

## Qualitätssicherung in der Software-Entwicklung am Beispiel des BMW i3

Peter Schiele<sup>1</sup>, Alexander Siller<sup>2</sup>

**Abstract:** Die E-Mobilität stellt hohe Ansprüche an ihre zugrunde liegende Entwicklung. Schnellere und kosteneffizientere Abläufe sind unumgänglich, um mit der Entwicklung im Wettbewerb Schritt zu halten. Das gilt nicht nur für die HW-Entwicklung des Energiespeichers oder der E-Maschinen, sondern auch für die eingebettete Software des E-Antriebsystems.

Die Entwicklung des BMW i3 wartete mit etlichen neuen und zu Beginn teils unbekanntenen Herausforderungen auf. Reorganisation im Großen wie auch im Kleinen, Optimierung der Infrastruktur und eine kontinuierliche Prozess- und Produkt-Optimierung – sie alle bildeten die Voraussetzung, um die erwarteten Qualitätsansprüche in einem angespannten Termin- und Kostenrahmen zu erfüllen. Die notwendigen Veränderungen waren der Schlüssel zum Erfolg. Veränderungen beinhalten aber auch das Risiko, Projektziele hinsichtlich Software-Qualität nicht zu erreichen.

Dieser Artikel beschreibt, wie in der Praxis Veränderungsbedarfe zielgerichtet anhand von Metriken ermittelt und gesteuert wurden, aber auch die Metriken selbst Änderungen unterlagen.

**Keywords:** Metriken, Kontinuierlicher Verbesserungsprozess, Qualitätssicherung, Projektsteuerung, E-Antriebe, Zielvereinbarung

### 1 Einleitung

Die SW-Qualitätssicherung beim E-Antrieb des i3 setzte auf unterschiedliche Industrie-Standards auf. Diese beschreiben, wie Prozess- und Qualitätsverbesserungen strukturiert umgesetzt werden:

Die ISO/TS 16949 [Is09] gibt Anforderungen vor, deren Erfüllung kontinuierliche Verbesserungen verspricht. Six Sigma [Lz07] liefert einen Werkzeugkasten an Methoden. CMMI [CK11] bietet Best Practices um stufenweise zu reiferen Prozessen zu gelangen. Alle haben eines gemeinsam: Messen und Analyse zählen zu den Kerninhalten, um Projekte zu reifen Prozessen zu führen. Allerdings sagte keine Norm etwas darüber aus, welche Messgrößen sinnvoll angewendet werden können oder sollten.

---

<sup>1</sup> BMW AG, EA-41, Taunusstr. 41, 80788 München, peter.schiele@bmw.de

<sup>2</sup> BMW AG, EA-41, Taunusstr. 41, 80788 München, alexander.siller@bmw.de

Die Auswahl der Metriken muss sich gemäß Goal/Question/Metric-Methode [SoBe99] an der Frage der Zielsetzung und an den Anforderungen des Entwicklungsprozesses orientieren. Dieser Aspekt wird am Beispiel des BMW i3 im folgenden Abschnitt erläutert.

Der Abschnitt danach geht auf die Definition der Metriken, die Infrastruktur für eine zuverlässige, effiziente und flexible Datenerfassung sowie deren Aufbereitung ein.

Im Anschluss werden die Analyse, die Umsetzung von Maßnahmen und wirksame Steuerungsmechanismen im Projekt i3 dargestellt.

Abschließend wird der Evolutionsverlauf und eine mögliche Standortbestimmung eines beliebigen Messsystems betrachtet.

## 2 Metriken leiten sich von Zielen ab

Orientierungsrahmen für die Metriken sind die Projektziele. Die Messgrößen bilden den Ausgangspunkt und den aktuellen Start von Verbesserungsprozessen ab. Gleichzeitig stellt sich aber auch die Frage nach der Wirksamkeit von bereits ergriffenen Maßnahmen.

Standardmäßig werden Ziele wie hundertprozentige Testabdeckung, Null Fehler, Reduzierung der Entwicklungskosten o. ä. genannt. Am besten noch alle drei in Kombination.

Auch wenn es den Entwicklerteams oft als nicht realistisch erschien, solche Zielvorgaben unter gegebenen Rahmenbedingungen zu erreichen, so war deren Forderung doch legitim. Wichtig war es bei solchen Metriken, einen zeitlichen Verlauf wie „Burndown“ o.ä. als Zielkurve zu vereinbaren, die realistisch erreichbar war, damit der erste Widerstand gebrochen wurde.

### 2.1 Ziele aus Projektanforderungen

**Testabdeckung:** Die Software für den elektrischen Antrieb beschleunigt und bremst das Fahrzeug über die E-Maschine und ist damit Teil der sicherheitsgerichteten Entwicklung. Aus der Anforderung an die funktionale Sicherheit folgte unmittelbar, dass die notwendige Testabdeckung auf Anforderungs-, Design- und Modul-Ebene für die am Sicherheitskonzept beteiligten Software-Einheiten erreicht werden musste. Jede Testlücke in der Software stellte ein Risiko dar. Jeder Fehler musste vermieden werden.

**Änderungsaufkommen und Produkt-Stabilität:** Der Innovationsgrad in diesem Projekt lieferte einen kontinuierlich wachsenden Ideenpool für Verbesserungen und Änderungsbedarf an den Konzepten. Diese Änderungen brachten ein Risiko mit sich, welches – je näher der SOP rückte – vermieden werden musste. Da die einzelnen Software-Teams im BMW i3 Projekt nach agilen Methoden und daher weitgehend

eigenverantwortlich arbeiteten, waren Metriken zu Änderungsaufkommen und zur Produkt-Stabilität notwendig, um den Scrum-Teams ein Steuerungsmittel an die Hand zu geben.

In Abb. 1 ist der prozentuelle Anteil an geänderten Codezeilen im Vergleich zu den letzten Releases für einzelne Software-Komponenten dargestellt. Die rechte Graphik verdeutlicht dies anhand der Farbgebung von Grün, über Gelb nach Rot. Je mehr die Farbe ins Gelb und dann ins Rot übergeht, desto mehr Änderungen sind eingeflossen. Die Änderungsrate gibt zugleich einen Hinweis, wo die Testschwerpunkte liegen sollten.

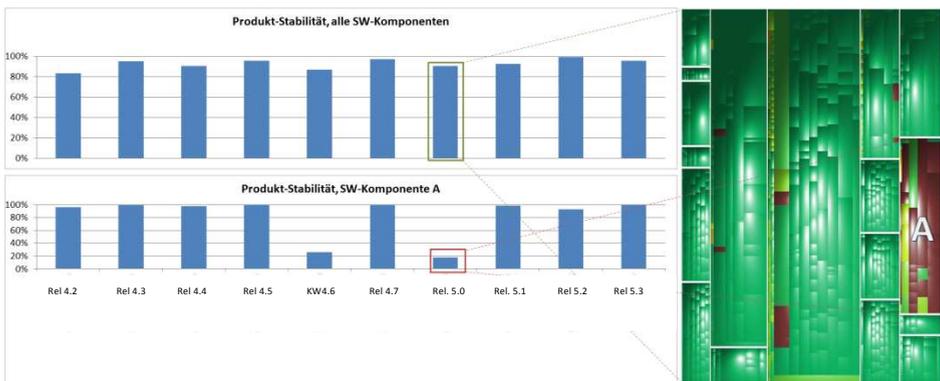


Abb. 1: Beispiel Metrik: Produkt-Stabilität

## 2.2 Ziele aus Prozessanforderungen

**Termintreue:** Mittels Meilensteintrendanalyse wurden Planungsrisiken, kritische Pfade, sowie Leistbarkeitsthemen identifiziert, indem der Änderungszeitpunkt des Meilensteins mit dem jeweiligen Zieltermin verglichen wurde. Dies verhalf zu einer realistischeren Terminplanung und entsprechender Termintreue.

**Schnittstellenkonsistenz:** Anhand von automatischen Prüfungen konnte die Prozessdurchgängigkeit und die Konsistenz zwischen den Prozessschritten (z.B. AUTOSAR-XML) sichergestellt werden. Die Konsistenz war damit frühzeitig hergestellt und ersparte aufwändige Korrekturschleifen bei der Software-Integration. Dabei wurde die Anzahl der Abweichungen zwischen Simulink<sup>®</sup>-Modell und AUTOSAR-XML gemessen.

**Richtlinienkonformität:** Umfangreiche Richtlinienchecks auf Funktions-Modell- und auf Code-Ebene verringerten das Fehlerrisiko in der Software, insbesondere das Risiko jener Fehler, die sonst nur durch hohen Testaufwand gefunden werden konnten. Zentrale Messgröße waren die Anzahl der Fehlermeldungen von QA-C.

**In- Outflow Problemtickets:** Fehler-In- / Outflow dienten als Prognose-Methode für

zukünftige Fehlerraten und Abbauraten. Dies führte zu dem Ziel, die geforderte Reife zu den entsprechenden Fahrzeug-Terminen zu erreichen.

### 2.3 Metriken aus internen Zielen

**Durchlaufzeiten:** Durch Messung der Geschwindigkeiten bei der Umsetzung und Integration von Änderungen und Fehlerbehebung wurden Langläuferthemen identifiziert und Fehlerabbauraten erhöht.

**Wartbarkeit, Änderbarkeit:** Komplexitätsmaße gaben ein Indiz für die Wartbarkeit und Änderbarkeit der Software. Neben geläufigen Metriken wie McCabe-Index auf Code-Ebene wurden auch Clone-Checker verwendet, um Redundanzen in der Modell- und Codeentwicklung zu identifizieren.

**Effizienz:** Die Wiederverwendung von Software für unterschiedliche Fahrzeugvarianten reduzierte den Aufwand in der Entwicklung und in den nachgelagerten Testaktivitäten. Die Wiederverwendung von sogenannten Baukastenelementen wurde durch Codevergleich gemessen. Durch die Einsparung von Codevarianten konnten die Entwicklungsaufwände über die Fahrzeugprojekte hinweg und damit auch im i3 deutlich reduziert werden.

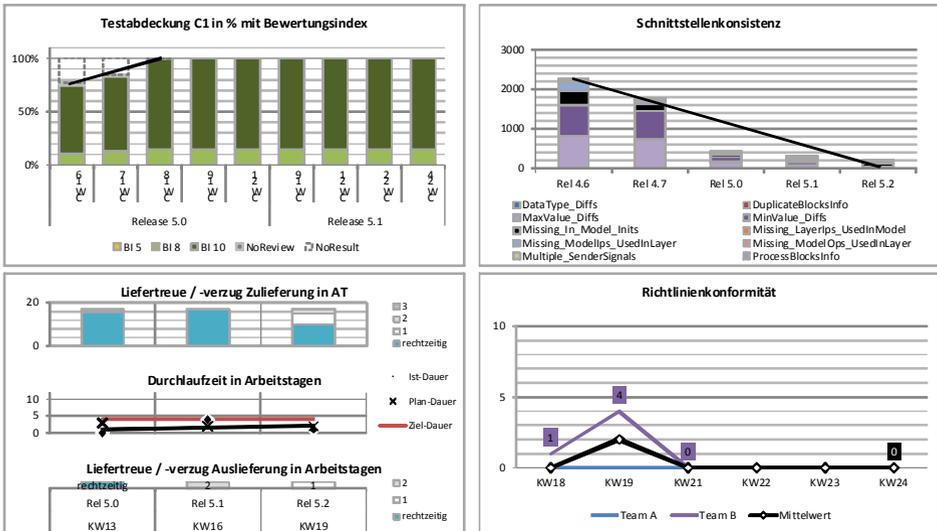


Abb. 2: Metriken für Testabdeckung, Schnittstellenkonsistenz, Durchlaufzeit und Richtlinienkonformität. Die Darstellung der Metriken ist auf Liniendiagramme und gestapelten Balken begrenzt. Auf- und Abbaukurven sowie Ziele geben den Sollverlauf vor.

## **2.4 Ziele und Metriken als Führungsinstrument**

Die Vereinbarung der Ziele erfolgte in den Fachbereichen typischerweise über verschiedene Management-Ebenen, innerhalb der Entwicklungsprojekte in Abstimmung mit den Projekt-Hierarchieebenen oder im Rahmen von aufgesetzten Prozessprojekten.

Als Vorbereitung zur Zielvereinbarung hat es sich bewährt, erste Messungen durchzuführen, um die Erreichbarkeit der Ziele zu gewährleisten. Dazu bedurfte es einer weiteren Verfeinerung in Form der Metrikdefinition, die das Ziel für die Messmethode beschreibt.

Auch im Rahmen des i3-Projektes hat sich gezeigt, dass sich nach einiger Zeit die erwünschten Verbesserungen und Veränderungen im Team-Verhalten nach Identifikation eines Problembereichs und Vereinbarung von entsprechenden Metriken ergeben haben.

## **3 Wer seine Ergebnisse nicht misst ...**

.. der wird seine Ziele nie erreichen. Die Grundlage der Messung bilden die Basisdaten, die Berechnungsformel, der Umfang der Messung und die Bewertung in Form eines Ampelstatus. Die daraus resultierenden Metriken stellen komplexe Sachverhalte einfach und überschaubar dar.

### **3.1 Vertrauen durch Objektivität**

Metriken definieren die Ziele sehr konkret. Weil sie Transparenz herstellen und evtl. Abweichungen vom Ziel quantitativ aufzeigen, sind sie teilweise konkreter als vom Zielenehmer gewünscht.

Damit der Zielenehmer nicht in den Konflikt zwischen objektiver Messung und Zielerreichung kommt, wurde die Aufgabe Metriken zu definieren im i3-Projekt einem unabhängigen Software-Qualitätsteam übertragen. Die Aufgabe beinhaltete auch die Abstimmung mit den beteiligten Fachstellen, das Durchführen der Messung sowie die Darstellung der Metrik. Das erhöhte das Vertrauen in die Messungen und verringerte zusätzliche Belastung der Entwicklungsteams. Durch die objektiven Messungen konnten subjektive Wahrnehmungen korrigiert werden und weiterer Maßnahmenbedarf aufgezeigt werden.

### **3.2 Nachschärfen von Zielen und Prozessen**

Die Metrikdefinitionen lieferten Angaben über die genaue Definition, Identifikation und vor allem die Verantwortung für Gesamt- oder Teilumfänge des ausgewählten Messumfanges, sowie die notwendige Messfrequenz. Sie wurden mit dem Zielenehmer

abgestimmt.

Die abgestimmten Definitionen trugen dazu bei, die Steuerung zu beschleunigen, da die nachher durchgeführten Messungen als Statusbericht direkt und ohne weitere Abstimmung mit den Steuerungsverantwortlichen an höhere Berichtsebenen weitergereicht werden konnte.

Der Status präventiver Qualitätssicherung, insbesondere die Prozesskonformität erlangte dadurch hohe Transparenz und Aktualität, was schnelleres Gegensteuern bei Prozessabweichungen ermöglichte.

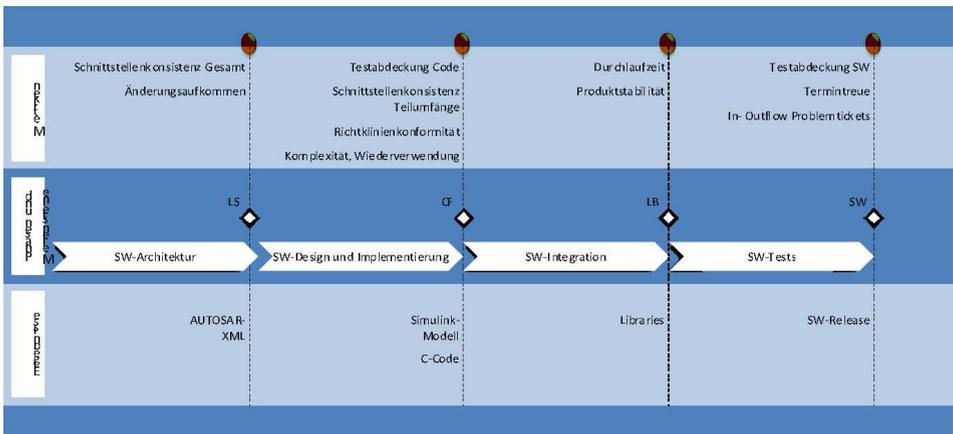


Abb. 3: Metriken werden einzelnen Phasen und Meilensteinen und den zu messenden Ergebnissen zugewiesen.

Die agil arbeitenden Teams blieben im Gesamtentwicklungsprozess eingebunden, weil Qualität und Termintreue gemessen und eingefordert wurden.

### 3.3 Erste Messung

Die erste Messdatenerhebung bedurf eines tieferen Einblicks in die verwendete IT-Infrastruktur und in die Struktur der abgelegten Daten. Zu den theoretischen Überlegungen erfolgte eine weitere Vertiefung durch die Betrachtung von Sonderfällen. Was gehört zur 100% Zielerreichung und was wird nicht mit gemessen?

Beispielsweise mussten nicht-freigegebene Anforderungen auch nicht getestet werden. Release-Kandidaten oder Entwickler-Software-Stände mussten von Software-Releases unterschieden werden. Verletzungen von Programmierrichtlinien durften nicht als solche gewertet werden, wenn sie der vereinbarten „Deviation Procedure“ unterzogen und in „Whitelists“ dokumentiert wurden.

Durch das richtige Filtern der Daten und die Auswahl der relevanten Informationen stieg die Signifikanz der Messung und die Akzeptanz der Ergebnisse.

Für die Darstellung der Metrik gilt die einfache Regel: „KISS“ bzw. „Keep it SMART and Simple“. Aus der Darstellung muss intuitiv der aktuelle Status, falls bereits mehrere Messpunkte vorhanden, der aktuelle Trend und vor allem Indizien für den Handlungsbedarf und dessen Verantwortung klar hervorgehen. Das ist in aller Regel ausreichend. Die Rückverfolgbarkeit der Messung ist nach dem gleichen Prinzip über die Metrikdefinition gegeben.

Mit diesem Ansatz konnten die eigenverantwortlichen Teams im i3-Projekt damit nicht nur ihre eigenen Ergebnisse einsehen, sondern auch die anderer Teams. Das förderte den internen Wettbewerb und auch den Austausch. Wo ein Team noch Aufholbedarf hatte, konnte es vom dem anderen Team lernen, das schon am Ziel angekommen war.

Für die Teams war nun klarer verständlich, was die Zielerwartung ist, wo sie aktuell stehen und wo noch Handlungsbedarf existiert. Die Metrik lieferte dafür ein geeignetes Medium zur Kommunikation.

### **3.4 Metriken automatisieren**

Sofern nicht Projektmeilensteine oder Baselines andere Taktraten vorgaben, wurde mindestens wöchentlich gemessen. Die hohe Messfrequenz erforderte eine Automatisierung, zum einen um den Aufwand gering zu halten und zum anderen um die Qualität der Messungen zu gewährleisten und keine Fehler durch manuelle Schritte zu machen. Deshalb bedurfte es für die Erstellung der Metriken einer eigenen Infrastruktur, die in Abb. 4 dargestellt ist.

- Eine automatische Importfunktion sammelte sämtliche Basisdaten aus unterschiedlichen Systemen und Formaten automatisiert ein und legte diese strukturiert, in Form einer SQL-Datenbank, ab. Der Abgleich in die Datenbank erfolgte zentral und skriptgesteuert nach einem fest definierten Messplan, welcher sich aus der Updatefrequenz der zu bedienenden Metriken ableiten ließ.
- Die Datenbankstruktur ließ Zusammenhänge der Daten einfacher erkennen und bot gezielte Abfragemöglichkeiten anhand von SQL-Queries.
- Die Darstellung der Metriken ergab sich durch initial erstellte Diagramm-Formate und dem Update der Datenbankabfrage. Metriken konnten zentral vom Software-Qualitätsteam oder von den einzelnen Steuerungsverantwortlichen aktualisiert werden. In beiden Fällen übernahm das Qualitätsteam die Verantwortung für die Korrektheit der Messung, Datenerfassung und Aufbereitung.

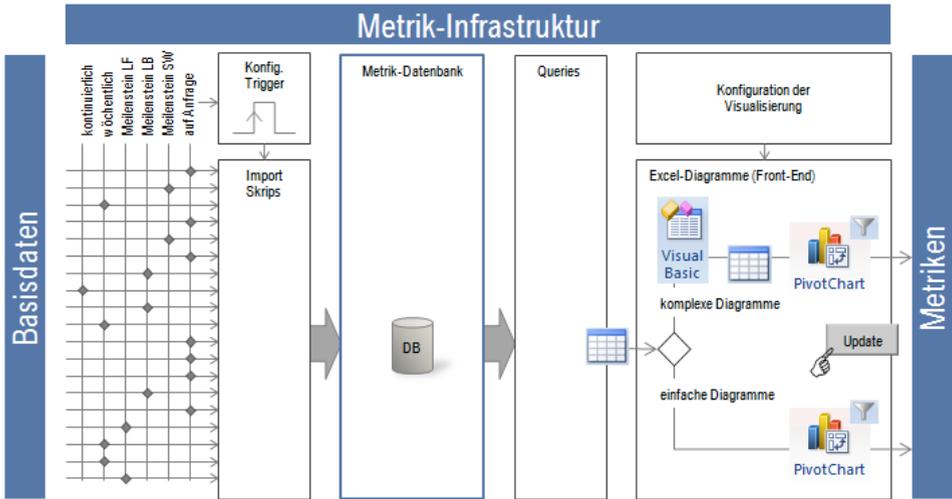


Abb. 4: Prinzipieller Aufbau der Metrik-Infrastruktur: Links der automatische Import der Basisdaten in die Datenbank, rechts die Datenabfrage mit Aggregation und Filterung der Daten in Diagrammen.

Verbesserung der Objektivität wurde durch die Automatisierung erreicht, da sie manuelle Eingriffe unterband.

Die Infrastruktur wurde für das i3-Projekt entwickelt und bediente sich gängiger Technologien für Skripte zur Automatisierung, wie Perl, Datenbanken, wie SQL und Diagrammerstellung mit Excel, welche den Anwendungsfällen gerecht wurden. Die schnelle und flexible Erweiterbarkeit zählte neben dem Verwendungszweck selbst zu einem der wichtigsten Anwendungsfälle und sorgte dafür, dass die Infrastruktur für aktuelle Projekte unverzichtbar geworden ist.

#### 4 Analyse und Steuerung

Eine gut gelungene Metrikdefinition vereinfacht die Analyse, ersetzt sie aber nicht vollständig. Defizite, welche zur Zielabweichung führen, müssen nicht unmittelbar beim Zielnehmer liegen, sondern stammen oft aus vorangegangenen Prozessschritten.

Dies hat sich auch im i3-Projekt am Beispiel Software-Integration gezeigt. Hier wurde die längere Integrationsdauer durch inkonsistente Software-Schnittstellen verursacht. Spätes Bugfixing und Nachliefern von Release-Ständen waren die Folge von vorhandenen Testlücken in den vorangegangenen Prozessschritten.

## 4.1 Probleme erkennen

Zur Identifikation des Handlungsbedarfes in einer kausalen Kette bewährte sich die Aufteilung von Information auf mehrere Metrik-Diagramme, z.B. Software-Liefertreue und Anzahl der Nachlieferungen. Aus der Analyse ergaben sich weitere zugrunde liegende Ziele, entsprechende Abweichungen und evtl. neue Verantwortliche. Den Verantwortlichen könnten die Metriken individuell zugewiesen werden und versetzten sie in die Lage, selbst zu steuern.

Das bot gleich zwei Vorteile:

- Komplexere Probleme wurden in einfachere zerlegt und
- Fehler wurden da behoben, wo sie entstanden.

Das Messsystem wurde somit sukzessive verfeinert und führte zu einem stetigen Zuwachs an Metriken. Damit wurden Defizite unmittelbar dort aufgezeigt, wo die Ursache für Risiken verborgen lag und sie konnten schnell behoben werden. Ziele zur Fehlervermeidung wurden hinzugefügt.

## 4.2 Metriken strukturieren

Der Zuwachs an Metriken warf neben dem wachsenden Aufwand bei der Metrikerstellung auch das Problem des wachsenden Steuerungsaufwandes inkl. Ursachenanalyse, Erarbeitung von Maßnahmenvorschlägen und deren Abstimmung auf.

Die Automatisierung löste das Aufwandsproblem bei der Erstellung. Der Aufwand für die Steuerung und aller damit verbundenen Abstimmungen blieb noch bestehen.

Sämtliche Metriken wurden deshalb zu steuerungsverantwortlichen Rollen zugewiesen und erhielten dadurch die notwendige Verankerung im Prozess. Das sorgte zusätzlich für mehr Nachhaltigkeit bei der Zielerreichung.

Ein Software-Architekt war nun selbst für die Steuerung mittels der ihm zugeteilten SW-Architekturmetriken verantwortlich. Gleiches galt für die Rolle des SW-Integrationsleiters, des Problemmanagers, des Testmanagers und aller anderen zugewiesenen Rollen.

Im gleichen Zuge wurden die Berichtswege und die Berichtsformate festgelegt, über welche die Metriken weitergeleitet und die Maßnahmen abgestimmt wurden.

Damit hat das i3 Projekt eine Infrastruktur für ein Regelwerk geschaffen, mit dem Veränderungen in Projekten, Prozessen und sogar Organisationen systematisch herbeigeführt werden können.

Steuerungs- verantwortlicher	Metriken	Ziele	Berichtsgremien			
			CCB-TPK	SW-BKT	SW-Pj.-Runde	Teststrat.-Runde
CCB-Leiter	JIRA-Tickets, Anzahl je I-Stufe	Änderungstickets müssen ab bestimmter I-Stufe auf bestimmte %-Anteile reduziert werden.			●	
	Termtreue JIRA-Tickets	Einhaltung Prozessvorgabe: Bestimmter Status muss zu bestimmten Meilenstein erreicht sein.				●
ECU-Architekt	CPU-Load	Vereinbarte Maximalgrenzen werden nicht überschritten.	●			
	Konformität GS95015	Vollständige Erfüllung der anwendbaren Anforderungen aus RAM/ROM (Memory)	●			
	RAM/ROM (Memory)	Vereinbarte Maximalgrenzen werden nicht überschritten.	●			
FDT-Leiter	Abdeckung Codeinspektion	Code-Inspektion wurde für alle Sicherheitsmodule durchgeführt		●		
	Abdeckung Modultests	100% Modell- und Codeabdeckung (Decision-Cov., Statement-Cov., MC-DC, C1)		●		●
	Compiler-Warnungen	0 Warnungen		●		
	Ergebnisse Codeinspektion	BI > 6		●		
	Ergebnisse Modultests	BI > 6		●		●
	MISRA	0 Verletzungen		●		
	Modellrichtlinienverletzungen	0 Verletzungen		●		
	Modellschnittstellenprüfung	0 Inkonsistenzen		●		
	Testabdeckung Anforderungen	100% Anforderungsabdeckung		●		
	Problemmanager	QC-Fehlerbericht	Anzahl QC- Tickets mit offener TIS <= 7 (Inhouse) TIS muss innerhalb von 10 Tagen festgelegt werden.			
SW-Integ.-Leiter	Anzahl PST-Schienen	100% je Entwicklungs-Generation			●	
	Baseline-Review	Alle Artefakte sind je Baseline vollständig und aktuell			●	
	Durchlaufzeit SW-Build	Reduzierung um 25% in 2015			●	
	Liefertreue-Artefakte	Alle vereinbarten Artefakte werden vom vollständig und rechtzeitig ausgeliefert			●	●
	Maximale Dauer ink. Build	<= 20min			●	
SW-Integ.-Test	Meilensteintrendanalyse	Meilensteine sind mindestens 2 Wochen vor Eintreten stabil			●	
	PST-Nachlieferungen	keine PST-Nachlieferungen (= PST-Lieferungen außerhalb			●	
	Abdeckung Layertests	100% der spezifizierten Signale inkl. Negativtests	●			●
SW-Sicherh.-Verantw.	Ergebnisse Layertests	BI > 6	●			●
	Testabdeckung Änderungen	100% Änderungsabdeckung		●		
	Termtreue SW-Freigabeempfehlung	Vereinbarter Termin für SW-Freigabeempfehlung wird keine Abweichung vom Plan			●	
	SW-Sicherheitsaudit			●		

Abb. 5: Auszug aus verwendeten Zielen und Metriken, sowie deren Zuordnung zu steuerungverantwortlichen Rollen im Prozess und deren Berichtsgremien

### 4.3 Berichtsgremien und Steuerung

Veränderungsprogramme sind nur dann erfolgreich, wenn sie vom Management getragen werden. Das gilt auch in dem hier beschriebenen Vorgehen.

In Fall des i3-Projektes wurde das Vorantreiben des Änderungsprozesses von der Projektorganisation unterstützt. Das Metriksystem wurde zum Führungsinstrument. Trends zur Ansteuerung von Zielen und die erreichten Verbesserung haben nicht nur Transparenz und Vertrauen geschaffen, sondern wesentlich zum Erfolg des i3-Projektes beigetragen.

Das Metriksystem ist mittlerweile über das i3-Projekt hinaus in der Entwicklung der E-Antriebe bei BMW etabliert und in der Prozesslandkarte sowie über das Berichtswesen verankert. Unterschiedliche Gremien schätzen die Transparenz und verwenden die Metriken im Berichtswesen sowie zur Projektsteuerung.

## 5 Evolutionstufen

Der folgende Abschnitt fasst die oben dargestellten Schritte zusammen und gibt die Möglichkeit, den aktuellen Stand bzgl. Steuerbarkeit über Metriken für ein Projekt oder eine Organisation zu bestimmen.

Schritt 1: Einzelne Entwicklungsteams stellen ihre Ergebnisse in Form von Zahlen, Daten, Fakten dar und steuern sich damit selbst. Der Einfluss auf andere Teams ist gering.

Schritt 2: Zielvorgaben von Management werden mit den Zielen und den Metriken der Entwicklungsteams in abgeglichen und abgestimmt. Jedes Team berichtet eigenständig.

Schritt 3: Metriken werden zu Prozessen zugeordnet und zentral verwaltet. Die Metrikerstellung wird automatisiert.

Schritt 4: Bei der Analyse entsteht der Bedarf weiterer Metriken in vorangelernten Prozessschritten und neuer Zielvorgaben.

Schritt 5: Die Metriken werden über Rollenzuordnung und das Berichtswesen verankert und nachgehalten.

## Literaturverzeichnis

- [Is09] ISO/TS16949 Qualitätsmanagementsysteme, Besondere Anforderungen bei Anwendung von ISO 9001:2008 für Serien- und Ersatzteil-Produktion in der Automobilindustrie, Dritte Ausgabe, Genf 2009
- [CK11] Chrissis, M.B., Konrad, M., CMMI 1.3 für die Entwicklung: Richtlinien für Prozessintegration und Produktverbesserung, Boston 2011
- [Lz07] Lunau, S. et.al.: Design for Six Sigma+Lean Toolset: Innovationen erfolgreich realisieren, Frankfurt 2007
- [SoBe99] Rini van Solingen, Egon Berghout: Goal/Question/Metric Method, 1999