von Anforderungen

## Funktional, nichtfunktional und Zusicherung differenzieren

Machen Sie sich dieser Anforderungsarten bewusst:

*Funktionale Anforderungen:*

Nach außen hin sichtbare / beobachtbare Funktionalitäten

*Nichtfunktionale Anforderungen:*

Qualitäten, Qualitätsmerkmale einzelner funktionaler Anforderungen oder des gesamten Systems

*Zusicherungen als Anforderungen:*

Standards, Normen, Prozesse, technische Zusicherungen, Projektanforderungen

## Qualitätsanforderungen definieren und erfassen

Beschreiben Sie nicht nur funktionale Anforderungen, sondern auch deren Qualitäten und genauso wie die Qualitäten für das gesamte System.

### (Abstrakte) Qualitätsmerkmale quantisieren

Quantisieren Sie abstrakte Qualitätsmerkmale wie beispielsweise Wiederverwendbarkeit. „Die Embedded-Software muss wiederverwendbar sein.“ → „Die Embedded-Software muss mit dem Grad 4 wiederverwendbar sein.“ Dabei müssen Sie die verschiedenen Grade einmalig spezifizieren. Daraus geht hervor, was die Embedded-Software erfüllen muss.

Bei konkreten Qualitätsmerkmalen, wie beispielsweise dem Leistungsverbrauch eines Systems, können Sie direkt den Wert mit Einheit setzen.

### (Abstrakte) Qualitätsmerkmale testbar / nachweisbar gestalten

In der oben beschriebenen Gradspezifikation beschreiben Sie zusammen mit dem Testteam auch die Testbarkeit / Nachweisbarkeit für die Erfüllung des (abstrakten) Qualitätsmerkmals.

Bei konkreten Qualitätsmerkmalen, wie beispielsweise dem Leistungsverbrauch eines Systems, ist die Testbarkeit / Nachweisbarkeit meist klar.

### Aus Standards gültige Requirements ableiten

Standards und Normen enthalten nicht immer direkt die Anforderungen für das Produkt. Vielmehr lassen die Texte häufig einen gewissen Interpretationsspielraum zu. Umso wichtiger ist es, dass Sie einmalig aus den für Ihre Produkte gültigen Standards und Normen die konkreten Anforderungstexte ableiten und diese gemeinsam mit den Beratungsstellen der zertifizierenden Organisationen vor der Umsetzung abstimmen.

### Anforderungsglossar als Teil des Projektglossars pflegen

Wenn Sie in den Anforderungstexten Substantive, Verben, Adjektive oder Abkürzungen verwenden, die bei Stakeholdern nicht geprägt sind, müssen Sie diese Begriffe in einem Glossar erklären. Führen Sie ein einziges Glossar anstelle mehrerer verschiedener für das gesamte Projekt → Vermeiden von Redundanz und Inkonsistenz!

## Prozess, Vorgehen, Methode

### Anforderungen auf allen Entwicklungsebenen etablieren

Führen Sie Requirements Engineering und Management nicht nur für Kunden- und Systemanforderungen ein, sondern für alle Entwicklungsdomänen, z.B. Subsysteme, Hardware, Embedded-Software, Software, Konstruktion, usw.

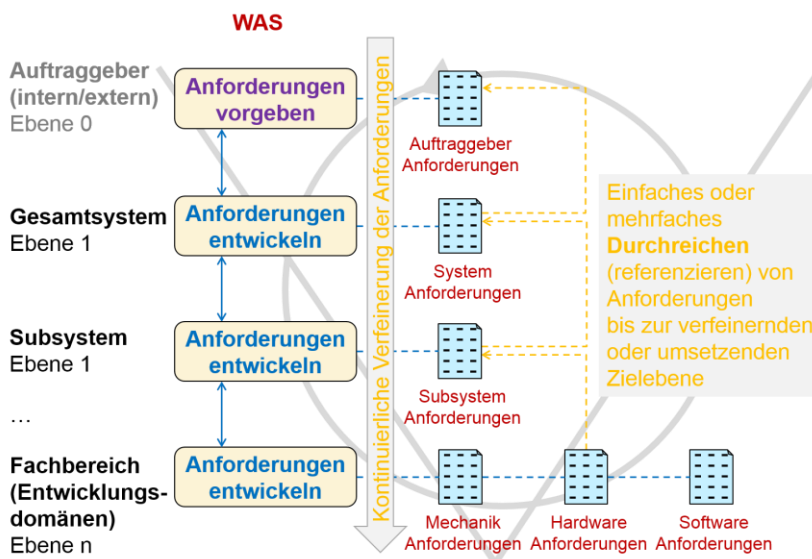


Bild 3: Anforderungsebenen

### **Workflows, Aktivitäten, Artefakte und Rollen definieren**

Egal ob Sie traditionelle [3] oder agile [4] Prozesse leben, definieren Sie eindeutige Workflows mit genau benannten Aktivitäten, deren ein- und ausgehenden Artefakten und deren verantwortlichen Rollen.

### **Verständnis und Commitment zu Anforderungen schaffen**

Insbesondere bei der Unternehmenseinführung von Requirements Engineering und Management in Pilotprojekten ist es wichtig, ein gemeinsames Verständnis sowie ein hohes Maß an Motivation, Wissen und Zustimmung unter den Beteiligten zu schaffen. Zweifler involvieren Sie punktuell, z.B. in Engineering Meetings.

### **Ist-Zustand der bisherigen Produkte analysieren**

Aus den gesammelten Erfahrungen mit den Vorgängerprodukten leiten Sie Anforderungen für das neue Produkt ab.

### **Stakeholder identifizieren und analysieren**

Stakeholder sind Personen und Institutionen, die ein berechtigtes Interesse an dem neuen Produkt haben. Nicht nur der Produktmanager sollte sich intensiv über potentielle Stakeholder Gedanken machen, sondern auch die Anforderungsverantwortlichen nachfolgender Entwicklungsebenen.

### **Stakeholder auswählen und dokumentieren**

Wählen Sie aus den identifizierten Stakeholder-Gruppen einzelne Repräsentanten aus und dokumentieren Sie Ihre Auswahl.

### **Stakeholder-Interviews durchführen**

Vereinbaren Sie mit den ausgewählten Stakeholder-Repräsentanten Interviewtermine und befragen Sie sie zu den von Ihnen vorbereiteten Anforderungsthemen. Lernen und üben Sie dazu Interviewtechniken. Hierzu profitieren Sie von Kenntnissen über NLP (Neuro-Linguistisches Programmieren) [7].

### **Anforderungen aus Interview-Ergebnissen ableiten**

Leiten Sie aus den Interviewergebnissen konkrete Anforderungen ab.

### **Anforderungen mit Auftraggeber / Stakeholdern abstimmen**

Stimmen Sie die aus den Interviewergebnissen abgeleiteten Anforderungen in einer nächsten Iteration mit den Stakeholdern ab.

### **Anforderungen analysieren, modellieren**

Modellieren Sie im Rahmen der Analyse die Anforderungen. So werden Inkonsistenzen und Unvollständigkeiten entdeckt, und ein verbessertes Themenverständnis wird erlangt. Dabei entwickeln Sie neben der Kontextsicht Funktionale-Anforderungssichten, interaktive und generische Verhaltenssichten. Als Notation dazu sei nochmals auf die UML / SysML [5] verwiesen. Um noch weitere Anforderungen aufzudecken, führen Sie auf Basis der entstandenen Sichten eine Fehleranalyse durch.

### **Anforderungen simulieren, ausführen**

Mithilfe von Toolunterstützung und entsprechender Vorbereitung führen Sie das Anforderungsmodell aus und simulieren so die Anforderungen. Mittels der Simulationsergebnisse verbessern Sie die Anforderungsqualitäten.

### **Anforderungsmachbarkeit prüfen**

Ein Anforderungsqualitätsmerkmal ist die Machbarkeit der Anforderung. Wenn nicht einmal der Fachspezialist die Machbarkeit abschätzen kann, müssen Sie eine Machbarkeitsstudie (Feasibility Study) beauftragen. Damit entscheiden Sie die Machbarkeit.

## Prototyp (Wegwerf- oder evolutionär) erstellen

Mittels des vom Prototypen-Team erstellten Prototyps erhalten Sie eine konkrete Aussage über die Machbarkeit der Anforderung. Den Wegwerf-Prototyp entsorgen Sie, den evolutionären Prototyp lassen Sie ins Produkt einfließen. Vereinbaren Sie vor der Prototypenerstellung dessen Art und halten Sie das Ergebnis schriftlich fest. Beachten Sie, dass ein Wegwerf-Prototyp niemals zu einem evolutionären Prototyp mutieren darf!

## QA: Reviews durchführen

Führen Sie mit ausgewählten Stakeholdern Reviews der textuellen und der modellierten Anforderungen durch. Mit den Review-Ergebnissen verbessern Sie die Anforderungsqualitäten.

## QA: Funktionalszenarien und Qualitätsszenarien ausführen

**Funktionalszenarien:** Analysieren Sie mit „Papier und Bleistift“ oder toolgestützt mit der UML / SysML Notation [5] typische und untypische funktionale Abläufe gegen die Schnittstellen des Systems. Basis dazu ist der aktuelle Kenntnisstand über das System. Prüfen Sie dabei die Konsistenz und Vollständigkeit der Anforderungen zu den Abläufen. Als nebenläufigen Nutzen können Sie diese Funktionalszenarien auch als Testsznarien gegen die externen Schnittstellen (Systemtest) einsetzen.

**Qualitätsszenarien:** Analysieren Sie, welche realistischen und konkreten Szenarien sich für (abstrakte) Qualitätsmerkmale für das System ergeben. Beispielsweise resultieren aus dem abstrakten Qualitätsmerkmal Portabilität in der Embedded-Software mehrere Dimensionen: Portierbarkeit bezogen auf Hardware, Betriebssystem, Kommunikation, Middleware. Für alle relevanten Dimensionen arbeiten Sie realistische Szenarien aus, z.B. Hardware-Portierung von Mikrocontroller A zu Mikrocontroller B, und nehmen diese als Anforderung mit auf.

## Konsistenz zwischen Anforderungen und Umsetzung gewährleisten

Gleichen Sie kontinuierlich die Umsetzung mit den Anforderungen ab. Im Fehlerfall korrigieren Sie die Anforderungen. Bei Anforderungsänderungen prüfen Sie die Umsetzung. Allerspätestens am Projektende sollte das System exakt die Umsetzung der Anforderungen widerspiegeln – nicht mehr und nicht weniger.

## Abnahmekriterium / Test-Case für jede Anforderung entwickeln

Das Testteam stattet jede Anforderung mit mindestens einem Test-Case aus. Durch die personelle Trennung (vier-Augen-Prinzip) ergeben sich positive Rückwirkungen auf die Anforderungsqualitäten. Mit dem späteren Ausführen des Test-Cases weist der Tester nach, dass die Anforderung im System umgesetzt ist. Es existiert keine Anforderung ohne Test-Case. In Umkehrschluss entfällt jede Anforderung ohne Test-Case.

## Kontinuierliches, proaktives Risikomanagement etablieren

Die Projektleitung führt während des gesamten Projektes ein Risiko-Tracking-Dokument, in das die Projektbeteiligten die identifizierten Risiken eintragen.

Proaktives und kontinuierliches Projekt-Risikomanagement															
Risiko ID	Bezeichnung	Kategorie	Ursache	Eintritt-Identifikationsindikator	Auswirkung	Notfallmaßnahme bei Risikoeintritt	1=gering; 2=mittel; 3=hoch	1=gering; 2=mittel; 3=hoch	RPZ = P*S	Schwellwert für Maßnahmen RPZS	Maßnahmen ID	Maßnahme	P'	S'	RPZ
							Eintrittswahrscheinlichkeit P	Schadensmaß S	Risiko-prioritätszahl RPZ						
1							1	2	2	4	1.1		1	1	1
											1.2		1	1	1
											1.3		1	1	1
											2.1		1	1	1
2						1	2	2	4	2.2		1	1	1	
										2.x		1	1	1	

Bild 4: Risiko-Tracking Dokument

Damit Sie die Eintrittswahrscheinlichkeit P und das Schadensausmaß S nicht nach Bauchgefühl angeben, müssen Sie für P und S konkrete Werte und Einheiten definieren.

## Änderungsprozess für Anforderungen etablieren

Während der Entwicklung ändern sich aus verschiedensten Gründen immer wieder Anforderungen, bzw. neue kommen hinzu und bestehende entfallen. Hierzu sollten Sie einen Ablauf definieren, in dem ein Team die Konsequenzen der Änderung bewertet (Impact-Analyse) und final über die Änderung entscheidet. Fällt die Entscheidung gegen die Änderung, sollten Sie die Anforderung für das Folgeprodukt speichern.

## Aus Fehlern lernen

Sorgen Sie dafür, dass Sie die gleichen Fehler (in der Anforderungserhebung) nicht mehrfach machen → Lessons learned.

## Prozess, Vorgehen, Methode und Tooling kontinuierlich verbessern

Jedes neue Projekt bietet die Chance, den Prozess und was dazugehört kontinuierlich zu verbessern. Notieren Sie sich die Optimierungspunkte während des Projektes (Projekttagbuch) und adaptieren Sie den Prozess vor dem nächsten Projekt.

## Requirements Management

(Anforderungen verwalten, verfolgen und verlinken)

### Anforderungen wiederverwenden

Insbesondere in Unternehmen, in denen Sie immer gleiche oder ähnliche Produkte entwickeln, bietet sich die Wiederverwendung von Anforderungen an. Dabei ist eine Möglichkeit der Minimal-Plattformgedanke: die für alle Produkte gleichen Anforderungen als Plattformanforderungen zu separieren und im individuellen Produkthanforderungsset zu referenzieren. Eine Alternative dazu ist der Maximal-Plattformgedanke. In der Maximalplattform sind alle Anforderungen für alle Produkte enthalten. Für neue Produkte referenzieren Sie die entsprechend gültigen und versionierten Anforderungen.

### Anforderungen parametrieren

Auch im Kontext der Anforderungswiederverwendung können Sie Anforderungstext mit Variablen und Einheiten besetzen, deren Werte Sie produktindividuell in deren Parameterlisten anpassen, z.B. die maximale Leistungsaufnahme  $P_{max}$  im Watt.

### Anforderungseigenschaften (Properties / Attribute) definieren

Statten Sie Anforderungen mit Eigenschaften aus, von denen Sie einen Nutzen haben. Manchmal ist weniger mehr. Neben der einzigartigen ID (z.B. REQ-EMB-SW\_000001) zur Identifikation und Referenzierung, sollte jede Anforderung auch einen Status (z.B. in Bearbeitung / freigegeben / umgesetzt / getestet) besitzen. Alle Eigenschaften müssen Sie pflegen, sonst haben diese keinen Nutzen.

Attribut	Erläuterung	Nutzen
ID	Eindeutige Identifikationsnummer	Identifikation, Referenzierung
Version	Version der einzelnen Anforderung	Verfolgbarkeit, Änderbarkeit
Stakeholder / Quelle	Anforderungsquelle	Nachfrage
Autor	Autor der Anforderung	Nachfrage
Verantwortlicher Autor	Verantwortlicher	Im Streitfall
Begründung / Rational	Begründung für die Anforderung	Notwendigkeit
Auftraggeber-Zufriedenheit	Bei Nichterfüllung der Anforderung	Umsetzungsreihenfolge
Auftraggeber-Unzufriedenheit	Bei Nichterfüllung der Anforderung	Umsetzungsreihenfolge
Art	Funktional   Nichtfunktional   ...	Bewusstsein wecken
Kategorie	Produkt   Projekt   Produktgruppe   Unternehmen	Wiederverwendbare Anforderungen identifizieren
Entwicklungsdomäne Detaillierungsstufe	Auftraggeber   System   Subsystem   HW   SW   MECH	Zuordnung zur Umsetzung
Priorität	Priorisierung nach definiertem Schema	Umsetzungsreihenfolge
Risiko	Risikoeinstufung nach definiertem Schema	Vorbeugen, Testtiefe
(Rechtliche) Verbindlichkeit	Verbindlichkeit zur Umsetzung gegenüber Auftraggeber	Umsetzungsreihenfolge

Bild 5a: Beispiele möglicher Anforderungseigenschaften

Attribut	Erläuterung	Nutzen
Geplant für Release	1. Umsetzung ab Release x	Projektplanung
Komplexität	Der Anforderung und damit auch der Umsetzung	Projektplanung
Enthalten im Use-Case ID(s)	Verbindung zum Use-Case	Verfolgbarkeit
Umgesetzt im Design ID(s)	Verbindung zum Design	Verfolgbarkeit
Verifiziert mit Test-Case ID(s)	Verbindung zum Test	Verfolgbarkeit
Status	Von der Erhebung bis zum Produkttest	Projektüberwachung
...	...	...
Abhängigkeiten	Zu anderen Elementen	Änderungen
Offene Fragen	Nur temporäre Einträge	Nichts vergessen
Konflikte		Vorbeugen
Umsetzungskosten	Kosten für die Umsetzung	Projektkalkulation,
Umsetzungszeit	Benötigte Zeit für die Umsetzung	Angebotsabgabe
Referenzen	Zu anderen Elementen	Änderungen
Historie	Einer Anforderung	Verfolgbarkeit, Nachvollziehbarkeit
Kommentar	Versuchen Sie besser, die Anforderung so zu schreiben, dass kein Kommentar erforderlich ist	Kein Nutzen!

Bild 5b: Beispiele möglicher Anforderungseigenschaften

## Versionierung von Anforderungen etablieren

Die Version als Eigenschaft der Anforderung erlaubt es Ihnen, die Entstehung nachzuverfolgen und Anforderungen zukünftig evolutionär weiterzuentwickeln.

## Baselining / Releases für Anforderungen etablieren

Zu einem definierten Zeitpunkt im Projektablauf halten Sie die aktuelle Reife der Anforderungen zur weiteren Umsetzung fest.

## Bidirektionale Traceability über alle Entwicklungsschritte etablieren

In denen für sicherheitskritische (Safety) Systeme gültigen Standards ist die bidirektionale Traceability (Verlinkung) zwischen Anforderungen über die Umsetzung bis hin zum Test-Case meist gefordert. Dies über verschiedene Entwicklungsdomänen und deren unterschiedliche Tools zu schaffen ist dabei die wesentliche Herausforderung → Toolintegration, Standards wie ReqIF [5] und STEP [7].

## Templates für Artefakte bereitstellen

Stelle Sie für die Artefakte im Prozess Templates bereit. Damit erreichen Sie, dass die gleichen Artefakt-Typen (z. B. Requirements Specification) immer gleich aussehen und erhöhen so das Verständnis. Durch die vorgegebenen Inhalte übersehen Sie keine Punkte, die zu bearbeiten sind. Für viele Artefakt-Typen gibt es heute bereits Standards.

## Requirements Management Tools

### Text vers. Application/Product Life Cycle (A/PLM) Tools evaluieren

Grundsätzlich müssen Sie bei Tools die Entscheidung zwischen rein textverarbeitenden oder datenbankorientierten Tools treffen. Die Komplexität Ihrer Produkte wird weiterhin steigen. Je mehr Anforderungen und andere Artefakte Sie managen müssen, desto besser unterstützen Sie dabei datenbankorientierte Tools.

### Tool anforderungsbasierend auswählen

Es gibt nicht das beste Tool am Markt. Es gibt höchstens das Tool am Markt, das Ihren Bedarf bzw. Ihre Toolanforderungen am besten erfüllt. Denken Sie bei der Toolauswahl an die folgende Reihenfolge: erst der Mensch, dann die Methode und dann das Tool! Die Toolauswahl sollte keine firmenpolitische Entscheidung sein.

## Tool integrieren, testen und schulen

Integrieren und testen Sie das ausgewählte Tool vor Projektbeginn. Um einer möglichen Toolablehnung durch die Anwender entgegenzuwirken, sollten diese vor der Toolnutzung eine Toolschulung erhalten.

## Tool in täglicher Arbeit etablieren

Etablieren Sie das Tool über mehrere Abteilungen hinweg. Auch hier ist ein kontinuierlicher Lern- und Verbesserungsprozess zu leben.

## Resümee

Bei jedem einzelnen Checklisten-Punkt müssen Sie für sich den dazugehörigen Nutzen im Verhältnis zum Umsetzungsaufwand abwägen. Die besten Punkte für Sie sind die mit hohem Nutzen bei kleinem Umsetzungsaufwand.

## Referenzierte und weiterführende Links:

[1] MicroConsult Download für diesen Beitrag: <http://download.microconsult.net/ese2017/req-checkliste.zip>

[2] **MicroConsult Trainings auf [www.microconsult.de](http://www.microconsult.de):**

**Requirements Engineering und Management für Embedded-Systeme**

**Funktionale Sicherheit (Safety) von Elektronik und deren Software**

**Security: Sicherheit von Embedded-Systemen und ihr Kontext zur funktionalen Sicherheit**

**Usability: Produkte benutzerfreundlich entwickeln**

**SysML: Systemanalyse und Systemdesign mit der Systems Modeling Language**

**UML-Praxis-Workshop: Praktischer Einsatz für Embedded- und Echtzeit-Softwareentwicklung**

[3] V-Modell: [www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell\\_xt\\_node.html](http://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html)

[4] Agiles Manifest: <http://agilemanifesto.org/iso/de/manifesto.html>

[5] Object Management Group (OMG): <http://www.omg.org>

[6] STEP (Standard for the Exchange of Product Model Data) ISO 10303: <http://www.iso.org>

[7] Neuro-Linguistisches Programmieren: [https://de.wikipedia.org/wiki/Neuro-Linguistisches\\_Programmieren](https://de.wikipedia.org/wiki/Neuro-Linguistisches_Programmieren)

[8] **Fachliteratur:** Requirements Engineering und Management, SOPHIST GmbH  
[HANSER Verlag](http://www.hanser-verlag.de), Print-ISBN: 978-3-446-43893-4, E-Book-ISBN: 978-3-446-44313-6

## Autor:



Dipl.-Ing. (FH) Thomas Batt ist gebürtiger Freiburger. Nach seiner Ausbildung als Radio- und Fernstehtechner studierte er Nachrichtentechnik in Offenburg. Seit 1994 arbeitet er kontinuierlich in verschiedenen Branchen und Rollen im Bereich Embedded-/Real-Time Systementwicklung. 1999 wechselte Thomas Batt zur MicroConsult GmbH. Dort verantwortet er heute als zertifizierter Trainer und Coach die Themenbereiche Systems /Software Engineering für Embedded-/Real-Time-Systeme sowie Entwicklungsprozess-Beratung.

[t.batt@microconsult.de](mailto:t.batt@microconsult.de)