

Anforderungen an eine Software für das Energiedatenmanagement – Requirements Engineering und Anforderungsintegration am Beispiel der Software EMSControl

Masterarbeit

Eingereicht von: **Ing. David Gernot Müller, BA**

Matrikelnummer: 01512327

im Fachhochschul-Masterstudiengang Wirtschaftsinformatik
der Ferdinand Porsche FernFH GmbH

zur Erlangung des akademischen Grades

Master of Arts in Business

Betreuung und Beurteilung: Dipl.-Ing.in Eszter Geresics-Földi, BSc MSc

Zweitgutachten: Prof. Dr. Tom Gross

Wien, März 2022

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Masterarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.
3. dass die vorliegende Fassung der Arbeit mit der eingereichten elektronischen Version in allen Teilen übereinstimmt.

Gleisdorf, 06.03.2022

Unterschrift

Kurzzusammenfassung: Anforderungen an eine Software für das Energiedatenmanagement – Requirements Engineering und Anforderungsintegration am Beispiel der Software EMSControl

In den letzten Jahren rückte das Thema Energie aufgrund des nachweislich festgestellten Klimawandels immer weiter in den politischen und unternehmerischen Fokus. Durch geeignete Maßnahmen, wie die Einführung eines betrieblichen Energiemanagements, soll gewährleistet werden, dass in den Unternehmen die effiziente Energienutzung forciert wird. Ein Teilbereich des Energiemanagements ist das sogenannte Energiedatenmanagement, bei welchem mithilfe von Softwaresystemen Energiedaten von Smart Meter ausgelesen, Verbrauchsberechnungen durchgeführt und die Ergebnisse in managementtauglicher Form dargestellt werden. Für Hersteller einer Energiedatenmanagement-Software ist es essenziell, stets die aktuellen Anforderungen an die Software zu kennen, damit diese entsprechend in das System implementiert werden können. Dadurch soll die Marktkonformität des Produktes gewährleistet werden. Es existieren für diese Softwaresysteme einerseits rechtliche Anforderungen sowie andererseits kundenspezifische Anforderungen. Im Rahmen dieser Masterarbeit werden die aktuellen Anforderungen an eine Energiedatenmanagement-Software mithilfe des Requirements Engineerings ermittelt und dokumentiert. Im Anschluss wird auf Basis der identifizierten Anforderungen demonstriert, wie diese in die obsoleete Software EMSControl der Firma evon GmbH implementiert werden können. Dazu wird eine Ist-Analyse der Software durchgeführt und mittels dem Forschungsparadigma Design Science die Integration der Anforderung vorgenommen. Die Design Science Artefakte werden nach dem Konzept der objektorientierten Programmierung im EMSControl erzeugt. Die Evaluierung dieser Artefakte erfolgt durch eine Befragung von Stakeholdern der evon GmbH. Im letzten Schritt der Masterarbeit wird die neu erstellte Version des EMSControl für alle KundInnen freigegeben.

Schlagwörter:

Energiemanagement, Energiedatenmanagement, Software, ISO 50001, Requirements Engineering, Design Science, Objektorientierte Programmierung

Abstract: Requirements of a software for energy data management - requirements engineering and requirements integration using the example of the software EMSControl

In recent years, the topic of energy has moved more and more into the political and corporate focus due to the identified climate change. Appropriate measures, such as the introduction of an operational energy management system, are intended to ensure that companies use energy efficiently. One area of energy management is energy data management, in which software systems are used to collect energy data from smart meters, calculate consumption and present the results suitable for the management. For companies which develop energy data management software, it is essential to always know the requirements of their product in order to be able to implement them accordingly. This is to ensure the delivery of a product that is in line with the market. For these software systems, there are legal requirements on the one hand and customer-specific requirements on the other hand. Within the scope of this master thesis, the current requirements for an energy data management software are determined and documented with the methodologies of requirements engineering. Afterwards it will be demonstrated how these requirements can be implemented in the obsolete software EMSControl of the company evon GmbH. For this purpose, an as-is analysis of the software is carried out, followed by the integration of the requirement using Design Science. The Design Science artifacts are generated in the EMSControl according to the concept of object-oriented programming. The evaluation of these artifacts is done by a survey of stakeholders of evon GmbH. In the last step of the master thesis, the newly created version of the EMSControl is released for all customers.

Keywords:

Energy Management, Energy Data Management, Software, ISO 50001, Requirements Engineering, Design Science, Object-Oriented Programming

Danksagung

Ich möchte mich bei all denjenigen bedanken, die mir bei der Erstellung meiner Masterarbeit geholfen haben. Besonders möchte ich mich bei meiner Freundin Klara bedanken, die mich immer unterstützt hat. Sei es beim Korrekturlesen der Arbeit, beim Verständnis zeigen, wenn ich wenig Zeit für sie hatte oder einfach beim Zuhören, wenn ich von meinen Problemen erzählte. Ein weiterer großer Dank geht an meine Eltern. Sie hatten immer ein offenes Ohr für mich und unterstützten mich in jeglicher Hinsicht während der stressigen Zeit.

Auch möchte ich mich im Besonderen bei meiner Betreuerin Frau DI Eszter Geresics-Földi bedanken. Sie betreute mich sehr gut in der Umsetzung und gab mir stets konstruktives Feedback innerhalb kürzester Zeit. Auch wurde es von ihr sehr wertgeschätzt, was es bedeutet, eine Masterarbeit neben einem Vollzeitjob zu verfassen.

Danke auch an die ExpertInnen, die sich die Zeit für die Interviews genommen haben.

Abschließend möchte ich mich noch bei meinen KollegInnen der Firma evon GmbH bedanken, die mir immer wieder mit Rat und Tat zur Seite standen.

Inhaltsverzeichnis

1	EINLEITUNG	1
1.1	Ausgangssituation	1
1.1	Problemstellung und Problemrelevanz	3
1.2	Arbeitsziel und Forschungsfrage	5
1.3	Methodische Vorgangsweise	7
2	ENERGIEMANAGEMENT	9
2.1	Definition und Ziele des Energiemanagements	9
2.2	Gesetzliche Rahmenbedingungen	10
2.2.1	Österreich	11
2.2.2	Deutschland	12
2.3	Ökonomische, ökologische und soziale Motive	14
2.4	ISO 50001	15
2.5	Aufbau eines Energiemanagementsystems nach ISO 50001	16
2.5.1	Führung und Energiepolitik	18
2.5.2	Planung	18
2.5.3	Unterstützung und Betrieb	20
2.5.4	Bewertung der Leistung	21
2.5.5	Verbesserung	23
3	ANFORDERUNGEN AN EINE ENERGIEDATENMANAGEMENT-SOFTWARE	25
3.1	Energiedatenmanagement-Software	25
3.2	Anforderungsidentifizierung	29
3.2.1	Anforderungen nach ISO 50001	30
3.2.2	Anforderungen des Bundesamtes für Wirtschaft und Ausfuhrkontrolle	32
3.2.3	Anforderungen des Marktspiegels der Energieagentur Nordrhein-Westfalen	33
3.2.4	Kundenspezifische Anforderungen	34
3.3	Anforderungsdokumentation	35
3.4	Anforderungsprüfung	37
3.5	Anforderungsverwaltung	38

4	IST-ANALYSE DES EMSCONTROL	39
4.1	evon GmbH	39
4.2	XAMControl	39
4.2.1	Allgemein	39
4.2.2	Technologie	40
4.2.3	Systemarchitektur	41
4.3	Systemmodule	42
4.4	Objektorientierung im XAMControl	43
4.5	EMSCControl	46
4.5.1	Allgemein	46
4.5.2	Technische Funktionsweise	47
4.5.3	Features	49
5	ANFORDERUNGSINTEGRATION MITTELS DESIGN SCIENCE	53
5.1	Design Science	53
5.2	Analyse	57
5.3	Entwurf	59
5.3.1	Online-Wetterdaten	62
5.3.2	Witterungsbereinigung	65
5.3.3	MSCONS	68
5.3.4	Zählerreset	71
5.3.5	Parametrierung der Einheiten	71
5.3.6	Kostenstellenzuordnung	72
5.3.7	Liniendiagramm	75
5.3.8	Mittelwerte	77
5.3.9	Automatische Ersatzwertbildung	78
5.3.10	Automatische Plausibilitätsprüfung	80
5.3.11	Kennzahlengenerierung und -darstellung	82
5.4	Evaluation	87
5.4.1	Methodische Vorgehensweise	88
5.4.2	Qualitative Auswertung der ExpertInneninterviews	88
5.4.3	Weitere Iteration	89
5.5	Diffusion	91
5.5.1	Veröffentlichung des neuen EMSControl	91
5.5.2	Veröffentlichung der aktuellen Marktanforderungen	92
5.5.3	Lessons Learned	92

6	RESÜMEE	94
6.1	Zusammenfassung	94
6.2	Kritische Reflexion und Ausblick	96
	LITERATURVERZEICHNIS	97
	ABBILDUNGSVERZEICHNIS	108
	TABELLENVERZEICHNIS	111
	FORMELVERZEICHNIS	112
	QUELLCODEVERZEICHNIS	112
	ABKÜRZUNGSVERZEICHNIS	114
	ANHANG A	115
	ANHANG B	116
	ANHANG C	118
	ANHANG D	145

1 Einleitung

1.1 Ausgangssituation

Effiziente und langfristige Nutzung von begrenzt verfügbaren Ressourcen sowie die Reduzierung von Kohlenstoffdioxid-, sprich CO₂-Emissionen, sind seit Jahren präsen- te Themen in Politik, Wirtschaft und Medien. Dabei wird zwischen energiebedingten und prozessbedingten Emissionen unterschieden. Knapp 75 Prozent der CO₂-Emissionen sind energiebedingt (Umweltbundesamt, 2020, S. 65). Diese Treibhausgas-Emissionen entstehen bei der Umwandlung von Energieträgern. Beispielsweise wird CO₂ bei der Verbrennung von Kohle für die Stromerzeugung freigesetzt. Bei prozessbedingten CO₂-Emissionen handelt es sich um Emissionen, die bei bestimmten industriellen Herstellungsprozessen entstehen (Weber, 2014, S. 247). Da der größte Teil an Emissionen auf die Erzeugung von Elektrizität und Wärme zurückzuführen ist, liegt ein großer Fokus auf der effizienten Nutzung dieser Ressourcen. Abbildung 1 zeigt eine Statistik über die prozentuale Verteilung der energiebedingten CO₂-Emissionen weltweit nach Sektor im Jahr 2018, welche im Jahr 2020 von der Internationalen Energieagentur veröffentlicht wurde. Dabei ist zu erkennen, dass mit einem Anteil von 42 Prozent die Elektrizitäts- und Wärmeerzeugung der größte Verursacher der energiebedingten CO₂-Emissionen ist (IEA, zitiert nach de.statista.com, 2020).

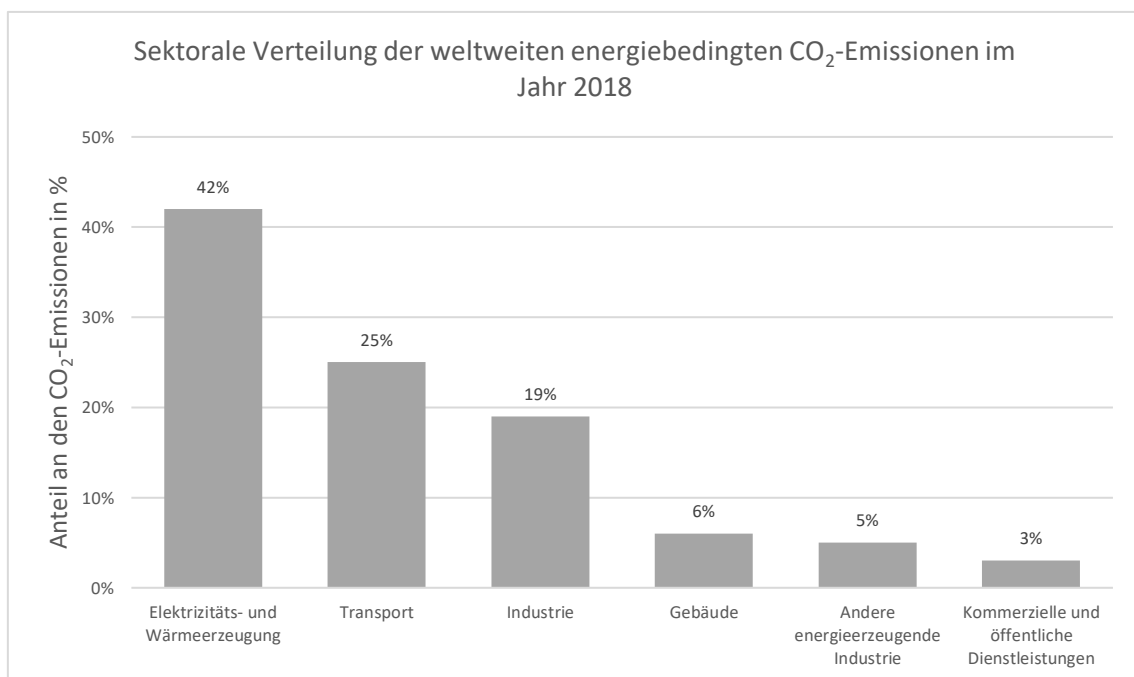


Abbildung 1: Sektorale Verteilung der weltweiten energiebedingten CO₂-Emissionen im Jahr 2018 (In Anlehnung an IEA, zitiert nach de.statista.com, 2020)

Des Weiteren zeigen die Energiebilanzen der Statistik Austria eine Steigerung des Energieverbrauchs von 1990 bis 2019 um 33 Prozent. Dabei entfielen im Jahr 2019 nur knapp 30 Prozent der verbrauchten Energie auf private Haushalte (Statistik Austria, 2020). Dies gibt Aufschluss, warum Regierungen die effiziente Energienutzung in Unternehmen forcieren und mit Gesetzen vorantreiben möchten. Dabei sollen diese politischen Maßnahmen nicht nur kurzfristige Kosteneffekte für Unternehmen erzielen, sondern langfristig dem Klimawandel entgegenwirken und somit die globale Erderwärmung stoppen (Geilhausen, Bränzel, Engelmann & Schulze, 2015, S. 3). In Österreich wird dies durch das Bundes-Energieeffizienzgesetz (EEffG) umgesetzt. Nach § 9 EEffG müssen Unternehmen – abhängig von ihrer Größe – Maßnahmen setzen, um die Energieeffizienz zu verbessern. Unternehmen, welche der Norm unterliegen, sind verpflichtet, ein Energiemanagementsystem (EnMS) in ihrem Betrieb zu implementieren. Mithilfe eines EnMS werden Unternehmen dabei unterstützt, Energieziele zu planen, daraus abgeleitete Maßnahmen zu realisieren und diese im Anschluss durch Analyse der Energiedaten auszuwerten sowie etwaige Korrekturen durchzuführen. Dieser gesamtheitliche Unternehmensprozess wird in der ISO 50001 Norm als Plan-Do-Check-Act-Zyklus (PDCA-Zyklus) bezeichnet (Geilhausen et al., 2015, S. 12).

Für Softwareunternehmen, welche Programme für EnMS entwickeln, ergeben sich somit Anknüpfungspunkte zu den rechtlichen Anforderungen aufgrund der gesetzlichen Normen. Die Unternehmen müssen diese Anforderungen identifizieren und in ihr System einpflegen, um marktkonform zu sein. Darüber hinaus existieren auch kundenspezifische Anforderungen, welche in die Systeme integriert werden müssen, um am Markt wettbewerbsfähig zu bleiben. Daher ist es für diese Unternehmen essenziell, kontinuierlich die Marktanforderungen mittels geeignetem Requirements Engineering zu ermitteln und die Ergebnisse den Softwareentwicklern für eine Implementierung zur Verfügung zu stellen.

Das EMSControl der evon GmbH stellt dabei eine nach ISO 50001:2011 zertifizierte Software für das Energiedatenmanagement (EDM) dar, mit welcher es den NutzerInnen ermöglicht wird, Daten aus smarten Energiemessgeräten auszulesen und zu analysieren. Mithilfe der Messdaten werden Verbräuche berechnet und diese für das Management eines Unternehmens in adäquater Weise zur Verfügung gestellt. Das EMSControl setzt dabei im Punkt „Check“ des PDCA-Zyklus an und soll die Überwachung und Auswertung der Energiedaten im Unternehmen vereinfachen. Der Vorteil dieses Systems ist die Unabhängigkeit hinsichtlich der eingesetzten Hardware, da die gängigsten Bussysteme aus der industriellen Leittechnik unterstützt werden. U. a. können die Daten der Zähler

über bewährte Protokolle der Gebäudeleittechnik wie Modbus, MBus oder BACNet ausgelesen und verarbeitet werden (Ahmad, Mourshed, Mundow, Sisinni & Rezgui, 2016, S. 85; Evon, 2021).

1.1 Problemstellung und Problemrelevanz

Aufgrund des nachweislich festgestellten Klimawandels versucht die Politik durch geeignete Maßnahmen in Form von Verordnungen und Gesetzen eine Trendumkehr einzuleiten und dadurch den Umweltschutz aktiv zu fördern (Janson, 2020). Aufgrund dieser Regularien ändern sich auch die Anforderungen an EnMS. Beispielsweise dürfen seit Beschließung des EEffG iSd § 9 EEffG, ausschließlich ISO 50001 zertifizierte EnMS eingesetzt werden. Neben den Anforderungen aufgrund von gesetzlichen Regularien, ändern sich auch die kundenspezifischen Anforderungen. Hersteller versuchen stets, die aktuellen Trends in ihre Systeme einzupflegen, um die Anforderungen der Stakeholder zu erfüllen. Für eine EDM-Software, wie das EMSControl, stellt sich somit das Problem, die unterschiedlichsten Marktanforderungen adäquat zu identifizieren, um einerseits eine gesetzeskonforme Software zur Verfügung zu stellen und andererseits den Wünschen der Stakeholder zu entsprechen. Aus diesem Grund sollen im Rahmen dieser Masterarbeit mithilfe eines Requirements Engineerings die aktuellen Marktanforderungen von EDM-Softwares erhoben, dokumentiert, validiert und verwaltet werden.

Ferner stellt sich das Problem, wie die erhobenen und geprüften Anforderungen aus dem Requirements Engineering in eine bestehende Software integriert werden können. Dies wird anhand des EMSControl der evon GmbH demonstriert, indem auf die Methodik des Design Science zurückgegriffen wird, um den Stakeholdern neue Artefakte in Form von Instanziierungen zur Verfügung zu stellen. Zusätzlich muss die Qualität der erstellten Artefakte adäquat überprüft werden, um einen Release einer unzureichend erstellten Software zu vermeiden.

Angesichts der immer größer werdenden Relevanz von EnMS, ist es für Hersteller unumgänglich, die Software stets up to date zu halten, um die Veränderungen am Markt hinsichtlich gesetzlichen Vorschriften und Kundenanforderungen berücksichtigen zu können. Durch eine rechtzeitige Implementierung der Anforderungen soll sichergestellt werden, dass die Systeme wettbewerbsfähig bleiben und nicht obsolet werden. Aufgrund eines fehlenden Requirements Engineering scheitern knapp 24 Prozent der Softwareprojekte, da die Anforderungen unzureichend ermittelt oder eingebunden wurden (Matyokurehwa, Mavetera & Jokonya, 2017, S. 14). Aus diesem Grund wird die

Relevanz dieser Masterarbeit begründet, dass durch neue oder angepasste Gesetze, der Einführung der ISO 50001:2018 sowie den sich ändernden Anforderungen der Stakeholder, keine gesammelten Dokumentationen der aktuellen Marktanforderungen hinsichtlich EDM-Softwares existieren. Demnach müssen die Softwarehersteller eigenständig die verschiedensten gesetzlichen Normen prüfen sowie Feedback der Stakeholder einholen, um die Marktanforderungen zu identifizieren. Diese Lücke soll mittels der Masterarbeit geschlossen werden, indem die notwendigen Anforderungen an EDM-Softwares mittels Requirements Engineering erhoben werden.

Darüber hinaus stellt eine Software, welche nicht mehr die zeitgemäßen Anforderungen erfüllt, ein Problem für die Unternehmen dar, da die Stakeholder aufgrund der obsoleten Software den Anbieter wechseln werden. Eine solche Software befindet sich nach den Produktlebenszyklus-Theorien in der Sättigungsphase, da der Umsatz stagniert bzw. sinkt und das Wachstum zum Stillstand kommt. Um einen Marktaustritt in der Degenerationsphase zu vermeiden, wird durch proaktive Maßnahmen versucht, die Marktanteile wieder zu steigern. Neben Marketingkampagnen können diese Maßnahmen auch Produktveränderungen bzw. Weiterentwicklungen sein, welche in der Fachsprache als Relaunch bezeichnet werden (Thommen, Achleitner, Gilbert, Hachmeister & Kaiser, 2017, S. 86). Abbildung 2 stellt die einzelnen Phasen des Produktlebenszyklus dar.

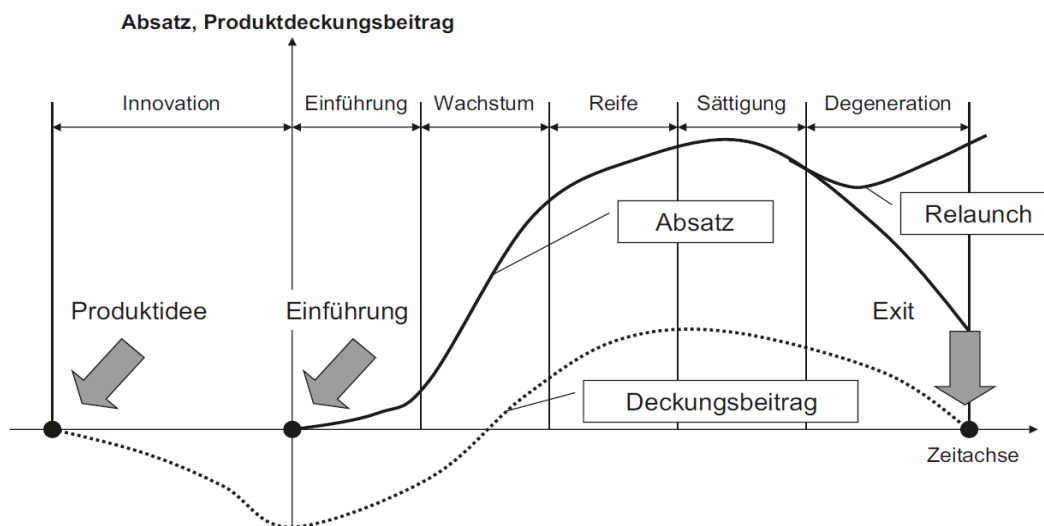


Abbildung 2: Phasen des Produktlebenszyklus (Aumayr, 2019, S. 295)

Am Beispiel der evon GmbH wurde vom Autor festgestellt, dass bei Betrachtung interner Unternehmensdaten die Absatzzahlen des EMSControl leicht sinken und demnach ausgesagt werden kann, dass sich das Produkt in der Sättigungsphase befindet. In dieser Arbeit soll demnach gezeigt werden, wie eine Integration der identifizierten

Anforderungen aus dem Requirements Engineering mittels Design Science in die bestehende Software erfolgen kann, um den Marktaustritt der Software zu vermeiden.

Ferner muss bedacht werden, dass aufgrund der Aktualität angesichts des Umweltschutzes, das Arbeitsthema Energiemanagement eine Relevanz darstellt. Es ist davon auszugehen, dass in den kommenden Jahren der Umweltschutz von der Politik weiter forciert wird und die Maßnahmen zur Überwachung von betrieblichen Energieverbräuchen bestehen bleiben bzw. erweitert werden. Aus diesem Grund wird im Laufe des Jahres 2022 eine Erweiterung des EEffG, das Energieeffizienzgesetz-Neu, in Kraft treten. Weiters wurde durch das Pariser Klimaabkommen aus dem Jahr 2015 ein langjähriger Plan zwischen der Europäischen Union und weiteren 174 Ländern vereinbart, um aktiven Klimaschutz zu betreiben (Dehli, 2020, S. 1; Gonzalez-Perez, 2016, S. 2). Zusätzlich zeigen die Ergebnisse aus einer Studie der deutschen Bundesstelle für Energieeffizienz beim Bundesamt für Wirtschaft und Ausfuhrkontrolle eine positive Marktentwicklung hinsichtlich EnMS (BfEE, 2018). Ebenfalls weist eine Studie im Auftrag des Bundesministeriums für Wissenschaft, Forschung und Wirtschaft auf große Marktpotenziale im Bereich des Energiemanagements hin (Windsperger, Windsperger & Schörner, 2014, S. 20–21). Somit ist gewährleistet, dass Systeme, welche dabei unterstützen den Energieverbrauch zu senken, auch in Zukunft ein Marktpotenzial aufweisen.

1.2 Arbeitsziel und Forschungsfrage

Wie bereits erwähnt, gibt es aufgrund verschiedener gesetzlicher Vorschriften unterschiedlichste Anforderungen an EDM-Softwares und durch Novellierungen der Gesetze müssen die Anforderungen stets adaptiert werden. Durch den vermehrten Einsatz der EDM-Systeme entstehen auch neue kundenspezifische Anforderungen, welche die Softwarehersteller berücksichtigen müssen, um marktfähig zu bleiben. Daher setzt sich diese Arbeit zum Ziel, durch eine Anforderungserhebung mithilfe Requirements Engineering, die aktuellen gesetzlichen und kundenspezifischen Marktanforderungen hinsichtlich EDM-Softwares zu erheben. Dadurch soll für die Softwarehersteller eine Basis für die Erstellung bzw. Überarbeitung der EDM-Softwares geschaffen werden.

Des Weiteren soll durch diese Arbeit gezeigt werden, wie die identifizierten Anforderungen mittels Design Science in eine bestehende EDM-Software implementiert werden können. Dies erfolgt am Beispiel des EMSControl, welches aktuell obsolet ist und durch die Anforderungsintegration die Wettbewerbsfähigkeit zu steigern. Dazu muss

zuerst eine Ist-Analyse des Systems durchgeführt werden, um diese anschließend mit den Anforderungen aus dem Requirements Engineering vergleichen zu können. Die dadurch identifizierten Disparitäten werden im Rahmen der Masterarbeit in das System eingepflegt. Diese Anforderungsintegration in das bestehende System erfolgt je nach Anforderung mit den Hochsprachen C#, XML, Function Block Diagram (FBD) oder SQL. Zum Abschluss wird durch qualitative ExpertInneninterviews evaluiert, ob durch die Änderungen an der Software, eine Marktkonformität des Produktes erzielt wurde.

Diese Arbeit befasst sich ausschließlich mit den rechtlichen und kundenspezifischen Anforderungen an EDM-Softwares für die Länder Österreich und Deutschland, da die evon GmbH im Bereich des EnMS ausschließlich in diesen Ländern operiert. Aus diesem Grund werden auch die ExpertInneninterviews für die Validierung der Artefakte mit Stakeholdern aus Österreich und Deutschland geführt.

Um das bestehende Problem einer fehlenden Übersicht über die aktuellen Marktanforderungen von EDM-Softwares zu lösen, muss folgende Hauptforschungsfrage beantwortet werden:

Welches Modell einer objektorientierten EDM-Software leitet sich aus den rechtlichen und kundenspezifischen Marktanforderungen ab, um ein marktkonformes Produkt anbieten zu können?

Um diese Hauptforschungsfrage beantworten zu können, müssen folgende Subforschungsfragen beantwortet werden:

- Welche rechtlichen und kundenspezifischen Anforderungen können durch Requirements Engineering am österreichischen und deutschen Markt identifiziert werden?
- In welcher Form kann eine adäquate Anforderungsintegration in das EMSControl durchgeführt werden?
- Kann nach der Anforderungsintegration mithilfe von qualitativen ExpertInneninterviews eine Marktkonformität des erstellten Artefakte bestätigt werden?

1.3 Methodische Vorgangsweise

Zu Beginn der Masterarbeit erfolgt eine allgemeine Einführung in das Themengebiet Energiemanagement. Dabei wird der Begriff Energiemanagement abgegrenzt und es werden die Ziele und Motive von EnMS erläutert. Dieses Kapitel wird mithilfe der Methode der Literaturrecherche erstellt. Für die Literaturrecherche wird bereits vorhandene Literatur aus dem Fachgebiet herangezogen, um dadurch den Stand der Technik zu erörtern (Rowley & Slack, 2004, S. 32; Xiao & Watson, 2019, S. 95). Als Datenbanken dienen dabei IEEE Xplore, SpringerLink, Google Scholar als auch die Rechtsinformationssysteme (RIS) für österreichische und deutsche Gesetzestexte.

Im anschließenden Kapitel werden die aktuellen Marktanforderungen mittels Requirements Engineering ermittelt. Das Requirements Engineering soll Unternehmen dabei unterstützen, die Anforderungen der Stakeholder proaktiv in die Entwicklung des Produktes einfließen zu lassen. Der Prozess besteht aus den Hauptaktivitäten Anforderungserhebung, -dokumentation, -prüfung und -verwaltung (Paetsch, Eberlein & Maurer, 2003, S. 308–311; Pohl & Rupp, 2015, S. 3–5).

Für die Ermittlung der rechtlichen Anforderungen wird zuerst eine Literaturrecherche durchgeführt. Um die aktuellen Anforderungen des deutschen und österreichischen Marktes aus Kundensicht zu identifizieren, stehen verschiedenste Ermittlungstechniken im Rahmen von Requirements Engineering zur Verfügung. Unter Betrachtung der Entscheidungsmatrix nach RUPP et al. wurden für diese Masterarbeit aufgrund der hohen Anzahl an Stakeholdern die Ermittlungstechniken der Selbstaufschreibung und der Literaturrecherche gewählt, da es im Rahmen der Masterarbeit nicht möglich ist, alle Stakeholder der evon GmbH zu befragen (2009, S. 96–98). Bei der Selbstaufschreibung dokumentieren die Stakeholder ihre Anforderungen, Änderungs- und Optimierungsvorschläge selbst (Rupp & Schüpferling, 2014, S. 98). Für die Selbstaufschreibung werden im Rahmen der Masterarbeit bereits vorhandene Aufzeichnungen, wie E-Mails oder Feedbacks der Stakeholder der evon GmbH herangezogen, um die kundenspezifischen Anforderungen zu sammeln und zu analysieren. Darüber hinaus wird auch hier die Literaturrecherche angewandt, um diese Anforderungen zu identifizieren.

Die Ergebnisse der Anforderungserhebung werden anschließend adäquat dokumentiert. Da nicht alle erhobenen Anforderungen aus der Befragung eine gute Qualität aufweisen müssen, ist eine Validierung vonnöten. Dies erfolgt in Form eines Walkthroughs (Maalem

& Zarour, 2016, S. 17; Pohl & Rupp, 2015, S. 106–107). Das Resultat aus der Validierung wird in einem Katalog festgehalten, welcher zeigt, welche Anforderungen in aktuellen EDM-Softwares umgesetzt sein müssen, um wettbewerbsfähig zu sein.

Anhand des EMSControl soll demonstrativ veranschaulicht werden, wie eine wissenschaftlich fundierte Anforderungsintegration mit der problemlösungs- und konstruktionsorientierten Forschungsmethode Design Science durchgeführt werden kann. Ziel des Design Science ist die Lösung von praktischen Problemen aus der realen Welt durch Erstellung von zielgerichteten Artefakten (Winter, 2008, S. 470–471). Als Software Development Life Cycle Modell für die Entwicklung des Artefakts wird auf das Wasserfallmodell, das von Winston Royce 1970 konzipiert wurde und in der Praxis verbreitet eingesetzt wird, zurückgegriffen (Balaji & Murugaiyan, 2012, S. 26–28; Burghardt, 2018, S. 151–152). Dies wird dadurch begründet, dass das EMSControl ebenfalls nach diesem Phasenmodell entwickelt wurde und agile Modelle, wie SCRUM, nicht für die Entwicklung derartiger Software bei der evon GmbH zum Einsatz kommen. Neben der Entwicklung der Artefakte ist ein wesentlicher Bestandteil von Design Science, eine adäquate Evaluation. Dabei wird geprüft, wie gut die entwickelten Artefakte funktionieren und damit die Anforderungen erfüllen (Deng & Ji, 2018, S. 14; Venable, Pries-Heje & Baskerville, 2016, S. 79). Darüber hinaus sollen in der Evaluierung die erstellten Artefakte praxisbezogen getestet werden. Dabei wird untersucht, ob das gegebene Artefakt auch für die Praxis geeignet ist. Nach Betrachtung der möglichen Evaluierungsmethoden wurden die ExpertInneninterviews ausgewählt, um die entwickelten Artefakte zu validieren (Sonnenberg & vom Brocke, 2012, S. 393–395). Für diese Untersuchung wurden vier Stakeholder der evon GmbH herangezogen, welche das EMSControl aktuell einsetzen und über Mails und Feedbacks bereits aktive Anforderungen an das EMSControl übermittelt haben.

Nach erfolgreicher Validierung der erstellten Artefakte hinsichtlich der rechtlichen und kundenspezifischen Anforderungen wird das überarbeitete EMSControl offiziell freigegeben und den Stakeholdern als Download zur Verfügung gestellt.

2 Energiemanagement

2.1 Definition und Ziele des Energiemanagements

Mithilfe des betrieblichen Energiemanagements kann durch gezielt koordinierte Energienutzung ein wesentlicher Beitrag zur Erhaltung der Umwelt geleistet sowie eine Erhöhung der Wirtschaftlichkeit des Unternehmens geschaffen werden. Kurz ausgedrückt dient das Energiemanagement der Handhabung der Energie im Unternehmen.¹ Bei der Umsetzung werden zwei verschiedene Ansätze differenziert. Einerseits liegt beim technischen Ansatz der Fokus bei den Anlagen, welche die Energie verbrauchen. Hierbei werden die jeweiligen Maßnahmen rein technisch ausgearbeitet und führen zum Ausscheiden bzw. zur Neuanschaffung der Anlagen. Andererseits werden beim organisatorischen Ansatz die Unternehmensprozesse stärker betrachtet. Darüber hinaus wird bei diesem Ansatz ein Schwerpunkt auf die Bewusstseinsentwicklung für Ressourcenschonung und Nachhaltigkeit gelegt (Geilhausen et al., 2015, S. 3–4; Kals & Würtenberger, 2012, S. 73; Posch, 2011, S. 147–148).

Die organisatorische Umsetzung eines Energiemanagements erfolgt in Form eines EnMS und wird aufgrund der Legislative am häufigsten auf Basis der ISO 50001 eingeführt (Geilhausen et al., 2015, S. 4). Das EEffG definiert EnMS in § 5 Abs 1 Z 12 als „anerkannte regelgebundene Managementsysteme, welche insbesondere oder auch die Energieflüsse in einem Unternehmen erfassen, abbilden und bewerten und Vorschläge für Einsparmaßnahmen generieren“. Bei EnMS werden die zuvor genannten technischen und organisatorischen Ansätze zu einer ganzheitlichen Betrachtungsweise verbunden. Für die erfolgreiche Umsetzung muss das EnMS in die Unternehmensstrategie und Organisation integriert werden (Geilhausen, 2015, S. 1). Die Einführung eines solchen EnMS dient neben der Erfüllung von gesetzlichen Normen auch für das Erreichen folgender Ziele (Geilhausen et al., 2015, S. 7–8):

- Senkung der Energiekosten bzw. des Energieverbrauchs
- Nutzung von Steuerentlastungen
- Beitrag zum Umweltschutz
- Steigerung der langfristigen Energieeffizienz
- Verbesserung des Unternehmensimages

¹ Nach der ISO 50001:2018 werden mit dem Begriff Energie die Medien Elektrizität, Brennstoff, Dampf, Wärme, Druckluft und vergleichbare Medien assoziiert (ISO, 2018, S. 6).

2.2 Gesetzliche Rahmenbedingungen

Das Thema Energie rückte in der Industrie als auch in privaten Haushalten in den letzten Jahren weiter in den politischen Fokus. Damit einher gingen eine Vielzahl von Auflagen und Verpflichtungen, aber auch Steuererleichterungen für besonders engagierte Unternehmen (Schmitt & Günther, 2014, S. 10). Durch internationale Staatsabkommen – wie dem Kyoto-Protokoll aus dem Jahr 1997 oder dem Pariser Abkommen aus dem Jahr 2015 – haben sich die beteiligten Länder bzw. die Europäische Union dazu verpflichtet, diverse Maßnahmen zur Reduktion von Treibhausgasemissionen sowie zur Begrenzung der Erderwärmung durchzuführen. Im Kyoto-Protokoll wurde vereinbart, dass die teilnehmenden Länder dazu verpflichtet sind, ihren jährlichen Emissionen an Treibhausgasen bis 2012 um mindestens fünf Prozent gegenüber dem Stand von 1990 zu verringern. Das völkerrechtliche Abkommen trat 2005 in Kraft und wurde bis 2020 verlängert. Ein Nachteil des Kyoto-Protokolls war allerdings, dass es nur Industrieländer zu den definierten Zielen verpflichtete. Allerdings war der größte Zuwachs an Treibhausgasemissionen in den vergangenen Jahrzehnten bei den Entwicklungsländern festzustellen (UNFCCC, 1997, S. 9; Dröge, 2015, S. 25; BMK, 2021). Im Pariser Abkommen, welches 2020 in Kraft trat und das Kyoto-Protokoll dementsprechend ablöste, wurde der Temperaturanstieg der Erde auf 1,5 °C begrenzt (Gonzalez-Perez, 2016, S. 2; BMK, 2021). Die Länder sind für das Erreichen dieser Ziele selbst verantwortlich und müssen dahingehend nationale Maßnahmen – z. B. in Form von Gesetzen – beschließen, um diese Zielvorgaben zu erfüllen.

Von der Europäischen Union wurden bereits mehrere Richtlinien zur Umsetzung der völkerrechtlichen Abkommen veröffentlicht. Die bedeutsamste Richtlinie innerhalb der EU ist die Richtlinie 2012/27/EU, welche auch als Energieeffizienzrichtlinie (EnEff-RL) bezeichnet wird. Diese wurde im Oktober 2012 durch das Europäische Parlament und den EU-Rat beschlossen und trat im Dezember 2012 in Kraft. Die jeweiligen Mitgliedsstaaten mussten die Richtlinie bis zum 5. Juni 2014 in einem nationalem Gesetz umsetzen. Durch die EnEff-RL wurde ein unionsrechtlicher Rahmen zur Forcierung der Energieeffizienz geschaffen sowie konkrete Maßnahmen zur Erreichung der festgelegten Ziele definiert. Diese Ziele beinhalteten bis 2020 eine Senkung der Treibhausgasemissionen um mindestens 20 Prozent, eine Energieeffizienzsteigerung um 20 Prozent und einen Anstieg des Anteils erneuerbarer Energien am Gesamtenergieverbrauch auf 20 Prozent gegenüber 1990. Sie lässt allerdings auf instrumenteller Ebene große Spielräume für die nationalen Gesetzgeber (Brauner, 2014, S. 114–115). Im Jahr 2018 erfolgte mit der Richtlinie

2018/2002/EU eine Novellierung der EnEff-RL, bei welcher weitere Energieeinsparziele bis zum Jahr 2030 und Umsetzungsfristen für fernablesbare Zähler (Smart Metering) definiert wurden. Zudem hat die Europäische Kommission am 14. Juli 2021 im Rahmen des „Fit for 55“-Paketes einen Vorschlag für die Überarbeitung der EnEff-RL vorgestellt (Europäische Kommission, 2021).

Hinsichtlich EnMS ist vor allem Artikel 8 in der EnEff-RL relevant. Dieser legt unter Abs 4 fest, dass Unternehmen, welche nach der EU-Empfehlung 2003/361/EG nicht zu den kleinen und mittleren Unternehmen (KMU) zählen, alle vier Jahre Energieaudits durch externe, unabhängige Experten durchführen lassen müssen. Nach der EU-Empfehlung 2003/361/EG wird ein Unternehmen als KMU bezeichnet, wenn dieses weniger als 250 MitarbeiterInnen beschäftigt und einen Jahresumsatz von weniger als 50 Millionen Euro erwirtschaftet oder eine Bilanzsumme von maximal 43 Millionen Euro aufweist. Unternehmen, die keine KMU sind und dahingehend externe Energieaudits durchführen müssten, können nach Art 8 Abs 6 RL 2012/27/EU alternativ ein, nach europäischen bzw. internationalen Normen zertifiziertes EnMS einsetzen, wenn dieses die im Anhang VI der RL 2012/27/EU beschriebenen Mindestkriterien erfüllt.

Nachfolgend wird die Umsetzung der EnEff-RL hinsichtlich EnMS in das österreichische und deutsche Recht erläutert. Wie in Kapitel 1.2 erwähnt, werden ausschließlich diese zwei Länder betrachtet, da die Software der evon GmbH, welche im weiteren Verlauf der Masterarbeit optimiert wird, ausschließlich in diesen Ländern eingesetzt wird.

2.2.1 Österreich

Österreich hat für die Umsetzung des Unionsrechts in das nationale Recht im Juli 2014 das EEffG beschlossen und im August 2014 kundgemacht. Neben dem Ziel der Umsetzung der EnEff-RL nach § 3 EEffG besteht iSd § 2 EEffG der Zweck dieses Bundesgesetzes darin, bis Ende 2020 einen Beitrag zur Verwirklichung einer kostenoptimierten, nachhaltigen und gesicherten Energieversorgung zu leisten. In § 9 Abs 2 EEffG wurde der Art 8 Abs 6 RL 2012/27/EU umgesetzt. Demnach haben große Unternehmen nach § 9 Abs 2 Z 1 lit a EEffG die Verpflichtung ab 2015 alle vier Jahre ein externes Energieaudit durchzuführen. Anstelle dieses Audits können sich iSd § 9 Abs 2 Z 1 lit b EEffG diese Unternehmen dafür entscheiden, ab dem 01. Jänner 2015

- „ein zertifiziertes Energiemanagementsystem in Übereinstimmung mit der Norm EN 16001 oder der ISO 50001 oder entsprechenden Nachfolgenormen oder

- ein zertifiziertes Umweltmanagementsystem gemäß ISO 14000 oder entsprechende Nachfolgenormen oder gemäß Art 13 der Verordnung (EG) Nr 1221/2009 über die freiwillige Teilnahme von Organisationen an einem Gemeinschaftssystem für Umweltmanagement und Umweltbetriebsprüfung oder
- ein einem Energiemanagement- oder Umweltmanagementsystem gleichwertiges, innerstaatlich anerkanntes Managementsystem

einzuführen, das auch ein regelmäßiges internes oder externes Energieaudit gemäß § 17 und § 18 umfassen muss.“

Entscheidet sich das Unternehmen für die Einführung eines EnMS, so ist nach Erläuterung RV 182 BlgNR 25 GP 19 ein – nach ISO 50001 – zertifiziertes Managementsystem anzuwenden. Ist im Unternehmen bereits ein nicht nach ISO 50001 zertifiziertes EnMS implementiert, gestattet der Gesetzgeber dieses System weiterhin zu verwenden und sieht von der Pflicht zur Anwendung eines neuen EnMS gemäß ISO 50001 ab. Aus diesem Grund ist die ISO 50001 die vorwiegend verwendete Norm für EnMS und wird im weiteren Verlauf der Arbeit detailliert erläutert. Kommen die Unternehmen den gesetzlichen Verpflichtungen nicht nach, drohen je nach Tatbestand iSd § 31 EEffG Geldstrafen zwischen 10.000 Euro und 100.000 Euro. Die Zuständigkeit für die Kontrollen und Evaluierungen der Energieeffizienzmaßnahmen liegt bei der nach § 24 EEffG eigens eingerichteten nationalen Energieeffizienz-Monitoringstelle.

Da einige Verpflichtungen des EEffG mit 31.12.2020 ausliefen und durch die Richtlinie 2018/2002/EU Änderungen an der EnEff-RL durchgeführt wurden, war mit Beginn 2021 ein Energieeffizienzgesetz-Neu geplant. Allerdings befindet sich dieses mit Stand März 2022 noch in Ausarbeitung. Infolgedessen wurden für das Jahr 2021 und 2022 Übergangsregelungen definiert und rechtliche Verpflichtungen zur Setzung von Energieeffizienzmaßnahmen für die, in der Richtlinie 2018/2002/EU definierte, Verpflichtungsperiode von 2021 bis 2030 entstehen erst ab dem Zeitpunkt des Inkrafttretens des Energieeffizienzgesetz-Neu (BMK, o. D.; Köppl & Schleicher, 2021, S. 1).

2.2.2 Deutschland

In Deutschland erfolgte die Umsetzung der EnEff-RL durch die Novellierung des Gesetzes über Energiedienstleistungen und andere Energieeffizienzmaßnahmen (EDL-G). Demnach sind iSd § 8 EDL-G große Unternehmen spätestens 20 Monate nach Inkrafttreten des Gesetzes bzw. 20 Monate nach dem Erlangen des Status eines großen

Unternehmens verpflichtet, ein Energieaudit durchzuführen. Auch in Deutschland wird für die Definition eines großen Unternehmens die EU-Empfehlung 2003/361/EG herangezogen. Nach dem erstmaligen Audit ist es erforderlich, im periodischen Abstand von vier Jahren ein weitere Energieaudits nachzuweisen. Es besteht auch nach dem EDL-G die Möglichkeit, alternativ ein zertifiziertes EnMS oder Umweltmanagementsystem anzuwenden. Entscheidet sich das Unternehmen für das EnMS, muss dieses iSd § 2 Z 17 EDL-G nach ISO 50001 zertifiziert sein. In Deutschland erfolgt nach § 9 EDL-G die Kontrolle über die Einhaltung der gesetzlichen Normen durch das Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA). Wie auch in Österreich werden bei einer Nichteinhaltung der Vorschriften Geldstrafen verhängt. Diese können laut § 12 Abs 2 EDL-G bis zu 50.000 Euro betragen.

Zudem sieht die Legislative steuerliche Vorteile in Form von Ermäßigungen und Entlastungen vor. Durch diese Anreize sollen die Betriebe dazu bewegt werden, auch freiwillig Energiesparmaßnahmen zu setzen und dadurch einen Beitrag hinsichtlich Energieeinsparungen zu leisten (BMW, 2010, S. 12). Diese Anreize wurden in verschiedenen Gesetzen normiert, welche nachfolgend erläutert werden.

Erneuerbare-Energien-Gesetz (EEG):

In Deutschland existiert eine sogenannte EEG-Umlage, mit welcher der Ausbau von Anlagen zur Produktion von erneuerbaren Energien – wie Wasser- oder Windenergie – finanziert werden soll. Betreiber dieser Anlagen, welche Strom in das öffentliche Netz einspeisen, erhalten eine festgelegte Vergütung pro Kilowattstunde von den jeweiligen Übertragungsnetzbetreibern. Diese verkaufen wiederum den Strom an der Strombörse. Aufgrund von schwankenden Strompreisen kann der Preis, welcher an der Börse erzielt wird, unter den gesetzlich festgelegten Vergütungssätzen liegen. Diese Differenz wird den Übertragungsnetzbetreiber in Form der EEG-Umlage von den Endverbrauchern erstattet (Bundesnetzagentur, 2021a). Diese EEG-Umlage kann unter gewissen Voraussetzungen iSd § 64 EEG reduziert werden, wenn im Unternehmen ein nach ISO 50001 oder EMAS zertifiziertes EnMS eingesetzt wird.

Durch das im Jahr 2019 in Kraft getretene Energiedatensammelgesetz (EnSaG) wurde das EEG novelliert und die sogenannte Drittmengenabgrenzung für EEG-umlagerereduzierte Unternehmen eingeführt. Diese Regelung sieht nach § 62 b EEG vor, dass die an Dritte weitergegebenen Strommengen mithilfe eines Messkonzeptes eichrechtskonform zu ermitteln sind. Somit wird verhindert, dass diese Dritten einen

EEG-umlagereduzierten, kostengünstigen Strom in Anspruch nehmen, welcher eigentlich nur dem Hauptverbraucher aufgrund der energiewirtschaftlichen Privilegierungen zusteht. Dritte sind hierbei u. a. Betreiber von Getränkeautomaten, Reinigungsfirmen oder MieterInnen von Büroräumen des Unternehmens. Eine Verwendung eines EnMS ist in diesem Gesetzestext nicht explizit gefordert, allerdings muss ein adäquates Messkonzept erstellt werden. Da im Rahmen der Einführung eines EnMS ebenfalls ein Messkonzept benötigt wird, kann hierbei eine Basis für die zukünftige Energieeffizienz geschaffen sowie die Einhaltung der gesetzlichen Forderungen des EEG erleichtert werden.

Stromsteuergesetz (StromStG):

Nach § 10 Abs 1 des StromStG kann die Stromsteuer auf Antrag reduziert werden, wenn der Strom durch Unternehmen des produzierenden Gewerbes zu betrieblichen Zwecken entnommen wird. Das Unternehmen muss auch hier iSd § 10 Abs 3 einen Nachweis erbringen, dass in der Organisation ein zertifiziertes EnMS, ein Umweltmanagementsystem oder ein alternatives System zur Verbesserung der Energieeffizienz eingesetzt wird.

Energiesteuergesetz (EnergieStG):

In diesem Gesetz wird die Steuer auf Energieerzeugnisse, exklusive Strom geregelt. Auch hier kann die Energiesteuer nach § 55 EnergieStG auf Antrag reduziert werden, wenn ein zertifiziertes Managementsystem im Unternehmen eingesetzt wird.

2.3 Ökonomische, ökologische und soziale Motive

Neben der Erfüllung von gesetzlichen Normen sowie der Nutzung der steuerrechtlichen Vorteile in Deutschland hat die Einführung eines EnMS im Unternehmen weitere Motive. Durch die Erfassung und Auswertung der Energiedaten werden die Energieeinsparpotenziale im Unternehmen aufgedeckt und können durch gezielte Maßnahmen ausgenutzt werden. Dabei reichen oftmals schon einfache Effizienzmaßnahmen, wie die Bewusstseinsbildung im Unternehmen aus, um eine deutliche Reduktion der Energiekosten zu erreichen (Geilhausen et al., 2015, S. 7; Schmitt & Günther, 2014, S. 13–14). Studien aus Dänemark haben gezeigt, dass durch Implementierung eines EnMS in den Betrieben der Energieverbrauch zwischen 10 und 40 Prozent gesenkt werden konnte (Danish Energy Agency, 2014, S. 4–13). Jedenfalls gilt, dass bei energieintensiven Unternehmen das EnMS stark fokussiert werden sollte, da hier

die Energiekosten einen großen Anteil der Gesamtkosten ausmachen und die monetären Einsparpotenziale am größten sind (Schmitt & Günther, 2014, S. 13–14).

Durch ein energieeffizientes Verhalten kommen Unternehmen der ökologischen und sozialen Verantwortung gegenüber der Gesellschaft und der Umwelt nach. Stakeholder fordern heutzutage von den Produktionsfirmen ein nachhaltiges Verhalten. Die Unternehmen sollten aus diesem Grund offen mit den definierten Zielen sowie mit den erreichten Energieeinsparungen umgehen. Dies kann ein nachhaltiges und verantwortungsbewusstes Image aufbauen, die Geschäftsbeziehung stärken bzw. zur Akquirierung von NeukundInnen beitragen. Dabei stellt die Veröffentlichung der Energieberichte auf der Homepage einen geeigneten Kommunikationsweg dar. Ferner wird in einigen Branchen im Rahmen der LieferantInnenbewertung ein Zertifikat des EnMS der potenziellen LieferantInnen gefordert. Da der Ausstoß von Treibhausgasen die Erderwärmung und den damit in Verbindung stehenden Klimawandel verursachen, können durch eine effiziente Nutzung der energetischen Ressourcen die CO₂-Emissionen reduziert und ein wesentlicher Beitrag zum Schutz der Umwelt geleistet werden (Dehli, 2020, S. 5; Petermann, 2018, S. 193; Schmitt & Günther, 2014, S. 16).

2.4 ISO 50001

Die ISO 50001 ist eine branchenunabhängige, internationale Norm der International Organization for Standardization und stellt den Aufbau eines Energiemanagementsystems in einem Unternehmen dar. Sie beschreibt die Systeme und Prozesse zur Erhöhung der Energieeffizienz und soll Einsparungen messbar und vergleichbar machen. Die ISO 50001 definiert Rahmenrichtlinien für die Einführung einer Energiepolitik in den Unternehmen und unterstützt bei der Festlegung von Zielen und Leitlinien hinsichtlich der Energiesparmaßnahmen. Des Weiteren beinhaltet die Norm einen kontinuierlichen Verbesserungsprozess – auch PDCA-Zyklus genannt – welcher die Effektivität der gesetzten Maßnahmen laufend überwacht. Ferner soll durch die Anwendung der Norm das Bewusstsein für den ressourcenschonenden Umgang mit Energie im Unternehmen geschaffen werden. Die Norm ist in zehn Abschnitte unterteilt. Die ersten drei Kapitel beschäftigen sich mit formellen Inhalten und definieren den Anwendungsbereich der Norm. Die Anforderungen an ein EnMS iSd PDCA-Zyklus werden in Kapitel 4 bis Kapitel 10 erläutert (Brugger-Gebhardt & Jungblut, 2019, S. 3–5; Geilhausen et al., 2015, S. 14–15).

Aufgrund der Standardisierung gilt die ISO 50001 für alle Branchen und Unternehmensgrößen. Sie wurde 2011 durch das technische Komitee der ISO erarbeitet und unter dem Namen ISO 50001:2011 veröffentlicht. Dadurch wurde die, bis zu diesem Zeitpunkt angewandte europäische Norm für EnMS, DIN EN 16001:2009 mit einer Übergangsfrist bis April 2012 zurückgezogen (Brugger-Gebhardt & Jungblut, 2019, S. 3; Geilhausen et al., 2015, S. 11). Im Jahr 2016 wurde von der ISO die Entscheidung für eine Revision der ISO 50001:2011 getroffen. Im August 2018 wurde die neue ISO 50001:2018 veröffentlicht und anschließend vom Deutschen Institut für Normung (DIN) übersetzt herausgegeben. Diese Novellierung diente einerseits dazu, Weiterentwicklungen und Anpassungen an die neuesten Erkenntnisse im Bereich Managementsysteme einzuarbeiten. Aus diesem Grund wurde die neue Norm nach der High Level Structure aufgebaut und kann demnach besser in strategische Managementprozesse und bestehende Managementsysteme integriert werden. Andererseits erfolgten auch energiemanagementspezifische Änderungen und Ergänzungen. Beispielsweise wird in der ISO 50001:2018 von EDM-Systemen gefordert, dass neben Energiemessungen auch weitere Daten zu erfassen sind, die im Zusammenhang mit den Energiedaten oder den Leistungskennzahlen stehen (Bränzel, Engelmann, Geilhausen & Schulze, 2019, S. 12–13; Dehli, 2020, S. 53).

2.5 Aufbau eines Energiemanagementsystems nach ISO 50001

Bei der Anwendung eines EnMS nach ISO 50001 werden einzelne Prozesse im Unternehmen eingeführt, welche die fortlaufende Verbesserung der Energieeffizienz vorantreiben und sicherstellen. Dieser fortlaufende Verbesserungsprozess (FVP) wird geschaffen, indem die neu eingeführten Prozesse in Form eines PDCA-Zyklus nach William Edwards Deming unterteilt werden. Im PDCA-Zyklus existieren Prozesse, welche sich mit der strategischen und operativen Planung des Energiemanagements beschäftigen. Ferner gibt es Prozesse, die sich mit der Umsetzung der Anforderungen auseinandersetzen sowie Prozesse, die eine Messung der Erfolge oder Nicht-Erfolge aufzeigen können. Schlussendlich werden die Messergebnisse analysiert, bewertet und die Möglichkeiten der Optimierung der energetischen Leistung angeführt. Dadurch wird der Kreislauf der fortlaufenden Verbesserung geschlossen (Brugger-Gebhardt & Jungblut, 2019, S. 5–6; Zabel, 2012, S. 91–92).

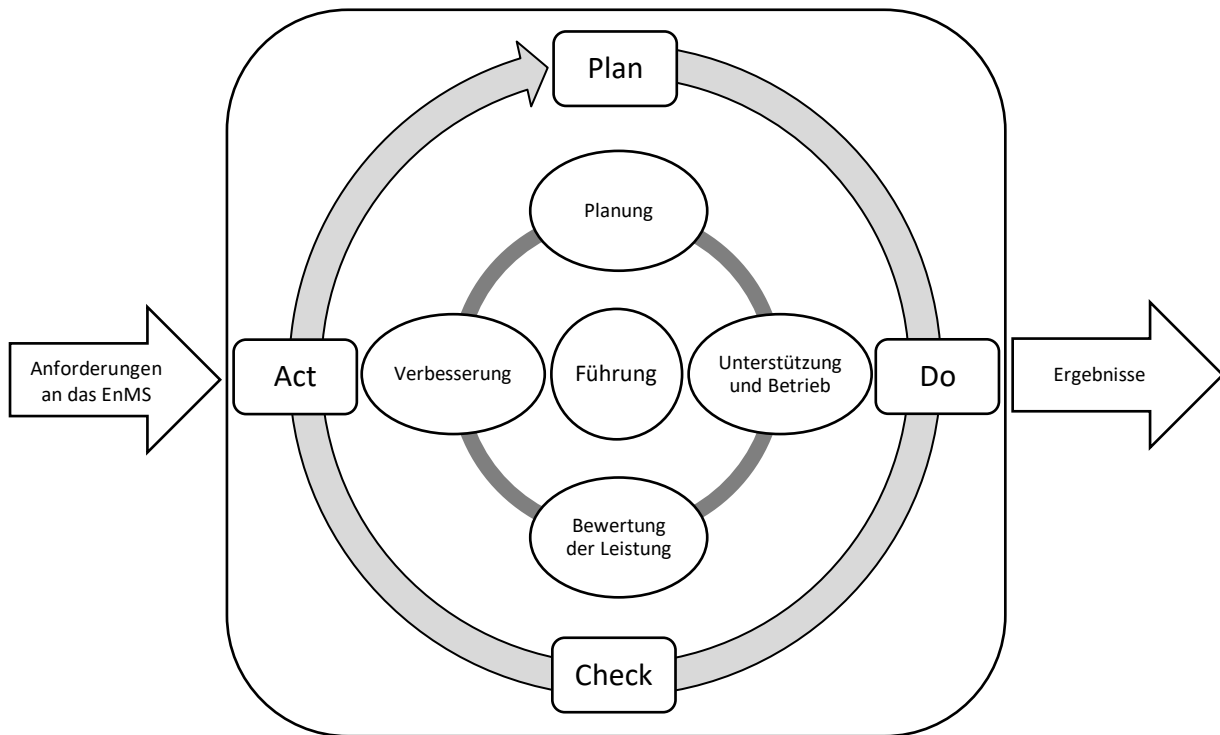


Abbildung 3: Aufbau eines EnMS nach ISO 50001 (In Anlehnung an Bränzel et al., 2019, S. 15)

Abbildung 3 stellt den Aufbau eines EnMS nach der ISO 50001 grafisch dar. Bevor das EnMS eingeführt wird, muss festgelegt werden, welche Anforderungen durch das Managementsystem bedient werden sollen. Dabei wird dieses durch eine Vielzahl an externen und internen Themen beeinflusst. Interne Themen stellen u. a. Kernprozesse, Unternehmensstandorte oder bereits bestehende Managementsysteme dar. Externe Einflüsse entstehen durch interessierte Parteien, auch Stakeholder genannt (Brugger-Gebhardt & Jungblut, 2019, S. 27–29). Dazu zählen demonstrativ folgende Parteien (Bränzel et al., 2019, S. 16):

- KundInnen
- EigentümerInnen
- Gesetzgeber und Behörden
- MitarbeiterInnen

Da es aufgrund sich stetig ändernder Umfeldler in Form neuer Gesetze oder neuer Unternehmensstrategien zu fortwährenden Veränderungen in den interessierten Parteien kommt, ist eine dynamische Liste der Anforderungen vonnöten. Das Unternehmen muss nach ISO 50001 sicherstellen, dass zu jeder Zeit die relevantesten Anforderungen bekannt sind und im EnMS berücksichtigt werden (Bränzel et al., 2019, S. 19).

2.5.1 Führung und Energiepolitik

Die ISO 50001 definiert in Kapitel 5.1 eindeutige Vorgaben für die Aufgaben des Top-Managements und fordert eine Verpflichtung innerhalb des EnMS für die Geschäftsführung und die oberen Managementebenen (ISO, 2018, S. 8). Demnach hat das Management eine Vorbildfunktion inne und beeinflusst ihre MitarbeiterInnen durch ihr Verhalten hinsichtlich des Themas Energie. Dies wird auch als Top-Down-Ansatz bezeichnet, da zunächst die oberste Führungsebene mit der Einführung beginnt und anschließend die unteren Hierarchiestufen der Organisation hinzuzieht und vom neuen System überzeugen möchte. Ein EnMS ist nur erfolgreich, wenn die Unternehmensführung und alle MitarbeiterInnen zu 100 Prozent hinter dem EnMS stehen und es sich dadurch in die Unternehmenskultur integriert. Abgesehen von der Vorbildfunktion und der Vermittlung der Bedeutung des EnMS an die MitarbeiterInnen ist das Top-Management nach ISO 50001 verantwortlich, dass beispielsweise der Anwendungsbereich des EnMS definiert ist, Rollen und Verantwortlichkeiten festgelegt sind und die erforderlichen monetären und nicht-monetären Ressourcen zur Verfügung stehen (Brugger-Gebhardt & Jungblut, 2019, S. 45–47; Geilhausen, 2020, S. 13–14).

Bei der sogenannten Energiepolitik legt das Management die strategische Ausrichtung des EnMS fest. Die Energiepolitik kann als eigenes Unternehmensleitbild für das Thema Energie gesehen werden und gehört demnach auch zur Unternehmenskultur. Im Rahmen der Energiepolitik wird nach ISO 50001 schriftlich festgehalten, in welcher Form mit der Ressource Energie im Unternehmen umgegangen werden soll. Zudem werden die Strategien zum Energiesparen definiert, sodass im Rahmen der Planung des EnMS Ziele abgeleitet werden können. Ferner muss die Energiepolitik eine Verpflichtung zur Erfüllung rechtlicher und anderer Anforderungen sowie zur fortlaufenden Verbesserung der Energieleistung beinhalten. Eine Energiepolitik muss des Weiteren im Unternehmen transparent und verständlich kommuniziert werden, sodass die MitarbeiterInnen diese jederzeit einsehen und verinnerlichen können (Brugger-Gebhardt & Jungblut, 2019, S. 49–50; Geilhausen, 2015, S. 13; Posch, 2011, S. 175–176).

2.5.2 Planung

Die ISO 50001 definiert in Kapitel 6, dass im ersten Schritt des FVP die Energieplanung durchgeführt werden muss. Bei der Durchführung dieses Planungsprozesses werden, aufbauend auf die festgelegte Energiepolitik, vorhandene Energiedaten gesammelt und analysiert, da strategische und operative Ziele nur dann formuliert werden können, wenn

eine Ist-Erhebung des derzeitigen Standes durchgeführt wurde. Dieser wesentliche Schritt der ISO 50001 wird auch als energetische Bewertung bezeichnet. Hierbei werden die kritischen Bereiche, Prozesse und Anlagen im Unternehmen identifiziert, welche einen wesentlichen Einfluss auf den Energieverbrauch und die Energieeffizienz aufweisen (Bränzel et al., 2019, S. 30–31). Dies soll durch ein kurzes Beispiel veranschaulicht werden. Beträgt der Verbrauch von Strom für die Lampen der Büros beispielsweise 4 MWh pro Jahr, verbrauchen allerdings die Kältemaschinen zur Kühlung des Gebäudes 100 MWh pro Jahr, dann sollten im ersten Schritt Maßnahmen geplant werden, um den Verbrauch der Kältemaschinen zu optimieren. Sind diese Potenziale ausgeschöpft, werden nach Priorität die weiteren Verbraucher optimiert (Brugger-Gebhardt & Jungblut, 2019, S. 72–73). Zur Identifikation dieser sogenannten SEUs (Significant Energy Uses) können Instrumente wie die ABC-Analyse, das Pareto-Prinzip oder Sankey-Diagramme angewendet werden (Bränzel et al., 2019, S. 39–41). Für die SEUs werden im Rahmen der Planung die Chancen und Risiken bestimmt, auf welche im weiteren Verlauf des Prozesses aufgesetzt wird (Bränzel et al., 2019, S. 30–31).

Im Anschluss an die energetische Bewertung werden Energieleistungskennzahlen (EnPI = energy performance indicator) definiert (ISO, 2018, S. 11). Hierbei dienen die EnPIs der besseren Darstellung und Analyse der Entwicklungen des Energieeinsatzes bei Vergleichen von Datensätzen, da Bezugsgrößen im Gegensatz zu absoluten Werten eine höhere Aussagekraft besitzen. Beispielsweise ist eine Reduzierung des Strombedarfs für die Produktionsanlagen in einem bestimmten Zeitraum um x Prozent zu einem anderen Zeitraum wenig aussagekräftig, wenn gleichzeitig eine Senkung des Durchsatzes bei der Produktion stattgefunden hat. Daher müssen entsprechende Kennzahlen entwickelt werden, um die Leistung in einem dynamischen Umfeld (Wetterbedingungen, Produktionsumfang, Anzahl der Mitarbeiter etc.) adäquat bewerten zu können (Brugger-Gebhardt & Jungblut, 2019, S. 76–78).

Neben den EnPI werden im Rahmen des Planungsprozesses Basiskennzahlen – auch energetische Ausgangsbasen (EnB = energy baseline) genannt – festgelegt, welche als Referenzgröße dienen (ISO, 2018, S. 12). Dadurch können positive oder negative Entwicklungen im Vergleich zum Ausgangswert aufgezeigt werden. Der Zeitraum für die EnBs sollte möglichst repräsentativ sein und daher nicht zu lang bzw. zu kurz gewählt werden. Ein Zeitbereich von einem Jahr stellt dabei eine gute Referenz dar, um saisonale Unterschiede einbeziehen zu können (Brugger-Gebhardt & Jungblut, 2019, S. 84; Geilhausen, 2020, S. 30–31).

Im letzten Schritt der Planung werden nach ISO 50001 konkrete Energieziele und Maßnahmen in Form von Aktionsplänen definiert. Weiters müssen Maßnahmen festgelegt werden, wie die Energiedaten gesammelt und analysiert werden sollen. Dafür muss das Unternehmen determinieren, welche technischen Einrichtungen als Datenerfassungsmethode zum Einsatz kommen sollen. Ferner wird die Häufigkeit der Messvorgänge festgelegt und adäquate Softwaresysteme eruiert, welche die Archivierung, Überwachung und Analyse der Energiedaten ermöglichen. Beim Beschließen der jeweiligen Maßnahmen ist vor allem die Wirtschaftlichkeitsbetrachtung ein wichtiger Aspekt, welcher berücksichtigt werden sollte. Dabei muss beachtet werden, dass der Aufwand für die jeweiligen Maßnahmen im Vergleich zum Nutzen in einem angemessenen Verhältnis steht. Es ist verständlicherweise nicht erwünscht, dass ein hoher Aufwand für einen minimalen Effekt hinsichtlich der Energieeinsparungen betrieben wird (Brugger-Gebhardt & Jungblut, 2019, S. 61; Geilhausen, 2020, S. 31–34).

2.5.3 Unterstützung und Betrieb

In der ISO 50001:2011 wurde der Umsetzungsprozess (Do) des PDCA-Zyklus in Kapitel 4.5 beschrieben (ISO, 2015, S. 15–17). Dies wurde in der ISO 50001:2018 abgeändert und in die Abschnitte Unterstützung und Betrieb aufgeteilt. Dabei sollen im Unterstützungsprozess die notwendigen Ressourcen in Form von Personal, Arbeitszeit oder technischen Messeinrichtungen zur richtigen Zeit bereitgestellt werden. Zudem ist es für das Unternehmen notwendig, dass für die Umsetzung der Energiesparmaßnahmen das Personal ausreichend qualifiziert ist. Durch gezielte interne oder externe Aus- und Weiterbildungen soll den MitarbeiterInnen Fachwissen vermittelt werden. Des Weiteren ist es notwendig, das Personal regelmäßig durch Workshops, Informationsveranstaltungen oder Newslettern zu sensibilisieren und die Bedeutung ihrer Tätigkeit innerhalb des EnMS zu kommunizieren (Geilhausen, 2020, S. 35–39; Jovanović, Filipović & Bakić, 2017, S. 19).

Ein weiterer – von der ISO 50001 geforderter – Aspekt im Hilfsprozess ist das regelmäßige Dokumentieren von wesentlichen Informationen im Hinblick auf das EnMS. Dabei dienen die aufgezeichneten Informationen für den Nachweis der Prozessqualität, der Einhaltung der Norm als auch der Messergebnisse. Zudem muss die Wirksamkeit des Managementsystems sowie die fortlaufende Verbesserung aufgezeichnet werden. Es gibt Informationen, deren Aufzeichnung von der ISO 50001 zwingend gefordert ist. Demonstrativ zählen zu diesen Informationen Nachweise über die Kompetenz von MitarbeiterInnen, Nachweise der Ergebnisse aus Überwachung und Messung und der

Reaktion auf wesentliche Abweichungen der energiebezogenen Leistung. Demgegenüber existieren Informationen, die sinnvoll für das Steuern des gesamten Prozesses sind, deren Aufzeichnung durch die Norm jedoch nicht gefordert ist. Dabei entscheidet das Unternehmen selbst, welche Dokumente als sinnvoll erachtet werden. Dies können u. a. Dokumente wie Verfahrensanweisungen für die MitarbeiterInnen oder Messpläne sein, welche für die tägliche Arbeit hilfreich sind. Für die Dokumente muss die Aktualität sowie die dauerhafte Verfügbarkeit, die Möglichkeit zur Identifikation der AutorInnen und die Rückverfolgung von Revisionen sichergestellt werden (Brugger-Gebhardt & Jungblut, 2019, S. 112–116; Geilhausen, 2020, S. 39–41; Jovanović et al., 2017, S. 20).

Der wichtigste Schritt im Umsetzungsprozess ist die Verwirklichung der Energiesparmaßnahmen anhand von Aktionsplänen, um die geplanten Ziele zu erreichen. Es ist auch der technisch, finanziell und zeitlich wichtigste Schritt innerhalb eines EnMS. Die im Maßnahmenplan aufgeführten Änderungen im Unternehmen sollten nach Priorität umgesetzt und – wenn möglich – zunächst pilotartig getestet werden. Dadurch ist es im weiteren Verlauf des PCDA-Zyklus leichter möglich, Fehler bei Strategien aufzuzeigen und durch entsprechende Korrekturen zu eliminieren. Erst im Anschluss ist es sinnvoll, die jeweiligen Maßnahmen auf alle Bereiche des Unternehmens auszuweiten (Dehli, 2020, S. 63–64). In der ISO 50001 wird die gezielte Steuerung der energieintensiven Prozesse als Ablauflenkung bezeichnet. Diese Ablauflenkung wird durch Verfahrensanweisungen und Handbücher erreicht, welche den internen und externen MitarbeiterInnen, DienstleisterInnen und LieferantInnen kommuniziert werden müssen. Des Weiteren fordert die Norm, dass bei Veränderungen an technischen Anlagen, Gebäuden oder der Organisationsstruktur eines Unternehmens, die Auswirkungen auf die Energieeffizienz stets zu berücksichtigen ist. Beispielsweise muss bei der Beschaffung von technischen Anlagen vor der Einleitung des Einkaufes die Energieeffizienz bewertet werden. Diese Bewertung fließt neben den Budgetvorgaben, den Amortisations- und den Lieferzeiten in die Investitionsentscheidung ein. Das Unternehmen muss zudem nach ISO 50001 Kapitel 8.3 im Rahmen der Ausschreibung seine LieferantInnen darauf hinweisen, dass die Bewertung der Energieeffizienz in die Entscheidung der LieferantInnenauswahl einfließt (Geilhausen, 2020, S. 44–46).

2.5.4 Bewertung der Leistung

Für den Erfolg des EnMS ist es von Bedeutung, dass die energetischen Kennzahlen und Energiedaten fortlaufend überwacht und analysiert werden. Dies lässt sich mithilfe von Kontrollmaßnahmen im Rahmen des EDM durch Messung und Überwachung erreichen.

Auf Basis der in der Planung festgelegten Messverfahren und Erfassungskonzepte werden nach Kapitel 9 der ISO 50001 Analysen zur Entwicklung der EnPIs im Vergleich zu den EnBs sowie der Wirksamkeit der Aktionspläne gefordert. Kommt es zu größeren Abweichungen von den Zielvorgaben, müssen die Messungen und Berechnungen entsprechend auf Plausibilität überprüft werden, um methodische Fehler ausschließen zu können. Systematische Fehler sind bereits in der Messplanung zu beachten. Zudem können Fehler aufgrund von zufälligen Ereignissen, wie beispielsweise Stromausfällen, entstehen. Hierbei kann nur individuell durch Mittelwertbildungen vorgegangen werden, um die Fehler zu bereinigen und die Datenintegrität zu wahren. Die bereinigten Werte sind jedenfalls zu kennzeichnen, damit eine Rückverfolgbarkeit der Korrektur ermöglicht wird. Können systematische und zufällige Fehler ausgeschlossen werden, gilt es, die Ursache für die negative oder auch positive Entwicklung in den Daten zu finden. Die Tätigkeiten im Rahmen des EDM müssen gemäß Normanforderung entsprechend dokumentiert werden. Dabei sind die Ergebnisse der Messungen und Analysen durch adäquate Softwareprodukte aufzuzeichnen und zu archivieren (Bränzel et al., 2019, S. 73; Geilhausen, 2020, S. 47–48).

Nach der ISO 50001 sind im Rahmen des EnMS nicht nur Energiedaten zu überwachen und zu analysieren, sondern auch das EnMS selbst muss regelmäßig intern auditiert werden. Dieses Audit ist das wesentlichste Instrument, um die adäquate Funktionsweise der Prozesse des Managementsystems zu validieren und zu verbessern (Brugger-Gebhardt & Jungblut, 2019, S. 146–147; Dehli, 2020, S. 69). Laut Normanforderungen müssen dabei folgende Punkte überprüft werden (Geilhausen, 2020, S. 48):

- Konformität mit den Anforderungen der ISO 50001 und der Systemdokumentation,
- Übereinstimmung der Ausprägung des Systems mit den definierten Zielen,
- Wirksamkeit des EnMS und die fortlaufende Verbesserung der energetischen Leistung.

Zudem fordert die Norm in Kapitel 9.2.2 ein konkretes Auditprogramm, welches auf Basis der Energierelevanz und – falls vorhanden – der Ergebnisse der letzten Audits zu erstellen ist. In diesem Auditprogramm werden die AuditorInnen, Umfang, Termine und Dauer des Audits festgelegt. Bereiche mit direktem Bezug zum EnMS bzw. mit negativen Ergebnissen in der Vergangenheit werden in kürzeren Intervallen sowie intensiver auditiert. Im Gegensatz dazu sind die Intervalle in Bereichen ohne direkten Bezug zum EnMS bzw. mit positiven Auditergebnissen länger und das Audit selbst weniger

umfangreich. Nach Abschluss des Audits muss nach ISO 50001 ein Bericht an die zuständigen Führungskräfte übermittelt werden. In diesem werden die festgestellten Ergebnisse zusammengefasst dargestellt. Daraus entsteht eine Liste mit Verbesserungspotenzialen oder Aufgaben, wenn Anforderungen nicht erfüllt werden (Brugger-Gebhardt & Jungblut, 2019, S. 145–146; Geilhausen, 2020, S. 48–50).

Zum Abschluss des Check-Abschnittes im PDCA-Zyklus erfolgt ein Management-Review durch die Unternehmensleitung. Dabei werden die Ergebnisse des internen Audits zusammengefasst, der Energieverbrauch des Betriebs im abgelaufenen Geschäftsjahr dargelegt bzw. bewertet sowie der Stand der Korrekturmaßnahmen aufgezeigt (Dehli, 2020, S. 69–70).

2.5.5 Verbesserung

Der vierte und letzte Schritt des Deming-Kreislaufes dient der Erarbeitung von Korrekturmaßnahmen auf Grundlage der festgestellten Abweichungen aus Schritt 3 und einer dahingehenden Optimierung des EnMS (Jovanović et al., 2017, S. 24). Zuerst erfolgt im Verbesserungsprozess die Fehlerbehebung, bei welcher das Unternehmen bestrebt ist, den Missstand zu beheben und die energetischen Ressourcen zu schonen. Die wesentlichste Tätigkeit bei der Optimierung ist die Ursachenermittlung, um die Hintergründe der Fehler zu beleuchten und das Wiederauftreten des Fehlers zu verhindern. Für die Ursachenermittlung kann u. a. auf Methoden wie Brainstorming zurückgegriffen werden. Aus den Ergebnissen der Ursachenermittlung werden Korrekturmaßnahmen abgeleitet. Ist beispielsweise die Ursache für den zu hohen Energiebedarf der Lüftungen auf falsches Regelverhalten zurückzuführen, dann wäre eine demonstrative Korrekturmaßnahme die Anpassung der Parameter von den PI- oder PID-Reglern. Bei den Korrekturmaßnahmen wird zwischen Mussmaßnahmen und Sollmaßnahmen unterschieden. Mussmaßnahmen sind jeweils zügig umzusetzen, da hier ein umgehender Handlungsbedarf besteht, damit der Fehler nicht ein weiteres Mal in kurzer Zeit auftritt. Wohingegen Sollmaßnahmen Empfehlungen im Sinne von Verbesserungsvorschlägen darstellen. Für die Korrekturmaßnahmen ist daher eine Kosten-Nutzen-Analyse durchzuführen, um das Verhältnis zwischen dem Aufwand der Risikominimierung und der Fehlerhäufigkeit sowie Fehlerauswirkung zu berücksichtigen. Wird eine Maßnahme als sinnvoll erachtet, fließt sie in den nächsten Planungsprozess des PDCA-Zyklus ein und wird im Anschluss umgesetzt. Um schlussendlich einen FVP zu erreichen ist es unabdingbar, eine stetige Wirksamkeitsprüfung der umgesetzten Maßnahmen durchzuführen, um eine dauerhafte

Verbesserung in der energetischen Leistung zu erreichen. Werden die geänderten Strategien und Prozesse als erfolgreich erachtet, sind diese in die Dokumentationen und Verfahrensanweisungen aufzunehmen (Bränzel et al., 2019, S. 78–80; Dehli, 2020, S. 70–71).

Neben der Handhabung von bereits identifizierten Abweichungen, ist auch eine vorausschauende Betrachtung hinsichtlich potenzieller Fehlerquellen essenziell. Dadurch werden mögliche Fehlerursachen frühzeitig analysiert und beseitigt. Potenzielle Fehlerquellen können dabei mittels Fehlermöglichkeits- und Einflussanalysen (FMEA) bewertet werden. Bei dieser Analyse wird im Rahmen eines interdisziplinären Teams jedem potenziellen Fehler eine Risikoprioritätszahl (RPZ) zugeordnet. Die RPZ berechnet sich auf Basis diverser Parameter. Es fließen die Auswirkung des Fehlers, die Wahrscheinlichkeit des Auftretens und die Wahrscheinlichkeit, dass der Fehler innerhalb des Unternehmens entdeckt wird, in die Berechnung ein. Jeder Fehler wird anschließend gewichtet und mit Maßnahmen belegt, welche zur Senkung der RPZ beitragen sollen (Geilhausen et al., 2015, S. 63).

Gemäß ISO 50001 ist ein Nachweis der fortlaufenden Verbesserung in der energiebezogenen Leistung zu führen. Ein adäquates Mittel im EnMS zur Dokumentation von Verbesserungen ist der Energiebericht. Im Energiebericht werden die Informationen hinsichtlich der erreichten Verbesserungen zusammengetragen und durch eine Reduzierung des Energieverbrauchs in Form der EnPIs belegt. Für den vollständigen Nachweis des FVP zur Erfüllung der Normanforderung muss neben der Einsparung von Energie und der dahingehenden Steigerung der Energieeffizienz, die Weiterentwicklung des Managementsystems ebenfalls belegt werden (Brugger-Gebhardt & Jungblut, 2019, S. 161; Geilhausen, 2020, S. 54–55).

3 Anforderungen an eine Energiedatenmanagement-Software

In diesem Kapitel erfolgt zuerst die Erläuterung der Grundlagendefinition sowie der Funktionsweise einer EDM-Software. Anschließend wird ein Requirements Engineering Prozess durchgeführt, um die aktuellen Anforderungen an eine EDM-Software adäquat zu ermitteln.

3.1 Energiedatenmanagement-Software

Damit das Top-Management sowie die AuditorInnen die Energieeffizienz und die Wirksamkeit des EnMS bewerten können, müssen die Energieverbräuche des Unternehmens quantifizierbar gemacht werden (Bränzel et al., 2019, S. 328). Die Erhebung der Energiedaten, die Berechnung der daraus resultierenden Energieverbräuche sowie die Erstellung von Trendkurven oder Balkendiagrammen zur Auswertung erfolgt in vielen Unternehmen noch immer buchstäblich zu Fuß. Dabei werden die Zählerstände durch die MitarbeiterInnen abgelesen und typischerweise mit Tabellenkalkulationsprogrammen ausgewertet. Grenzwertüberwachungen, um Lastspitzen und daraus resultierende hohe Stromkosten zu vermeiden, können ohne eine entsprechende Software kaum durchgeführt werden. Durch den Einsatz eines EDM kann hingegen die Effizienz des EnMS gesteigert werden. Zeitliche und personelle Kapazitäten werden reduziert, indem das Auslesen der Rohdaten sowie die Berechnung und Visualisierung der Verbräuche teil- bzw. vollautomatisch erfolgt (Kappler, 2014, S. 226). Das EDM ist ein zentrales Instrument zur Gewinnung von quantitativen Informationen über die Energieflüsse im Unternehmen. Es ist hierbei für die kontinuierliche Erfassung, Archivierung, Überwachung und grafische Aufbereitung der Daten verantwortlich. Die einzelnen Prozesskomponenten werden in Abbildung 4 abgebildet (Geilhausen et al., 2015, S. 287; Mudaliar & Sivakumar, 2020, S. 1; Schweinfurth, 2020, S. 232; Trentler, 2013, S. 273).

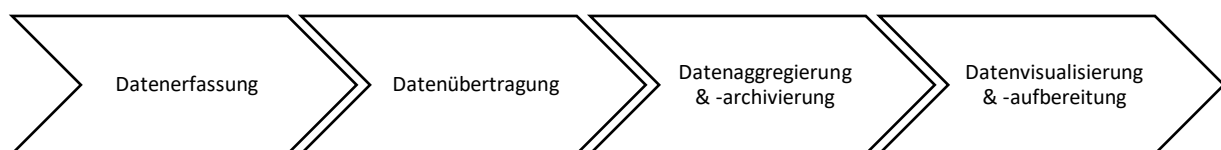


Abbildung 4: Prozesskomponenten des EDM (In Anlehnung an Schweinfurth, 2020, S. 232)

Auf Feldebene werden die Messungen mittels Energiezählern durchgeführt. Je nach Anforderung existieren Zähler für die unterschiedlichsten Medien wie Strom, Druckluft, Gas, Wasser sowie Wärme- und Kältemenge (Kappler, 2014, S. 226–228). Des Weiteren unterscheiden sich die Zähler auch anhand der Technologien. Einfache Zähler liefern einen digitalen 24 Volt Impuls, sobald ein Verbrauch festgestellt wird. Wohingegen moderne Multifunktionsmessgeräte über Displays verfügen und damit detaillierteste Informationen über die Erfassungsstationen liefern sowie Feldbusse zur automatischen Datenübertragung nutzen. Abbildung 5 zeigt ein Multifunktionsmessgerät des Herstellers Janitza. Neben dem fortlaufenden Energieverbrauch erfasst dieses Strommessgerät u. a. auch die aktuellen Blind-, Schein- und Wirkleistungen sowie Spannungen, Stromstärken und Frequenzen der einzelnen Phasen.



Abbildung 5: Multifunktionsmessgerät UMG 96 RM-E (Janitza, o. D.)

Weiters ist zu beachten, dass aufgrund der Novellierung der ISO 50001:2018 im Rahmen eines EDM auch weitere Daten für die Generierung von EnPI zu erfassen sind. Beispielsweise die Außentemperatur mittels PT1000 Sensoren oder die produzierte Stückzahl mittels Impulsgeber (Bränzel et al., 2019, S. 309).

Nach der Datenerfassung müssen die Daten für die weitere Verwendung an die übergeordnete EDM-Software übermittelt werden. Dies geschieht über diverse Schnittstellen zu den Zählern. Energiezähler verwenden hierbei häufig die Feldbusse KNX/EIB, Modbus, M-Bus oder BACnet (Masek et al., 2018, S. 32; Schellong, 2016, S. 217; Schödwell, Drenkelfort, Erek, Zarnekow & Behrendt, 2012, S. 192). Verfügt die EDM-Software über keine direkte Schnittstelle zu dem jeweiligen Feldbus, müssen die Protokolle mithilfe von Gateways übersetzt werden. Sind die Energiedaten in der Software angekommen, werden die Verbräuche berechnet sowie die Datenarchivierung

vorgenommen. Die Software und die dazugehörige Datenbank können entweder auf einem lokalen Server installiert oder in Form eines Cloud-Computings von einem externen Anbieter gehostet werden (Kappler, 2014, S. 229–233).

Wurden die Daten von der Software entsprechend aufbereitet, werden diese den NutzerInnen über eine grafische Oberfläche bereitgestellt. Die aggregierten und archivierten Messdaten sind nur dann nützlich, wenn sie in entscheidungsrelevante Informationen umgewandelt werden. Aus diesem Grund ist der letzte Prozessschritt im EDM die Datenaufbereitung und -visualisierung, um sinnvolle Informationen aus dem vorhandenen Datensatz extrahieren zu können. Dieser Schritt wird auch als Energiemonitoring bezeichnet (Bränzel et al., 2019, S. 311). Das Energiemonitoring beinhaltet vor allem folgende Elemente (Geilhausen et al., 2015, S. 301; Schellong, 2016, S. 231):

- Visuelle Darstellung und Analyse von Lastgängen des Energieverbrauchs
- Vergleich und Korrelation von Datenreihen (Verbräuche, EnPI)
- Plausibilitätsprüfung
- Darstellung der Energiekosten
- Alarmgenerierung zur Lastspitzenvermeidung

Je nach Softwarelösung stehen für den Zugriff auf die Energiedaten Desktopanwendungen, Webservices oder mobile Anwendungen für Tablets und Smartphones zur Verfügung. Mit diesen Anwendungen können die Daten in den unterschiedlichsten grafischen oder tabellarischen Formen dargestellt werden (Kappler, 2014, S. 229–233; Schödwell et al., 2012, S. 192; Schweinfurth, 2020, S. 233; Trentler, 2013, S. 273). In Abbildung 6 wird demonstrativ ein Stromlastgang eines Verbrauchers dargestellt, welcher mithilfe einer EDM-Software erstellt wurde.

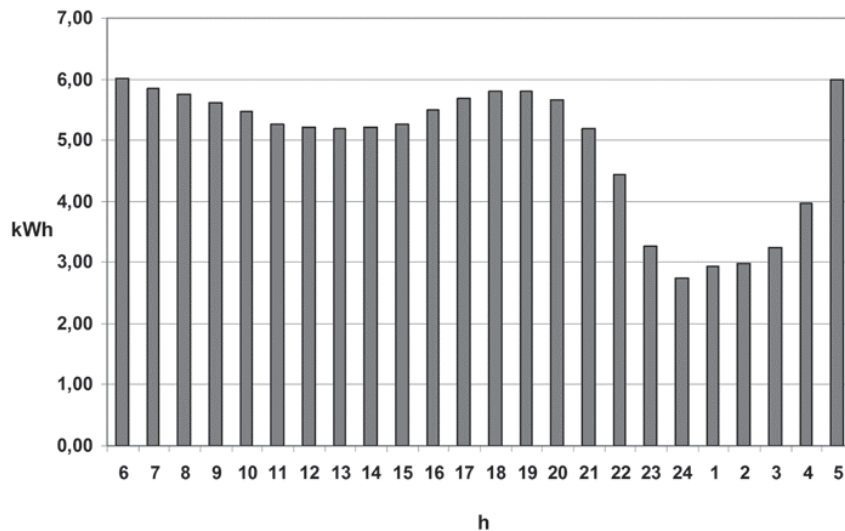


Abbildung 6: Beispiel eines Stromlastgangs (Schellong, 2016, S. 395)

Somit besteht ein EDM einerseits aus der Hardware in Form von Zähl- und Messeinrichtungen sowie Gateways zur Feldbuskonvertierung und andererseits aus einer EDM-Software, welche die Energiedaten von den Messeinrichtungen abfragt, im Anschluss verarbeitet und in der Datenbank speichert sowie den AnwenderInnen über ein Graphical User Interface (GUI) für Analysen zur Verfügung stellt. EDM-Softwares werden in der Regel als Supervisory Control and Data Acquisition (SCADA)-System realisiert und kommunizieren mit Speicherprogrammierbare Steuerungen (SPS), welche hardwarebasiert oder auch als Soft-SPS ausgeführt sein können (Golovatchev, 2020, S. 607; Schellong, 2016, S. 217; Tiedemann, 2021, S. 84). Die Topologie wird nochmals in Abbildung 7 grafisch demonstriert.

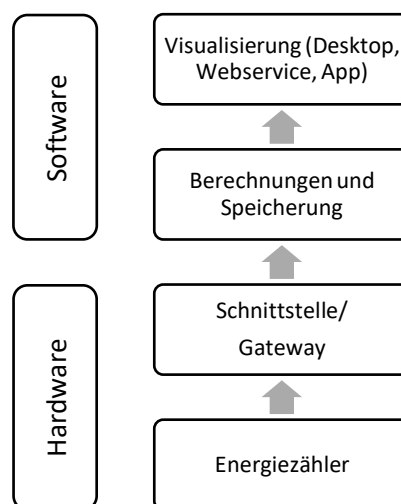


Abbildung 7: Topologie eines EDM-Systems (In Anlehnung an Kappler, 2014, S. 227)

Wie in Kapitel 2.5.4 bereits erwähnt, befindet sich das EDM im Check-Abschnitt des PDCA-Zyklus. Da das EDM somit nur einen Teilbereich der ISO 50001 betrifft, kann eine EDM-Software auch nur für diesen Teilbereich zertifiziert werden. Diese Zertifizierung wird für Österreich und Deutschland von der TÜV Süd AG durchgeführt und nennt sich „Zertifiziertes Energiedatenmanagement“. Dabei legt TÜV Süd Mindestkriterien für Hardwaregeräte, wie z. B. Messeinrichtungen, Datenlogger und SPS, zur Datenerfassung und Mindestkriterien für Softwaresysteme zur Auswertung und Darstellung energierelevanter Daten fest. Dadurch wird festgestellt, dass die gesetzlichen Anforderungen für die Darstellung, Analyse und Dokumentation erfüllt werden. Bei einer erfolgreichen Zertifizierung werden die Anforderungen folgender Normen erfüllt (TÜV, 2021):

- ISO 50001 – Energiemanagementsysteme
- DIN EN 16247 – Energieaudits

3.2 Anforderungsidentifizierung

Im ersten Schritt des Requirements Engineerings wird eine Anforderungsermittlung durchgeführt. Dabei existieren drei Arten von Anforderungsquellen. Es können die Anforderungen von den Stakeholdern über direkte, persönliche Kommunikation erhoben werden. Ferner enthalten verschiedenste Dokumente Informationen, aus denen Anforderungen gewonnen werden können. Diese Dokumente sind u. a. Gesetzestexte oder Normen. Als dritte Quelle dienen sogenannte „Systeme in Betrieb“. Dies sind Alt- bzw. Vorgänger- sowie Konkurrenzsysteme, welche von Personen analysiert und daraus Anforderungen abgeleitet sowie dokumentiert werden. Im Requirements Engineering wird neben den möglichen Anforderungsquellen zwischen drei Arten von Anforderungen unterschieden. Bei den funktionalen Anforderungen handelt es sich um bereitzustellende Funktionen oder Services, welche die Software beherrschen sollte. Qualitätsanforderungen definieren gewünschte Qualitäten der Software. Dies sind typischerweise Anforderungen hinsichtlich der Performance, Zuverlässigkeit, Skalierbarkeit oder Portabilität. Die dritte Anforderungsart sind sogenannte Rahmenbedingungen. Dies sind organisatorische, gesetzliche oder technische Vorgaben, die vom Hersteller nicht aktiv beeinflusst werden können und welche die Umsetzungsmöglichkeiten der Software einschränken (Balzert, 2009, S. 455–456; Pohl & Rupp, 2015, S. 21).

Im Rahmen dieser Masterarbeit werden zur Ermittlung der Anforderungen an eine EDM-Software alle drei erwähnten Quellen verwendet. Zur Identifizierung der rechtlichen Anforderungen wird eine Literaturrecherche durchgeführt und auf die Dokumente der Gesetzgeber und der Normeninstitutionen zurückgegriffen. Für die Marktanforderungen werden Informationen der Stakeholder sowie die Quelle der „Systeme in Betrieb“ verwendet. Hinsichtlich der „Systeme in Betrieb“ wird der Marktspiegel der Energieagentur Nordrhein-Westfalen herangezogen, bei welchem die Features der Konkurrenzsysteme der evon GmbH erhoben wurden. Die Zuordnung der identifizierten Anforderungen zu den jeweiligen Anforderungsquellen und -arten erfolgt in Kapitel 3.3.

3.2.1 Anforderungen nach ISO 50001

Für die Erhebung der Anforderungen nach der ISO 50001 wurde eine Literaturrecherche durchgeführt. Dabei wurde vom Autor dieser Masterarbeit festgestellt, dass in der ISO 50001 ausschließlich allgemein erläutert wird, welche Prozesse im Unternehmen für ein EnMS vorhanden sein, aber nicht, wie diese exakt umgesetzt werden müssen. Nach Kapitel 9.1.1 der ISO 50001 muss die Organisation für ein der Norm entsprechendes EnMS sicherstellen, dass ihre Haupttätigkeiten, welche die energiebezogene Leistung bestimmen und beeinflussen, in definierten Zeitabständen validiert werden. Folgende Aspekte sind bei diesem Monitoring mindestens zu berücksichtigten (ISO, 2018, S. 15–16):

- a) „die wesentlichen Energieeinsatzbereiche und weitere Ergebnisse der energetischen Bewertung
- b) die relevanten Variablen der wesentlichen Energieeinsatzbereiche
- c) EnPIs
- d) Wirksamkeit der Aktionspläne hinsichtlich der Erreichung strategischer und operativer Ziele
- e) Bewertung des aktuellen Energieverbrauchs gegenüber dem erwarteten Energieverbrauch“

Es liegt im Entscheidungsbereich der Organisation, die Vorgehensweise hinsichtlich der Messungen zu bestimmen. Einerseits besteht die Möglichkeit, die Energiedaten entweder von reinen Elektrizitätszählern zu messen und die Analysen sowie Berechnungen ohne Software durchzuführen. Dies ist vor allem für kleine Organisationen zu empfehlen. Andererseits können nach ISO 50001 auch umfangreiche Überwachungs- und Messsysteme mit Softwareanwendungen eingesetzt werden. Diese Softwaresysteme

müssen dabei in der Lage sein, die Energiedaten automatisch zu erfassen. Ferner muss die Organisation sicherstellen, dass die verwendete Software fehlerfreie und reproduzierbare Daten liefert. Zur Reproduzierbarkeit der Daten müssen Aufzeichnungen geführt werden (ISO, 2018, S. 15–16).

Wie zu erkennen ist, sind diese Anforderungen an eine EDM-Software nur rudimentär in der ISO 50001 ausgeführt. Aus diesem Grund stellen die Zertifizierungsstellen, welche eine EDM-Software hinsichtlich eines „Zertifizierten Energiemanagementsystems“ auditieren, Kriterienkataloge zur Verfügung. Werden die in den Katalogen festgehaltenen Anforderungen eingehalten, ist die Software ISO 50001 konform (TÜV, 2021). Der aktuelle Kriterienkatalog der TÜV Süd AG wurde in einer persönlichen E-Mail dem Autor übermittelt und ist in Anhang A abgebildet. Nachfolgend werden die Mindestkriterien, welche die TÜV Süd AG für eine ISO 50001 konforme EDM-Software festgelegt hat, tabellarisch dargestellt. Es wurden ausschließlich die Anforderungen aus dem Kriterienkatalog entnommen, welche Software- und nicht Hardwaresysteme betreffen, da das EMSControl als reine EDM-Software anzusehen ist.

Kriterium	Mindestanforderungen
Standardisierte Bussysteme oder Protokoll	5 Systeme
Export- und Importfunktionen (SAP, Datenbanken, Excel etc.)	Export in Office-Programme
Kopplung an andere Datenbanken	Werte partiell lesbar
Fernzugriff auf Software	Zugriff über Internet/Intranet
Warnfunktion für Mess- oder Grenzwerte	Ausgabemedium E-Mail, SMS, Logdatei etc.
Nutzerrechtevergabe, Administration	Lese- und Schreibrechte
Prüfung der Messdaten auf Konsistenz und Plausibilität	Plausibilitätsprüfung wird durchgeführt (manuell oder automatisch)
Funktionsweise der Datenerfassung	Manuell und automatisch
Verwaltung der Zähler / Wechsel der Zähler	Verwaltung der Zählernummern ist möglich
Virtuelle Zählstellen	Definition virtueller Zählstellen ist möglich
Kostenstellenzuordnung	Zuordnung der Zählstellen auf Kostenstellen möglich
Verteilschlüssel je Zählstelle oder Kostenstelle	Prozentuale Verteilung einer Zählstelle auf Kostenstellen
Abbildung von Gebäuden und Liegenschaften	Baum-Darstellung
Witterungsbereinigung	Gradtagzahlen
Verwaltung der Abrechnungsdaten des/r Versorger(s)	Verwaltung der Daten für mehrere Versorger
Verwaltung von Abrechnungsdaten gegenüber Sub-Abnehmern	möglich
Datums- und Zeitverwaltung	Datumsverwaltung und Echtzeitverwaltung oder EVU-Synchronisation
Tarif- und Zeitplanung	möglich
Verwaltung von Wartungs- und Instandhaltungs-Aktivitäten	möglich
Verwaltung von Betriebsstunden der Anlagen	über GLT
Kennzahlen (KPI, EnPI)	manuelle Konfiguration von Kennzahlen möglich
Grafische Auswertung von Kennzahlen	möglich
Auswertezyklen, Planungsfunktionen für Auswertungen	monatlich
Zeitbasis für Lastgänge	1-Stunden-Intervall Gas, 1/4-Stunden-Intervall Strom

Filter- und Suchfunktionen	möglich
Energiebericht oder Energiereport (Export einer Berichtsdatei aus der Software)	kann generiert werden, enthält Angaben zu Kostenstellen und Energieverbrauch
Editoren für Berichte, Reports und/oder Druckausgaben	möglich
BAFA	Software ist in der BAFA Liste für förderfähige Energiemanagementssoftware gelistet
Datensicherheit	Software erfüllt DSGVO
Schulungen	Schulungen beim Kunden
Produktsupport, Software-Support	Telefon- oder Internet-Hotline
Handbuch, online Hilfe, Kontextmenü, Hilfeseite im Internet (FAQ), Anwenderforen	Online-Hilfe
Test-Software, Demo-Software, Gastzugang im Internet	Demo-Version

Tabelle 1: Mindestanforderungen an eine EDM-Software nach ISO 50001 (In Anlehnung an TÜV Süd AG, persönliche Kommunikation, 27. Juli 2021)

3.2.2 Anforderungen des Bundesamtes für Wirtschaft und Ausfuhrkontrolle

ISd der Bundesförderung für Energieeffizienz in der Wirtschaft des deutschen BAFA besteht nach Nummer 5.3 die Möglichkeit, eine Förderung für den Kauf und die Einrichtung von EDM-Softwares sowie für die Schulung des Personals zu erhalten. Dabei müssen nach dem Merkblatt „Modul 3 – MSR, Sensorik und Energiemanagement-Software“ des BAFA die in Tabelle 2 dargestellten Mindestanforderungen eingehalten werden (BAFA, 2020, S. 6).

Funktion	Beschreibung
Datenauswertung	Ausgabe von Summen, Mittelwerten, Extremwerten
	Bildung von Kennzahlen zu Energieverbrauch, spezifischen Energieverbräuchen, Energieverbrauch pro Bezugsgröße, Brennstoffverbrauch pro Bezugsgröße
	Auflösung der Daten in vorgegebenen Zeitintervallen, frei definierbar
	Kostenermittlung: Energietarif-Eingabefunktion, Zuordnung von Kostenstellen
Visualisierung	Darstellung per Liniendiagramm (Ganglinie), Balkendiagramm
	Möglichkeit der individuellen Diagrammanpassung, freie Wahl der zeitlichen Auflösung, Aufnahme mehrerer Kurven in einem Diagramm, Einblenden von Grenzwerten
Berichtswesen	Ausgabe zeitgesteuerter Energieberichte (z. B. monatlicher Bericht), Darstellung lang- und kurzfristiger Verbrauchsentwicklung
	Elektronischer Versand der Berichte, Ausgabe in gängigem Format (z. B. PDF/Word)
Alarmer	Frühwarnmechanismus, individuelle Festlegung von Schwellenwerten, automatische Alarmierung bei Überschreitung von Schwellenwerten
	Elektronische Übermittlung des Alarms
Integration in bestehende Systeme	Datenimport zur Integration beliebiger Messdaten, Datenexport in gängige Formate (z. B. csv, xlsx)
	Generierung von Datenpunktlisten
	Leittechnik: Gebäudeleittechnik (GLT), Prozessleittechnik (PLT)
Support	Support bei Problemen über mindestens drei Jahre ab Erwerb, Mitarbeiterschulung, Einrichtung der Software, Updateservice

Tabelle 2: Mindestanforderungen an eine EDM-Software nach dem BAFA (In Anlehnung an BAFA, 2020)

3.2.3 Anforderungen des Marktspiegels der Energieagentur Nordrhein-Westfalen

In diesem Kapitel werden die Anforderungen dargestellt, welche im Rahmen einer Befragung von Herstellern von Softwares zur Energiedatenerfassung und zum Energiemonitoring ermittelt wurden. Diese Erhebung hat die deutsche Energieagentur NRW in Kooperation mit der perpendo GmbH durchgeführt und dahingehend einen sogenannten Marktspiegel erstellt. Die Daten wurden per Fragebogen erhoben und basieren auf den Angaben der Hersteller. Mit Stand Oktober 2021 wurden 50 EDM-Softwaresysteme analysiert. Der Marktspiegel gibt eine Gesamtübersicht über die aktuell am Markt verfügbaren Systeme. Interessierte Unternehmen oder Körperschaften des öffentlichen Rechts können mit dem online zur Verfügung gestellten Tool der Energieagentur NRW verschiedenste Produkte miteinander vergleichen und den Marktspiegel als Entscheidungshilfe zur Produktfindung nutzen. Außerdem werden den Herstellern von EDM-Softwares Informationen bereitgestellt, anhand derer die Funktionen der eigenen Software validiert werden können (Energieagentur NRW, 2021). Es ist zu beachten, dass diese Anforderungen keine Mindestkriterien im engeren Sinne darstellen, sondern aufzeigen sollen, welche Features die Softwares der einzelnen Anbieter aufweisen. Zudem kann ein Softwarehersteller für EDM-Systeme erkennen, ob sein Produkt den Anforderungen des Marktes entspricht. Je mehr Features die Software unterstützt, umso besser kann sie sich am Markt etablieren. Tabelle 3 zeigt diese Anforderungen des Marktes an eine EDM-Software nach dem Marktspiegel der Energieagentur NRW.

Funktion	Beschreibung
Aufbau des Systems und Schnittstellen	Zugriff auf die Datenbestände via Inter-/Intranet erforderlich
	Cloud-Installation möglich
	M-Bus
	BACNet
	OPC
	LON
	EIB/KNX
	TCP /IP
	Modbus
	Anbindung an weitere Datenbanksysteme möglich
	Software ohne spezielle eigene Hardwarekomponenten funktionsfähig
	Berechtigungskonzept für den Datenzugriff vorhanden (Nutzerverwaltung mit Nutzerprofilen)
	Mandantenfähigkeit vorhanden
Exportschnittstellen für Rohdaten	

	Vordefinierte Schnittstellen für den manuellen Import von Datensätzen vorhanden
	Schnittstelle zum Import von Versorgerdaten via Meteringcode vorhanden
Datenerfassung	Manuelle Zähldateneingabe in das System möglich
	Ankopplung mobiler Geräte zur Datenerfassung möglich
	Plausibilitätsprüfung automatisch erfasster Daten
	Ersatzwertbildung bei Datenlücken
	Erfassungszeit unterschiedlicher Datensammler synchronisierbar
	Verwaltung Zählerdaten
	Verwaltung Produktionsdaten
	Verwaltung Flächen
	Verwaltung Kostenstellen
	Verwaltung Betriebsdaten / Betriebszustände
	Verwaltung von virtuellen Zählern
	Verwaltung von Grenzwerten
	Generieren von Kennzahlen
	Eingabe individueller Attribute als Datenfeld für jedes Medium möglich
	Zusätzliche Zustandsgrößen und Stoffdaten als Zeitreihen im System verwaltbar
	Georeferenzierung von Zählstellen möglich
Witterungsreinigung	
Kompatibilität mit Normen	System nach BAFA-Liste förderfähig
	Abbildung der DIN EN ISO 50001 möglich
	Integriertes Dokumentenmanagement
Abrechnung	Verwaltung von Kosten
	Kostenstellen im System verwaltbar und Zählern zuzuordnen
	Verwaltung von unterschiedlichen Tarifmodellen
	Definition individueller Verteilschlüssel
Visualisierung und Berichte	Diagramm Standard (Balken, Kurve)
	Diagramm x/y (beide Achsen frei wählbar)
	Programmoberfläche durch den Anwender in Form eines Dashboards frei konfigurierbar
	Standardisierte vorgefertigte Berichte möglich
	Automatische Erstellung von Excel-Berichten mit zusätzlichen Funktionen
	Eigener Reportgenerator / -editor zur Erstellung von Berichten vorhanden
	Auswertzyklen bei der Berichtserstellung frei wählbar
Allgemein	Schulungen sind vorhanden
	Kostenfreier Testzugang/Demoversion
	Supporthotline

Tabelle 3: Anforderungen an eine EDM-Software nach der Energieagentur NRW (In Anlehnung an Energieagentur NRW, 2021)

3.2.4 Kundenspezifische Anforderungen

Um weitere kundenspezifische Anforderungen zu identifizieren, wird auf Feedbacks von Stakeholdern der evon GmbH zurückgegriffen. Diese Feedbacks wurden dem Unternehmen per Telefon oder E-Mail übermittelt und stichpunktartig in einem eigenen

Microsoft Teams Kanal dokumentiert, allerdings nicht auf Qualität geprüft bzw. in die Software implementiert. Alle Feedbacks stammen von Stakeholdern, welche die Software EMSControl der evon GmbH bereits im Einsatz haben. Diese Anforderungen werden in Tabelle 4 zusammengefasst dargestellt.

Anforderungen aus den Feedbacks
Witterungsbereinigung mithilfe von Gradtagzahlen
Verwendung von Online-Wetterdaten zur Witterungsbereinigung, wenn keine Außentemperatursensoren vorhanden sind
Automatische, rückwirkende Aufteilung der Zählerdaten bei Ausfällen
Automatische Erkennung von Peaks
Zähler in der Software komplett zurücksetzen
Einheiten können manuell für das jeweilige Medium geändert werden, derzeit in der Software fix vorgegeben
Erstellung und Darstellung von EnPIs
MSCONS Export der Zählerdaten
Aktuelle ISO 50001 Zertifizierung

Tabelle 4: Kundenfeedbacks der evon GmbH (eigene Darstellung)

3.3 Anforderungsdokumentation

Im Anschluss erfolgt im Requirements Engineering die Anforderungsdokumentation. Hierbei werden die ermittelten Anforderungen zusammengefasst und beschrieben, um die Verständigung zwischen den einzelnen AdressatInnen zu erleichtern. Die Dokumentation kann hierbei mittels natürlichsprachigem Text in Prosaform bis hin zu modellbasierten Techniken, wie Use-Case-Diagrammen, erfolgen (Pohl & Rupp, 2015, S. 35). Da die identifizierten Anforderungen in dieser Masterarbeit für unterschiedlichste AdressatInnen, z. B. TechnikerInnen, ManagerInnen oder einfache BenutzerInnen der EDM-Software verfügbar sein sollen, hat sich der Autor für die Erstellung der Dokumentation in Prosaform entschieden. Dies hat den Vorteil, dass natürliche Sprache ohne Einarbeitungszeit von allen AdressatInnen gelesen und verstanden werden kann (Pohl, 2010, S. 308–309; Pohl & Rupp, 2015, S. 53). Die zuvor identifizierten Anforderungen wurden in 51 Requirements zusammengefasst, beschrieben und den in Kapitel 4.1 erläuterten Quellen und Anforderungsarten zugeordnet. Die Anforderungsdokumentation wird in Tabelle 5 dargestellt.

Nr.	Name	Beschreibung	Quellen	Art
Req-1	Hardwareunabhängigkeit	Es können unterschiedlichste Hardwarekomponenten für die Zählerdatenaufzeichnung verwendet werden, solange diese über die verfügbaren Bussysteme kommunizieren.	Dokumente, Systeme in Betrieb	Qualität
Req-2	Standardisierte Bussysteme	Die Software unterstützt zur Kommunikation mit den Hardwarezählern folgende Bussysteme: Modbus, KNX, BACnet, OPC, LON, TCP/IP, MBus.	Dokumente, Systeme in Betrieb	Funktional
Req-3	Datenimport und-export	Es können Zählerdaten mittels Excel- bzw. CSV-Dateien in die Datenbank der Software gespielt werden.	Dokumente, Systeme in Betrieb	Funktional
Req-4	Kopplung an andere Datenbanken	Es können Daten aus anderen Datenbanken in die Datenbank der Software integriert bzw. exportiert werden.	Dokumente, Systeme in Betrieb	Funktional

Req-5	Fernzugriff auf Software	Es kann aus der Ferne auf das Softwaresystem zugegriffen werden.	Dokumente	Qualität
Req-6	Cloud-Installation	Die Software kann als SaaS zur Verfügung gestellt werden.	Systeme in Betrieb	Qualität
Req-7	Webservice	Die aktuellen Zählerstände können über einen Internetbrowser abgerufen werden (auch außerhalb des Firmennetzwerks). Dadurch können auch mithilfe mobiler Geräte Zählerdaten erfasst werden.	Stakeholder, Systeme in Betrieb	Qualität
Req-8	Datenerfassung	Die Zählerdaten können im System manuell oder automatisch erfasst werden. Bei der manuellen Erfassung tragen die Benutzenden die Daten händisch in der Visualisierung ein. Bei der automatischen Erfassung funktioniert dies vollautomatisch über die Buskommunikation zum Smart Meter.	Dokumente, Systeme in Betrieb	Funktional
Req-9	Summen, Mittel- und Extremwerte	Die Daten werden als Summen-, Mittel- und Extremwerte dargestellt.	Dokumente	Funktional
Req-10	Plausibilitätsprüfung	Die automatisch erfassten Zählerdaten werden automatisch auf Plausibilität geprüft. Wird eine Abweichung festgestellt, wird dieser Eintrag markiert.	Dokumente, Systeme in Betrieb, Stakeholder	Funktional
Req-11	Ersatzwertbildung	Entstehen aufgrund von Kommunikationsausfällen Lücken in den Daten, erfolgt eine automatische, rückwirkende Aufteilung der Zählerdaten. Die aufgeteilten Werte werden markiert.	Dokumente, Systeme in Betrieb, Stakeholder	Funktional
Req-12	Virtuelle Zählstellen	Es können in der Software virtuelle Zählstellen definiert werden. Diese Berechnung der Werte erfolgt über mathematische Grundrechnungsarten.	Dokumente, Systeme in Betrieb	Funktional
Req-13	Kostenstellenzuordnung	Zähler können in der Software einer Kostenstelle zugeordnet werden.	Dokumente, Systeme in Betrieb	Funktional
Req-14	Verwaltung Zählernummern	Es können Zählernummern für die einzelnen Softwarezähler vergeben werden.	Dokumente, Systeme in Betrieb	Funktional
Req-15	Verteilschlüssel	Energiedaten können prozentual auf virtuelle Zähler oder Kostenstellen aufgeteilt werden.	Dokumente	Funktional
Req-16	Medium und Einheiten	Für jeden Zähler kann das Medium und die dazugehörige Einheit manuell eingestellt werden.	Systeme in Betrieb, Stakeholder	Funktional
Req-17	Zeitsynchronisation	Es ist möglich eine EVU-Impulsvorgabe für die Zeitsynchronisation zu verwenden.	Dokumente, Systeme in Betrieb	Funktional
Req-18	Drittmengenabgrenzung	In der Software ist eine Drittmengenabgrenzung möglich.	Dokumente	Funktional
Req-19	Tarifmodelle	Es können Tarife verwaltet werden.	Dokumente, Systeme in Betrieb	Funktional
Req-20	Betriebsstunden	Eine Verwaltung der Betriebsstunden der Anlagen ist möglich.	Dokumente, Systeme in Betrieb	Funktional
Req-21	Zeitbasis für Lastgänge	Die Datenaufzeichnung erfolgt in 15-Minuten-Intervallen.	Dokumente	Funktional
Req-22	Georeferenzierung	Für jeden Zähler können Georeferenzen definiert und auch grafisch abgebildet werden.	Dokumente, Systeme in Betrieb	Funktional
Req-23	Verwaltung von Grenzwerten	Individuelle Festlegung von Grenzwerten möglich.	Dokumente, Systeme in Betrieb	Funktional
Req-24	Alarmausgabe	Werden Grenzwerte überschritten, gibt die Software die Alarme über die Visualisierung, SMS und E-Mail aus.	Dokumente, Systeme in Betrieb	Funktional
Req-25	Generieren von Kennzahlen	In der Software können vom Benutzer Kennzahlen definiert werden. Die Berechnung und Speicherung erfolgt automatisch in der Software.	Dokumente, Systeme in Betrieb, Stakeholder	Funktional
Req-26	Berechtigungsverwaltung	In der Software können Lese- und Schreibberechtigungen für die jeweiligen Benutzerrollen definiert werden.	Dokumente, Systeme in Betrieb	Funktional
Req-27	Filter- und Suchfunktion	Es kann in der Software nach Zählern gesucht und auch nach diesen gefiltert werden.	Dokumente, Systeme in Betrieb	Funktional
Req-28	MSCONS Export	Es können Zählerdaten in das MSCONS-Format konvertiert und exportiert werden.	Systeme in Betrieb, Stakeholder	Funktional
Req-29	Witterungsbereinigung	Mittels Wetterdaten können Gradtage berechnet werden, welche zur Witterungsbereinigung dienen.	Dokumente, Systeme in Betrieb, Stakeholder	Funktional
Req-30	Online-Wetterdaten	Sind keine Temperaturfühler vorhanden, kann die Software zur Berechnung der Gradtage auf Online-Wetterdaten zugreifen.	Stakeholder	Funktional
Req-31	Generierung von Datenpunktlisten	Es können die mit den Hardwarekomponenten verknüpften Datenpunkte exportiert werden.	Dokumente	Funktional
Req-32	Rücksetzen von Zählern	Zählerdaten können von den Benutzern zurückgesetzt werden.	Stakeholder	Funktional
Req-33	Verwaltung von Wartungsaktivitäten	In der Software können Facility-Management-Einträge in den Zählern erstellt werden.	Dokumente	Funktional
Req-34	Verwaltung von Kosten	Es können Energiepreise hinterlegt werden, mit welchen die gesamten Energiekosten berechnet werden können.	Systeme in Betrieb	Funktional
Req-35	Grafische Kennzahlenauswertung	Die generierten Kennzahlen können grafisch dargestellt werden.	Dokumente, Systeme in Betrieb, Stakeholder	Funktional
Req-36	Automatische Auswertungen	Es können Energieauswertungen in den gängigen Formaten PDF, Excel sowie Word automatisch generiert und abgespeichert oder per Mail versendet werden.	Dokumente, Systeme in Betrieb	Funktional
Req-37	Auswertungszeiträume	Die Auswertungszeiträume können frei definiert werden.	Dokumente, Systeme in Betrieb	Funktional
Req-38	Berichtvorlagen	Es gibt in der Software fertige Vorlagen für Berichte, die verwendet und bei Bedarf erweitert bzw. angepasst werden können.	Dokumente, Systeme in Betrieb	Funktional
Req-39	Reporteditor	Über einen internen oder externen Reporteditor können die Berichtvorlagen angepasst oder neue Berichte erstellt und in das System implementiert werden.	Dokumente, Systeme in Betrieb	Funktional
Req-40	Dynamische Reportgenerierung	Es besteht die Möglichkeit in der Visualisierung der Software dynamische Reports zu generieren und diese auch zu exportieren.	Dokumente, Systeme in Betrieb	Funktional

Req-41	Diagrammarten	Die Daten können in Linien- und Balkendiagrammen dargestellt werden.	Dokumente, Systeme in Betrieb	Funktional
Req-42	Dashboard	Das Dashboard kann von den Benutzenden selbst erstellt werden.	Systeme in Betrieb	Funktional
Req-43	Dokumentenmanagement	Die Software hat ein integriertes Dokumentenmanagement, in welchem Dateien verschiedenster Formate strukturiert abgelegt werden können.	Systeme in Betrieb	Funktional
Req-44	Leittechnik	Schnittstelle zwischen EDM-Software und Leittechniksystem verfügbar.	Dokumente	Funktional
Req-45	Schulungen	Es werden Schulungen hinsichtlich der Benutzung der Software angeboten.	Dokumente, Systeme in Betrieb	Qualität
Req-46	Supporthotline	Es steht eine Supporthotline zur Verfügung.	Dokumente, Systeme in Betrieb	Qualität
Req-47	Demoversion	Der Kunde kann zum Testen der Software eine Demoversion verwenden.	Dokumente, Systeme in Betrieb	Qualität
Req-48	Online-Hilfe	Es stehen Online-Dokumentationen zur Verfügung.	Dokumente	Qualität
Req-49	DSGVO	Die Software erfüllt die Anforderungen der DSGVO.	Dokumente	Rahmenanforderung
Req-50	BAFA-förderfähig	Die Software entspricht den Anforderungen der BAFA.	Dokumente, Systeme in Betrieb	Rahmenanforderung
Req-51	ISO 50001	Die Software entspricht den Anforderungen der ISO 50001.	Systeme in Betrieb, Stakeholder	Rahmenanforderung

Tabelle 5: Anforderungsdokumentation einer EDM-Software (eigene Darstellung)

3.4 Anforderungsprüfung

Nach der Anforderungsdokumentation erfolgt die Validierung. In diesem Schritt werden die dokumentierten Anforderungen auf Qualität geprüft, um typische Dokumentationsfehler wie Mehrdeutigkeit, Unvollständigkeit und Widersprüchlichkeit zu vermeiden. Dazu werden die dokumentierten Anforderungen den Stakeholdern vorgelegt und im Rahmen der Überprüfung sichergestellt, dass alle relevanten Anforderungen vorhanden sind. Zudem wird geprüft, ob diese adäquat beschrieben wurden, damit keine Missinterpretationen bzw. -verständnisse auftreten können und mit den Anforderungen der Stakeholder übereinstimmen (Pohl & Rupp, 2015, S. 95–97).

Die Anforderungsprüfung erfolgte im Rahmen dieser Masterarbeit via Online-Meeting, in welchem die Dokumentation anhand eines Walkthroughs besprochen und analysiert wurde. Dabei wurde schrittweise durch die Anforderungen geführt und diese genauer erklärt. Zu den Teilnehmern der Besprechung zählten der Autor, der Teamleiter der Abteilung Gebäudeautomation sowie drei Stakeholder der evon GmbH. Die Besprechung wurde am 19.10.2021 mithilfe von Microsoft Teams durchgeführt. Zusammenfassend lässt sich sagen, dass alle Stakeholder sowie der Abteilungsleiter die Ausarbeitung des Autors und die dahingehende Identifizierung der Anforderungen sowie deren Dokumentation als gut befunden haben. Ferner gaben die Stakeholder im Rahmen des Meetings bekannt, welche Anforderungen für sie und ihre Unternehmen eine besondere Relevanz aufweisen und jedenfalls im Rahmen der Implementierung berücksichtigt werden sollten. Das zugehörige Besprechungsprotokoll des Online-Meetings befindet sich in Anhang B.

3.5 Anforderungsverwaltung

Ein weiterer Prozess im Requirements Engineering ist die Anforderungsverwaltung – auch Requirements Management genannt – und sollte flankierend zu den anderen Hauptaktivitäten erfolgen. Das Ziel dieser Aktivität ist die strukturierte Änderungssteuerung, die Versionskontrolle, die Priorisierung und die persistente Verfügbarkeit der Anforderungen über den gesamten Produktlebenszyklus hinweg (Pohl, 2010, S. 594–597). Die aktuelle Anforderungsdokumentation einer EDM-Software wurde als Version 1.0, in einem vom Autor dieser Masterarbeit eigens erstellten Microsoft Teams-Kanal, abgelegt und dies den entsprechenden MitarbeiterInnen kommuniziert. Ziel soll es in Zukunft sein, regelmäßig die Anforderungen zu ermitteln und diese in das Dokument mit neuer Versionsnummer einzupflegen. Des Weiteren sollen die Anforderungen priorisiert werden, sodass diese je nach Relevanz abgearbeitet werden können. Zudem ist es notwendig, die bereits in die Software implementierten Anforderungen entsprechend zu markieren.

4 Ist-Analyse des EMSControl

Das EMSControl ist ein Softwaremodul der Automatisierungssoftware XAMControl von der Firma evon GmbH. Um das ganzheitliche System besser verstehen zu können, erfolgt in diesem Kapitel zuerst eine kurze Firmenbeschreibung sowie eine Erläuterung der Software XAMControl. Im Anschluss werden der technische Aufbau, die Systemarchitektur und die Funktionen des EMSControl formal und mit Abbildungen aus einem Live-System grafisch dargestellt.

4.1 evon GmbH

Die evon GmbH ist ein junges und dynamisches IT-Unternehmen im Dienstleistungsbereich. Ziel des Unternehmens ist die zukunftsfähige und nachhaltige Realisierung von Automatisierungslösungen für die unterschiedlichsten Anwendungsfälle. Für die Umsetzung der industriellen Automatisierung wird das firmeneigene Leitsystem XAMControl verwendet. Die evon GmbH bietet Hardwarekomponenten sowie Softwarelizenzen und -dienstleistungen an. Im Bereich Hardware werden Rechner sowie Eingangs- und Ausgangskomponenten, wie beispielsweise Beckhoff Busklemmen und Dell Monitore vertrieben. Für die Nutzung von XAMControl werden Lizenzgebühren nach Anzahl der verwendeten Datenpunkte erhoben. Zu den Softwaredienstleistungen zählen die Programmierung als auch die Inbetriebnahme von Softwareapplikationen, welche mittels XAMControl entwickelt werden. Das Unternehmen mit Sitz in St. Ruprecht an der Raab wurde 2009 gegründet und hat mit März 2022 einen Mitarbeiterstand von 60 Personen. Seit April 2011 ist die Dürr Austria GmbH, Tochter der deutschen Dürr Group GmbH, an der evon GmbH beteiligt (Evon, 2021).

4.2 XAMControl

4.2.1 Allgemein

Die Gründer der evon GmbH hatten die Vision eine zukunftssichere, hardwareunabhängige, vollständig skalierbare sowie schnittstellenfreundliche Automatisierungs- und Leittechnikplattform zu entwickeln. Aus diesem Grund wurde die Software XAMControl entwickelt, welche auf moderne Softwarewerkzeuge und Hochsprachen zurückgreift. Auf einfachste Art und Weise kann mit XAMControl die Digitalisierung und Vernetzung von Maschinen und Anlagen realisiert werden.

XAMControl deckt den Bereich der Automatisierungspyramide von der Visualisierung bis zum Hardwarefeld, inklusive eines Programmiersystems für SPSen, ab. Die Softwareentwicklung basiert auf dem Konzept der Objektorientierung. Als hardwareunabhängiges System erzielt XAMControl den Vorteil, Fremdsysteme einfach einzubinden und damit die Flexibilität der Kunden zu erhöhen (Evon, 2021).

In dem Bereich Gebäudeleittechnik erfolgt die Messung, Steuerung und Regelung (MSR) von Lüftungsanlagen, Heiz- und Kühlaggregaten bis hin zur Fernalarmierung im Fehlerfall mittels Email- oder SMS–Alarmweiterleitung. Der Facility-Manager erhält durch XAMControl einen Überblick über alle Gewerke des Gebäudes. Bei der Verkehrstechnik wird auf Tunnelautomatisierung und Tunnelleittechnik fokussiert und durch Verkehrsdatenerfassung sowie Verkehrssteuerung ein zentrales Leitsystem geschaffen. In der Industrieabteilung erfolgt die Entwicklung von verfahrenstechnischen Prozessen in der Pharma-, Halbleiter- und Lebensmittelindustrie. Das EnMS-Modul im XAMControl dient der Datenerfassung, Verbrauchsberechnung und grafischen Darstellung von Energiedaten für Analysezwecke im Rahmen des betrieblichen EnMS (Evon, 2021).

4.2.2 Technologie

XAMControl basiert auf dem Microsoft .NET Framework 4.7. Die Programmierung erfolgt nach dem SPS Standard IEC-61131-3 Function-Block-Diagram. Zudem kann für komplexere Anwendungsfälle auf die Hochsprache C# zurückgegriffen werden. Ein großer Vorteil von XAMControl ist die Verwendung von Soft-SPSen. Dadurch wird eine Abstrahierung der Automatisierungshardware erzielt und die Applikation kann entkoppelt von den Feldgeräten projiziert werden. Im Anschluss werden die virtualisierten SPSen auf die Laufzeitumgebung verteilt. Das Datenrouting zwischen den SPSen wird automatisch vom System durchgeführt. Aufgrund der Virtualisierung kann im Fehlerfall einer Hardware, das jeweilige virtuelle SPS-Programm von einem anderen Feldgerät ausgeführt werden. XAMControl kann als Laufzeitumgebung auf alle gängigen Hardwarekomponenten mit Windows Server 2019 oder Windows 10 (IoT) als Betriebssystem zurückgreifen. Für Windows 11 ist XAMControl aktuell nicht freigegeben (Evon, 2021).

Alle Daten, sowohl Applikations- und Laufzeitdaten, werden in einer Microsoft SQL Server 2019 Datenbank archiviert. Dadurch können Fremdsysteme aufgrund der Offenheit einfach angebunden werden. Des Weiteren können durch die Microsoft Excel-

Integration einfache Betriebsdatenauswertungen ohne umfangreiches Datenbank Know-How per Knopfdruck erstellt werden. Ferner wird Microsoft Excel auch zur Anlagenparametrierung verwendet. Parameter können dadurch einfach exportiert, bearbeitet oder wieder importiert werden (Evon, 2021).

4.2.3 Systemarchitektur

Applikationen werden im XAMControl zentral entwickelt und anschließend in verteilten, dezentralen Systemen ausgeführt. Das System ist in Anlehnung an die Automatisierungspyramide in drei Ebenen untergliedert. (Evon, 2021). Abbildung 8 stellt die Systemarchitektur grafisch dar.

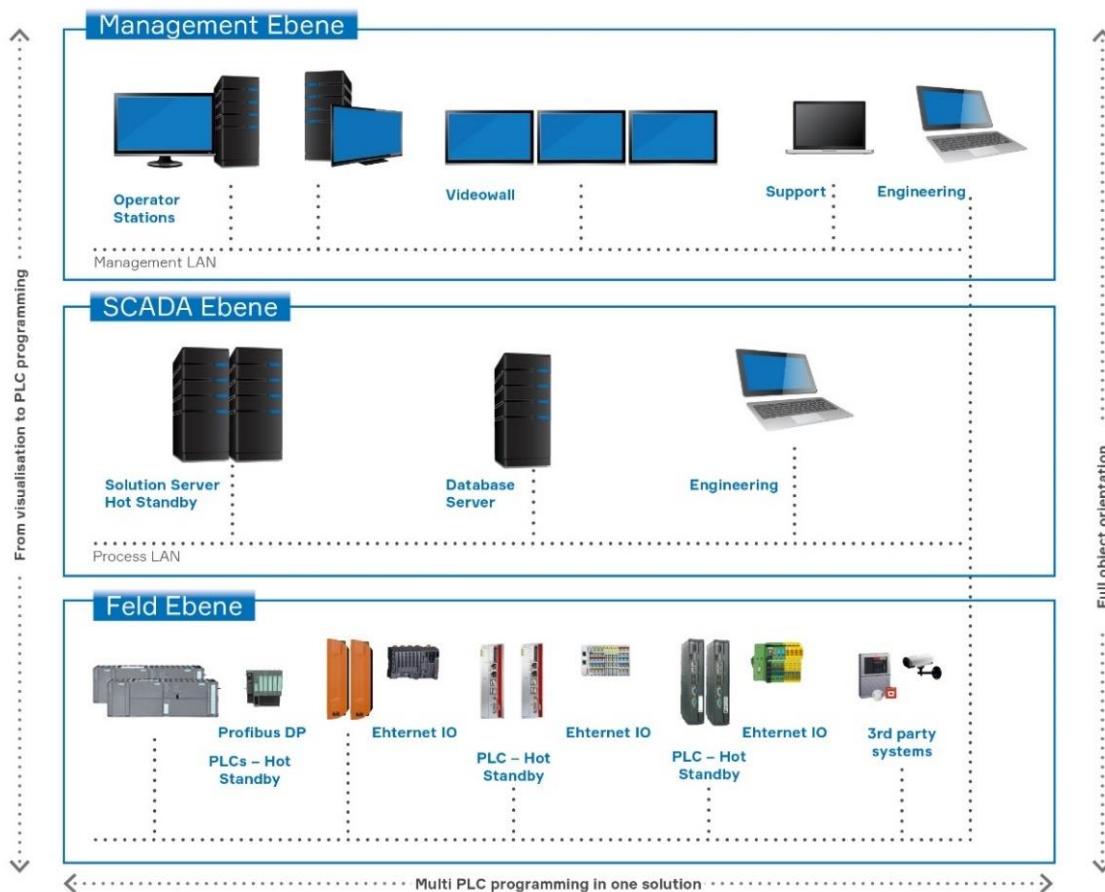


Abbildung 8: Systemarchitektur XAMControl (Evon, 2021)

Auf der Feldebene befinden sich Buskoppler, Gateways, Sensoren und Aktoren. Des Weiteren befinden sich in dieser Ebene die Industrie-PCs, auch Controller genannt, welche als Soft-SPS eingesetzt werden. Durch diesen Systemaufbau können die einzelnen SPS-Programme auf den verschiedenen Controllern dezentral ausgeführt werden. Dazu werden die ausgelagerten Programmteile in einer eigenen Microsoft SQL-Datenbank gespeichert, welche sich auf dem jeweiligen dezentralen Controller befindet. Zudem

können die Controller redundant ausgeführt und dadurch – bei einem Hardwarefehler eines Industrie-PCs – der Betrieb aufrechterhalten werden. Damit ist die Ausfallsicherheit bereits auf Feldebene gewährleistet. Die Controller verfügen über Ethernet-Schnittstellen und können mit weiteren Geräten über proprietäre Protokolle, wie z. B. EIB/KNX, MBus, Modbus, OPC UA oder BACnet, kommunizieren (Evon, 2021).

In der zweiten Ebene, auch als SCADA-Ebene bezeichnet, befindet sich der Kern des Systems, der Solution Server, welcher die Hauptdatenbank des Systems beinhaltet. Der Solution Server kann ebenfalls Soft-SPSen ausführen und stellt die Visualisierung des Systems bereit. Auch der Solution Server kann redundant ausgeführt werden. Ferner kann die SQL-Datenbank der Server abgesetzt, in einem Datenbank-Cluster betrieben werden (Evon, 2021).

In der Management-Ebene befinden sich die Bedienstationen. Dies sind einfache Arbeitsplatzrechner im Leittechniknetzwerk, welche die Visualisierungsumgebung von XAMControl, die XAMIris, ausführen. Mithilfe der XAMIris kann von den Bedienstationen aus auf die Daten des Solution Servers zugegriffen und diese entsprechend dargestellt werden. Parameteränderungen werden zunächst an den Solution Server übermittelt und im Anschluss an die dezentral betriebenen Controller weiterverteilt (Evon, 2021).

4.3 Systemmodule

XAMControl umfasst eine Vielzahl von Prozessen, welche gemeinsam im Systemverbund interagieren. Es gibt die Standard-Dienste, welche für das laufende System immer nötig sind und weitere branchenspezifische Anwendungen für die unterschiedlichen Anwendungsgebiete, u. a. Treiber für Kommunikationen mit externen Systemen. Jede Anwendung ist als eigenes Modul ausgeführt und kann bei Bedarf auf verschiedener Hardware laufen, um gegebenenfalls eine Lastaufteilung zu erreichen. Alle Anwendungen von XAMControl setzen auf das Microsoft .NET Framework 4.7. Davon werden die Frameworks WPF zur grafischen Darstellung, WCF für die Kommunikation zur Außenwelt und das ADO.NET Entity Framework für die Anbindung der Datenbank verwendet (Evon, 2021). Tabelle 6 stellt die Systemmodule dar, welche branchenunabhängig auf nahezu jedem System zum Einsatz kommen.








Symbol	Name	Beschreibung
	XAMPLC	Die XAMPLC ist der Grundkern von XAMControl. Sie ist für die zyklische Ausführung der SPS-Programme sowie für Kommunikation mit der SQL-Datenbank zuständig.
	XAMControlIDE	Die XAMControlIDE ist die Programmierumgebung des Systems.
	XAMiris	Die XAMiris ist die Visualisierungsumgebung des Systems.
	XAMArchiver	Der XAMArchiver dient zur Datenaufzeichnung.
	XAMAlarmServer	Der Alarmmanager wickelt das gesamte Alarmhandling ab.
	XAMAlertingServer	Der Alerting Server ist für die SMS- bzw. E-Mail-Alarmierung zuständig.
	XAMTraceViewer	Der XAMTraceViewer ist für die Logeinträge des Systems zuständig.

Tabelle 6: Systemmodule XAMControl (In Anlehnung an Evon, 2021)

4.4 Objektorientierung im XAMControl

Durch die durchgängige Objektorientierung von XAMControl wird den AnwenderInnen eine maximale Wiederverwertbarkeit und Vervielfältigung der entwickelten Objekte zur Verfügung gestellt (Evon, 2021). Bei diesem Programmierparadigma werden Objekte, u. a. Motoren, Ventile oder Energiezähler, in sogenannten Klassen modelliert. Die Klasse repräsentiert dabei den Bauplan für ein Objekt. Im Gegensatz dazu stellt das Objekt eine konkrete Ausprägung der jeweiligen Klasse dar. Wichtig anzumerken ist, dass jedes Objekt auf nur einer Klasse basieren kann. Aus diesem Grund kann es in einem Projekt bzw. einer Applikation eine Vielzahl an Objekten geben, welche auf derselben Klasse basieren. Beispielsweise definiert die Klasse eines Energiezählers welche Attribute und Funktionen einen konkreten Zähler beschreiben – sie stellt aber keinen konkreten Zähler mit dessen spezifischen Daten dar. Erst ein Objekt der Klasse Energiezähler repräsentiert einen konkreten Energiezähler. Der Vorgang zur Erzeugung eines Objektes wird als Instanziierung und das Objekt selbst auch als Instanz einer Klasse bezeichnet, wobei jede Instanz einen eindeutigen Namen besitzen muss (Roden, 2008, S. 12–13).

Diese im XAMControl genannten Klassen werden auch als Automation Control Container (ACC) bezeichnet und beinhalten einen Programmteil sowie grafische Elemente. Der Programmteil wird in der Soft-SPS ausgeführt und besteht aus Variablen und aus einer Logik. Die Variablen werden in jedem ACC neu definiert, wobei sie in unterschiedliche Variablentypen (Eingangs-, Ausgangs-, interne Variablen) mit verschiedenen Datentypen (Bool, Double, Int, String, etc.) unterteilt werden können. Für die Programmierung der zugehörigen Logik kann zwischen FBD und C# gewählt werden. Die grafischen Elemente werden in Panels und Popups unterteilt und werden über XAML erstellt. Unter Panel wird die grafische Darstellung der Klasse verstanden, welche in der Visualisierung dargestellt wird. Das Popup ist die grafische Bedienoberfläche und öffnet sich durch einen Klick auf ein Panel. Popups werden dazu verwendet, Detailinformationen anzuzeigen sowie dem Benutzer Eingabemöglichkeiten zu bieten und bestehen wiederum Panels. Es können eigene Klassen bzw. ACCs erstellt oder aus dem sogenannten ACC-Store der evon GmbH heruntergeladen werden. Somit müssen die Standardklassen wie einstufige Motoren oder stetige Ventile – welche nahezu in jedem Projekt in der Gebäudeleittechnik vorhanden sind – von den AnwenderInnen nicht neu programmiert, sondern können einfach aus dem Store geladen und instanziiert werden. Auch für Energiezähler gibt es bereits fertige Klassen, was die Erstellung eines EDM-Systems erleichtert (Evon, 2021).

Um die zuvor beschriebene Objektorientierung näher zu veranschaulichen, wird dies anhand einer Klasse aus dem XAMControl ACC Store demonstriert. Dazu wurde die Programmierumgebung XAMControlIDE geöffnet und das Package TCP.Janitza heruntergeladen. Dieses Package ist Teil des EMSControl und beinhaltet diverse ACCs für die Kommunikation mit Janitza Smart Meter. In Abbildung 9 wird der ACC Store, die heruntergeladenen Smart Meter Klassen sowie die beinhalteten Elemente dargestellt.

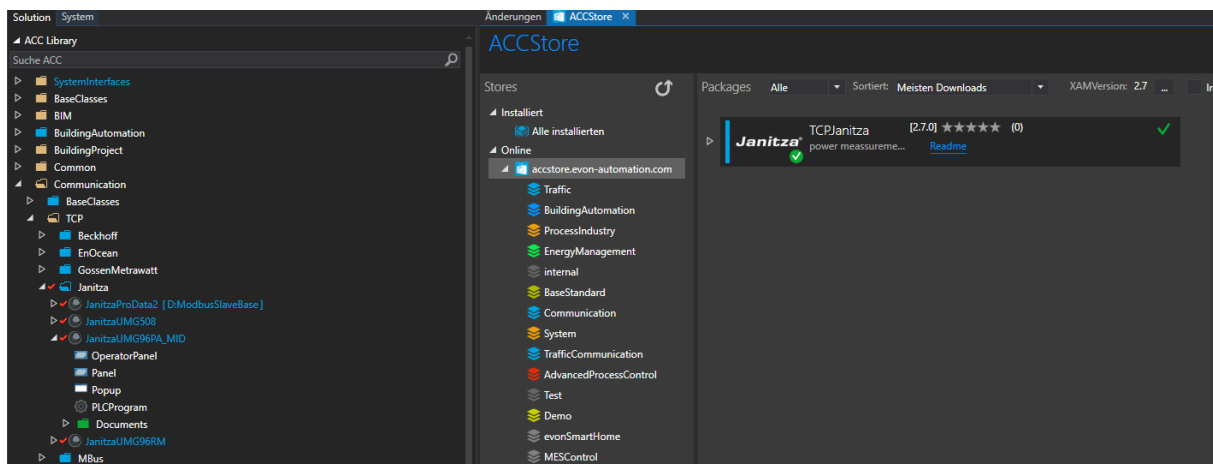


Abbildung 9: ACC Store und Janitza Smart Meter Klassen im XAMControl (eigene Darstellung)

Wie in der obigen Abbildung 9 zu erkennen ist, besteht die Klasse bzw. das ACC „Janitza UMG96PA_MID“ aus den Subelementen, Panel, Popup, OperatorPanel und PLCProgram. Die nachfolgenden Abbildungen stellen die einzelnen Elemente grafisch dar.



Abbildung 10: Beispiel Panel (eigene Darstellung)

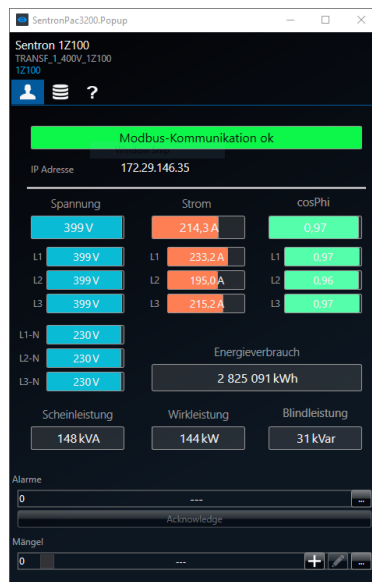


Abbildung 11: Beispiel Popup mit OperatorPanel (eigene Darstellung)

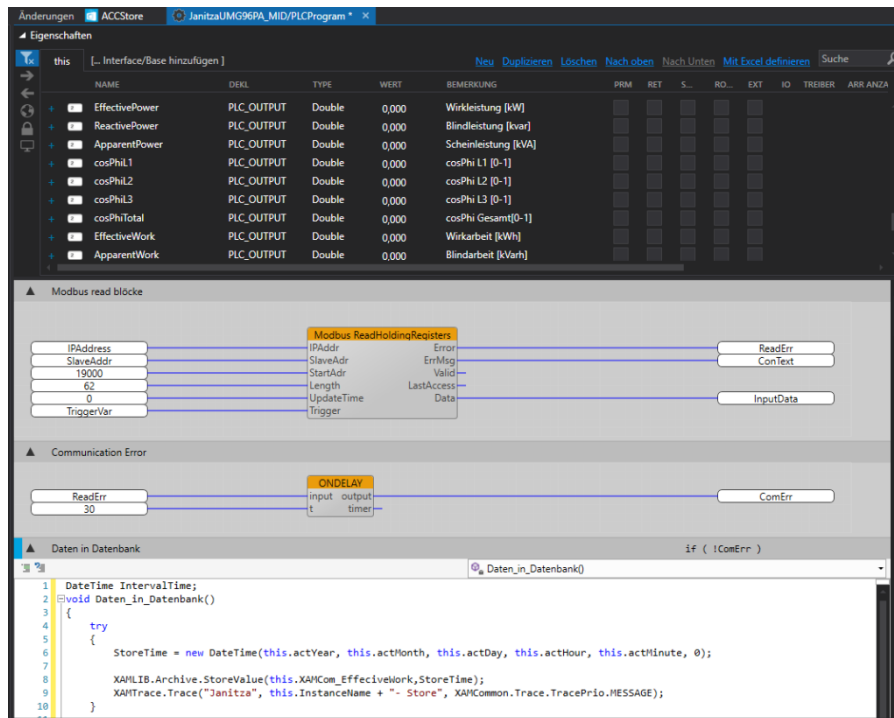


Abbildung 12: Beispiel PLCProgram (eigene Darstellung)

4.5 EMSControl

4.5.1 Allgemein

Die evon GmbH bietet mit dem EMSControl ein ganzheitliches Konzept für eine optimierte und wirtschaftliche Betriebsführung. Das EMSControl ist nach ISO 50001:2011 zertifiziert und bei dem BAFA für förderfähige Energiemanagementsoftware gelistet. Durch die Softwarelösung werden den AnwenderInnen, neben der Erfassung und Verarbeitung der Energiedaten, auch Funktionen wie Energiemonitoring, Energieprognosen, Verbrauchs- und Leistungsüberwachung sowie grafische Auswertungen zur Verfügung gestellt. Aufgrund der Hardwareunabhängigkeit können zur Datenerfassung verschiedenste Schnittstellen und Zählertypen verwendet werden. Aufgrund des offenen Datenbanksystems mit Microsoft SQL wird eine optimale Integration mit bestehenden Systemen ermöglicht. Das EMSControl der evon GmbH ist keine eigenständige Software, sondern besteht aus einzelnen ACCs im XAMControl, welche zusammen genutzt eine EDM-Software ergeben. Wird die Umsetzung eines EDM-Systems beauftragt, wird XAMControl als Softwarelösung eingesetzt und die für das EDM benötigten ACCs aus dem ACC Store geladen und in der Soft-SPS instanziiert. Ist ein spezieller Zählertyp nicht als ACC vorhanden, können die erfahrenen ProgrammiererInnen der evon GmbH dieses in kurzer Zeit entwickeln und den Stakeholdern über den ACC Store einfach zur Verfügung stellen.

4.5.2 Technische Funktionsweise

Wie in Kapitel 3.1 erwähnt, werden zuerst die Energiedaten mittels Messsystemen am Feld erfasst und anschließend an die EDM-Software übertragen. Dabei werden im XAMControl eine Vielzahl an Übertragungsprotokollen unterstützt, welche auch im EMSControl zur Anwendung kommen können. Die Liste dieser Treiber wird dabei ständig erweitert und an den Markt angepasst. Die aktuell zur Verfügung gestellten Schnittstellen zur Kommunikation mit Smart Meter sind nach Angaben des Autors dieser Masterarbeit:

- EIB/KNX
- Modbus TCP
- MBus
- BACnet
- S7TcpIpLink
- Profibus
- OPC UA
- REST
- MQTT

Des Weiteren besteht die Möglichkeit Smart Meter einzubinden, welche andere Protokolle zur Übertragung von Daten verwenden. U. a. verwenden Hersteller bei Smart Meter LoRaWAN oder LonWorks als Protokoll, welche zurzeit nicht unterstützt werden. Hierbei kann auf Gateways zurückgegriffen werden, welche LoRaWAN oder LonWorks auf Modbus oder BACnet umsetzen. Da das XAMControl und dahingehend auch das EMSControl auf ethernetbasierten Geräten, wie Server oder Industrie-PCs mit Windows 10 als Betriebssystem, ausgeführt werden, müssen alle seriellen Protokolle auf TCP übersetzt werden, sofern die Smart Meter keine direkten Ethernet-Anschlüsse besitzen. Die eigentlichen Protokolleigenschaften werden dabei nicht beeinflusst. Beispielsweise besitzen einige Modbus RTU Datenlogger ausschließlich eine serielle RS485 Schnittstelle. Um die Daten von diesen Geräten in das EMSControl übertragen zu können, wird hier ein Gateway eingesetzt, welches die serielle Schnittstelle Modbus RTU auf Modbus TCP übersetzt. Somit ist es durch Zuhilfenahme der Gateways möglich, die gängigsten Zähler des Marktes mit dem EMSControl zu verbinden. Dies wird nachfolgend durch ein Beispiel veranschaulicht.

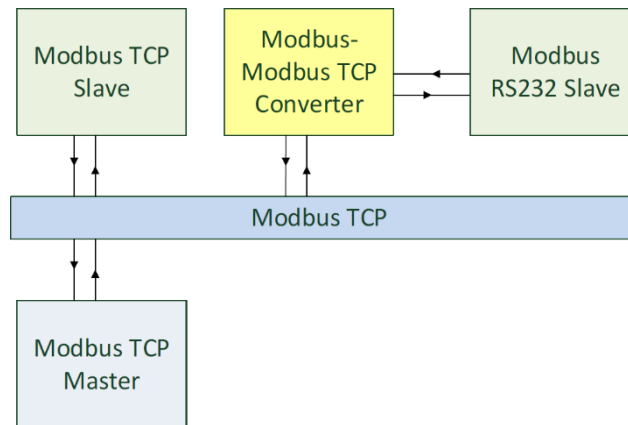


Abbildung 13: Architektur einer Modbus Datenübertragung (In Anlehnung an Vresk & Cavrak, 2016, S. 1200)

Abbildung 13 stellt eine Architektur von Modbus TCP Master, Modbus TCP Slaves und Modbus RTU/RS485 Slaves dar. Der Modbus TCP Master wäre in diesem Fall das EMSControl und ist über Ethernet mit einem Modbus TCP Slave verbunden, beispielsweise ein Janitza UMG 96RM-E Smart Meter. Ferner ist ein serieller Modbus Zähler, wie der Siemens PAC 1600, mit dem EMSControl verbunden, indem ein Modbus RTU zu Modbus TCP Gateway eingesetzt wird.

Ist die hardwareseitige, ethernetbasierte Verbindung zwischen Messeinrichtungen und dem Server bzw. Industrie-PC gegeben, wird mit der Erstellung des SPS-Programms zur Datenübertragung begonnen. Dafür werden die benötigten ACCs aus dem ACC Store geladen und instanziiert. Demonstrativ wird in Abbildung 14 eine Siemens PAC Sentron 3200 Instanz mit den ausgelesenen Werten abgebildet.

Address	Value
ComOK	True
VoltageL1	228,996826171875
VoltageL2	229,550201416016
VoltageL3	229,303405761719
VoltageL1L2	397,525573730469
VoltageL1L3	397,394165039063
VoltageL2L3	396,459686279297
VoltageL3L1	229,283493041992
VoltageAverageLN	397,12646484375
VoltageAverage	19,5182914733887
CurrentL1N	14,1542539596558
CurrentL2N	15,7740964889526
CurrentL3N	16,482213973999
CurrentAverage	0,515512228012085
cosPhiL1	0,476668122674942
cosPhiL2	0,98460191488266
cosPhiL3	0,658926755189896
cosPhiTotal	11,33579296875
ApparentPower	4,31676416015625
EffectivePower	6,230001953125
ReactivePower	27057,9351498645
EffectiveWork	27070,139170374
ApparentWork	

Abbildung 14: Beispiel EMSControl PAC Sentron 3200 Instanz (eigene Darstellung)

Der Smart Meter hat dabei die IP-Adresse 172.21.10.192 und die Daten werden über das Modbus TCP/IP Protokoll abgerufen. Auf die Logik und die genaue Programmierung der Datenübertragung wird hier nicht näher eingegangen, da dies für den weiteren Verlauf der Masterarbeit nicht relevant ist. Da die Smart Meter grundsätzlich nur Daten wie aktuelle Schein-, Blind- und Wirkleistungen, Spannungen und Stromstärken sowie den aktuellen Gesamtzählerstand erfassen und übermitteln, bedarf es für die Erstellung von Lastprofilen adäquate Verbrauchsberechnungen. Dies erfolgt beim EMSControl über weitere ACCs, welche die Zählerstände der Smart Meter verwenden und daraus Verbräuche berechnen. Es werden Intervall-, Stunden-, Tages-, Monats- und Jahresverbräuche berechnet und in die SQL-Datenbank gespeichert. Die Länge eines Intervalls kann dabei zwischen 10, 15, 20 Minuten frei gewählt werden. Auch eine EVU-Synchronisation ist möglich. Sind keine Smart Meter vorhanden, können Zählerstände manuell in das System eingetragen werden. Abbildung 15 demonstriert eine EMSControl Zählerinstanz, welche Verbrauchswerte aus dem Zählerstand des PAC Sentron 3200 Hardwarezählers berechnet.

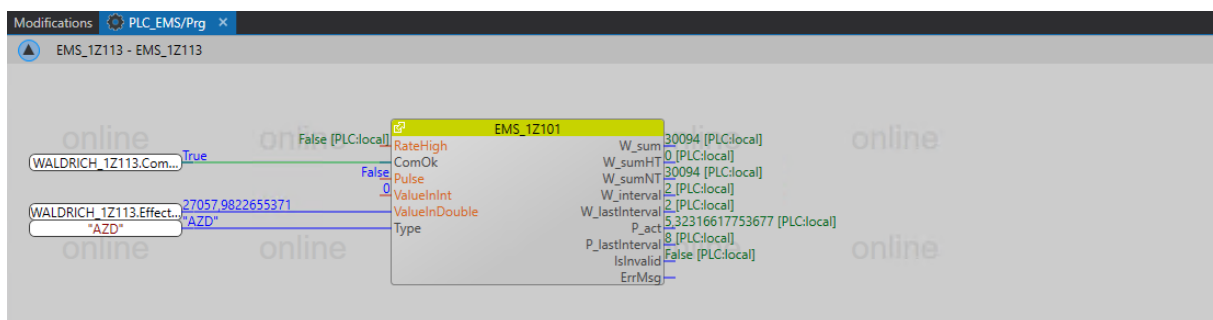


Abbildung 15: Beispiel EMSControl Zähler Instanz (eigene Darstellung)

4.5.3 Features

Neben der automatischen Verbrauchsberechnungen bzw. der manuellen Eingabe von Zählerständen können virtuelle Zählstellen angelegt werden, durch welche fiktive Werte generiert oder prozentuale Verbräuche von realen Zählstellen mittels frei definierbaren Verteilungsschlüsseln zugeordnet werden können. Für jede Zählstelle können Parameter wie Kostenstelle, Tarif, Zählernummer, Standort oder Medium angegeben werden. Ferner ist es möglich Grenzwerte zu definieren, um Alarme bei Überschreitung in der Visualisierungsumgebung XAMiris auszugeben oder per Fernalarmierung via E-Mail oder SMS zu übermitteln. Durch ein integriertes Berechtigungskonzept im XAMControl werden Rollen vergeben, sodass nur berechtigte Personen Parameter verändern können. Ebenfalls besteht die Möglichkeit zur Generierung von Hardwaredatenpunktlisten sowie von Facility-Management-Einträgen zur Protokollierung der Wartungsaktivitäten.

Aktuell kann ausschließlich eine manuelle Plausibilitätsprüfung durchgeführt werden, sodass ein Zähler grafisch markiert wird, wenn bei der Analyse unplausible Werte entdeckt wurden und ein entsprechender Button im Popup betätigt wird. Ferner existiert im System ein Dokumentenmanagement, um Dateien beliebigen Formates abzulegen. Da als Datenbank Microsoft SQL Server verwendet wird, können externe Systeme mit entsprechenden Berechtigungen mittels SQL-Statements Zählerdaten exportieren als auch importieren. Nachfolgend wird das Popup mit den aktuellen Leistungen, historischen Verbräuchen sowie die einstellbaren Parametern eines EMSControl Zählers dargestellt.

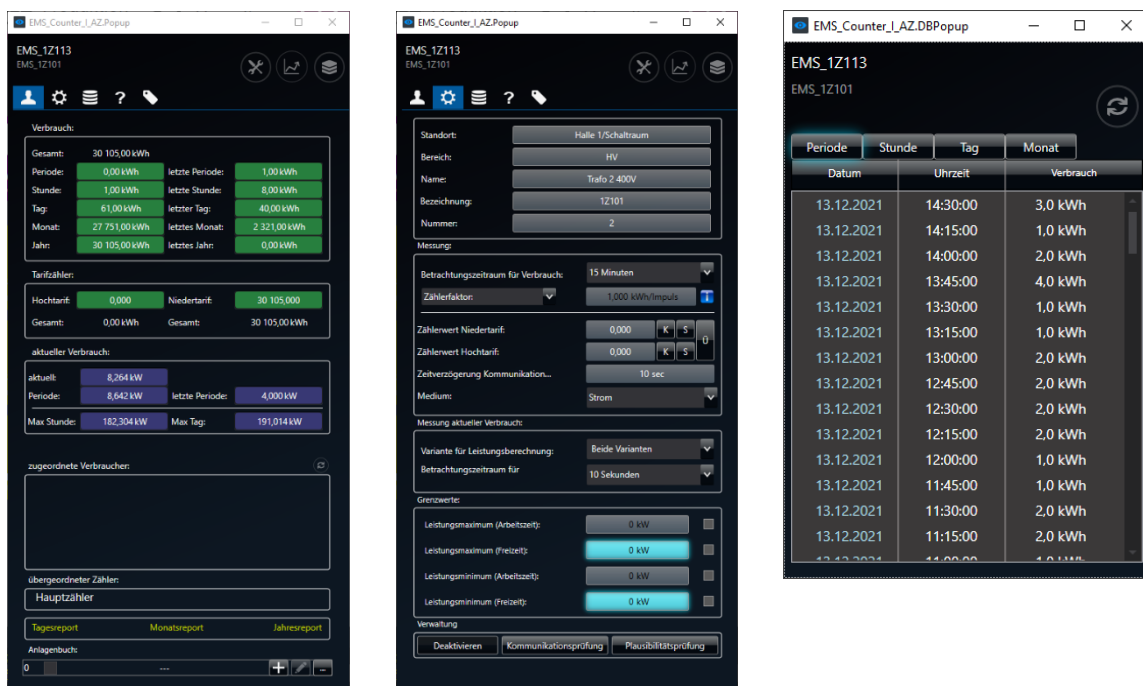


Abbildung 16: Beispiel EMSControl Zähler Popup (eigene Darstellung)

Zur Visualisierung der Zähler stehen den AnwenderInnen verschiedene Möglichkeiten zur Verfügung, welche je nach definierten Berechtigungen, selbst angepasst werden können. Als Standardfunktion erfolgt eine Listendarstellung der Zähler inkl. Such- und Filterfunktion. Zusätzlich können Baumdiagramme als grafische Darstellung erstellt werden. Demonstrativ zeigt Abbildung 17 eine dynamisch gefilterte Zählerliste.

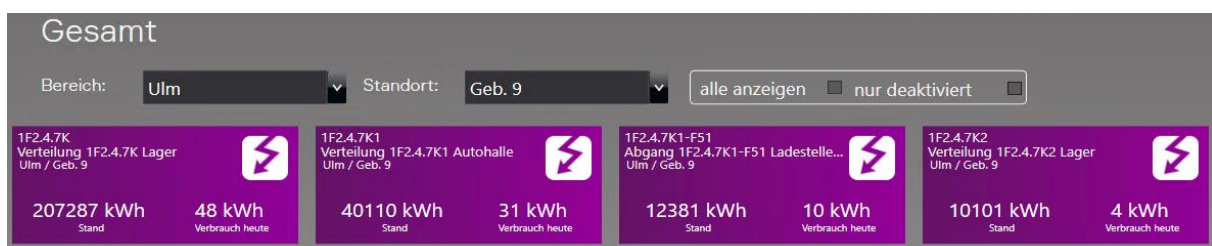


Abbildung 17: Beispiel EMSControl Listendarstellung mit Filterfunktion (eigene Darstellung)

Des Weiteren können die aktuellen Zählerdaten durch einen integrierten Webserver über Internetbrowser wie Google Chrome oder Microsoft Edge abgerufen werden. Über einen sogenannten Relay-Server, welchen die evon GmbH bereitstellt, kann auch außerhalb des Firmennetzwerks auf den Webserver zugegriffen werden.

Das EMSControl verwendet für die Auswertungen SQL Reports, welche mit dem kostenfreien SQL Report Builder von Microsoft erstellt werden. Standardmäßig existieren für jede Zählerstelle drei Reports für Tages-, Monats- und Jahresverbräuche, bei welchen die Lastprofile via Balkendiagramme dargestellt werden. Die Zeiträume sind dabei frei definierbar. Die Reports sind über das Popup des Zählers abrufbar. Ferner haben die NutzerInnen die Möglichkeit, mithilfe des SQL Report Builders, eigene Auswertungen für spezifische Anforderungen zu erstellen. Zudem können durch den SQL Reporting Service die Reports automatisch, zu frei wählbaren Zeitpunkten generiert und abgespeichert sowie auch per Mail versendet werden. Abbildung 18 stellt einen Monatsreport mit den Tagesverbräuchen des Zählers dar.

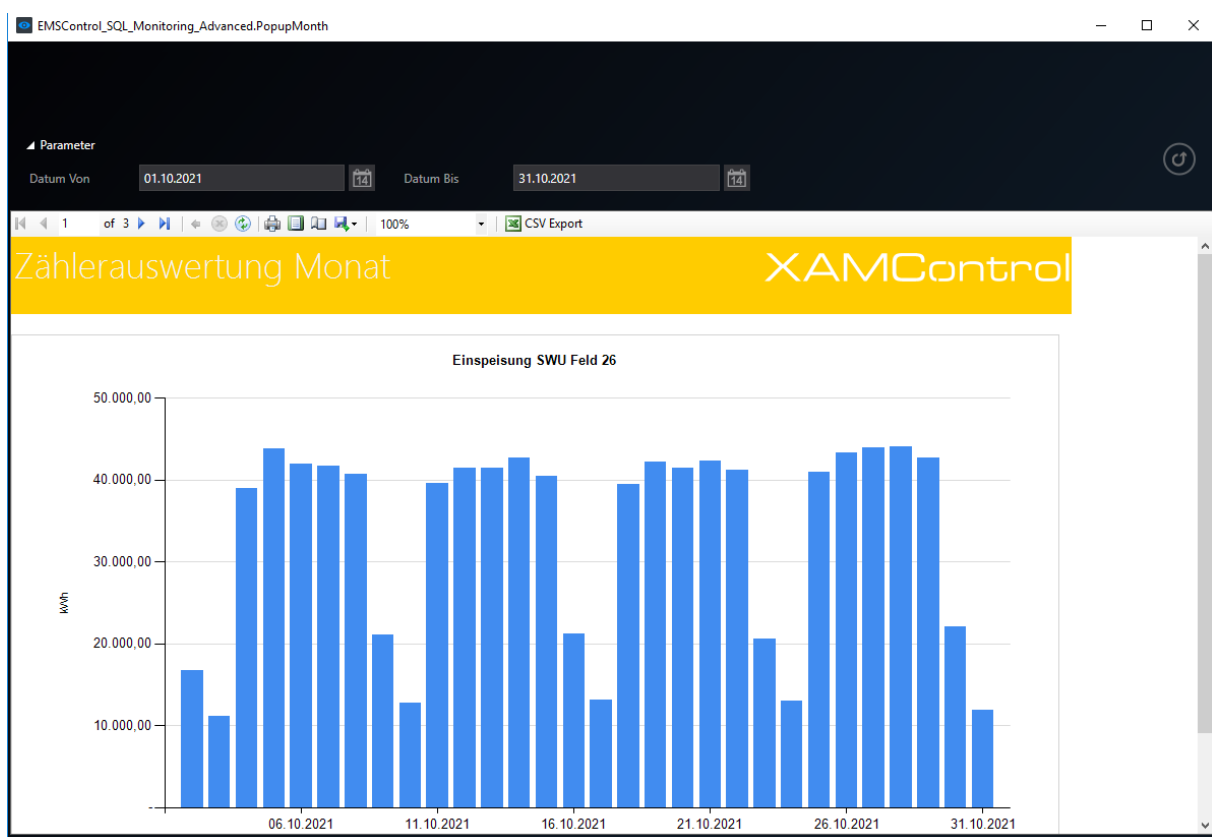


Abbildung 18: Beispiel EMSControl Monatsreport (eigene Darstellung)

Für die Managementebene gibt es als weitere Auswertungsmethode ein Dashboard zur Generierung von dynamischen Reports. Per Drag-and-Drop können Verbräuche von verschiedenen Zählern in einem Diagramm zusammengefasst dargestellt werden.

Dadurch können Verbräuche miteinander verglichen und somit die Analysemöglichkeiten erweitert werden. Zudem können im Dashboard Energiepreise hinterlegt und Energiekosten berechnet werden. Ferner existiert im XAMControl ein Trendviewer. In diesen können Trenddaten ebenfalls per Drag-and-Drop hinzugefügt und exportiert werden.

Hinsichtlich der Supportmöglichkeiten der Software bietet die evon GmbH Online-Dokumentationen sowie eine 24/7 Supporthotline an. Da das System auf Windows 10 Server oder Industrie-PCs installiert wird, kann bei vorhandener Internetverbindung über VPN-Clients oder Tools wie TeamViewer auf das System zugegriffen und dadurch Fernwartungen durchgeführt werden. Des Weiteren können die Stakeholder Schulungen buchen, welche entweder in den Schulungsräumen der evon GmbH, extern bei den KundInnen oder online durchgeführt werden. Damit potenzielle AuftraggeberInnen das System testen können, existiert eine Demoversion mit zeitlicher Begrenzung.

5 Anforderungsintegration mittels Design Science

In diesem Kapitel erfolgt der empirische Teil der Masterarbeit, in welchem versucht wird, ein aktuelles Praxisproblem der evon GmbH zu lösen, indem die Marktanforderungen in das obsoleete EMSControl implementiert werden. Als Forschungsparadigma dient dabei die gestaltungsorientierte Wirtschaftsinformatik, auch Design Science genannt.

5.1 Design Science

Design Science ist ein praxisnaher Forschungsansatz mit dem Ziel, neue Ergebnisse in Form von sogenannten Artefakten zu entwerfen und praxistauglich nutzbar zu machen. Die IT-Artefakte werden auf Basis einer Problemstellung entwickelt und evaluiert. Häufig sind dies sogenannte Instanziierungen, sprich keine kompletten lauffähigen Systeme, sondern Ausschnitte aus Gesamtsystemen. Dabei geht es im Vergleich zu den klassischen Wissenschaften weniger um einen Erkenntnisbeitrag, sondern um einen Ergebnisbeitrag. Einerseits liegt der Fokus von Design Science klar auf der Praxisnähe. Andererseits ist es notwendig, dass der Forschungsansatz die wissenschaftliche Rigorosität wahrt. Dazu existieren in der Literatur einige Modelle für das Vorgehen nach Design Science. Alle Modelle setzen sich das Ziel, den Design Science Prozess strukturiert aufzubauen, um die Verfolgung von Relevanz und Rigorosität zu unterstützen (Benner-Wickner, Kneuper & Schlömer, 2020, S. 4–6; Robra-Bissantz & Strahringer, 2020, S. 173–174; Winter, 2008, S. 470).

Trotz der wissenschaftlichen Vorgehensweise ist die Kernanforderung des Design Science, für die Praxis relevante Artefakte zu schaffen. Ein Artefakt kann aus physischen oder abstrakten Objekten bestehen, welche von Menschen entwickelt wurden. Ein physisches Artefakt zur Lösung des Problems der Mobilität wäre beispielsweise das Auto. In den Wirtschaftswissenschaften handelt es sich bei den Artefakten hauptsächlich um abstrakte Objekte. Diese reichen von mathematischen Algorithmen bis hin zu vollständigen Softwarelösungen in Form von Smartphone-Apps (Frauchinger, 2017, S. 116; Johannesson & Perjons, 2014, S. 3). In der Informationstechnologie werden diese Artefakte in die vier Kategorien Konstrukte, Modelle, Methoden und Instanziierung klassifiziert (Baskerville, Baiyere, Gergor, Hevner & Rossi, 2018, S. 361; March & Smith, 1995, S. 265). Die im Rahmen dieser Masterarbeit erzeugten Artefakte sind Instanziierungen, welche einem Teilsystem in der objektorientierten Programmierung entsprechen (Robra-Bissantz & Strahringer, 2020, S. 174). Dies können u. a.

Softwarebibliotheken, grafische Elemente, Dynamic Link Libraries, Frameworks oder Algorithmen sein (Mijač, 2019, S. 313).

Die Forschungsmethode besteht grundsätzlich aus den zwei Basisaktivitäten „build“ und „evaluate“. Erstere ist für die Erstellung des Artefakts zuständig, wohingegen letztere die Nützlichkeit des erzeugten Artefakts untersucht und beurteilt (March & Smith, 1995, S. 254). Diese beiden Aktivitäten werden in den etablierten Modellen konsequent durchgeführt. Je nach Modell kommen allerdings weitere Nebenprozesse zur Anwendung. Aufgrund der einfachen Verständlich- und Umsetzbarkeit hat sich der Autor dieser Masterarbeit für das im europäischen Raum etablierte Modell nach ÖSTERLE ET AL. entschieden. Dieses besteht aus einem vierstufigen Prozess, welcher sich in die Iterationsphasen Analyse, Entwurf, Evaluation und Diffusion gliedert und nachfolgend näher erläutert werden (Benner-Wickner et al., 2020, S. 4–6; Österle et al., 2010, S. 667–668; Robra-Bissantz & Strahringer, 2020, S. 173–174).

Zuerst erfolgt in der Analysephase die Klärung des Kontextes für die Gestaltung des Artefakts. Dabei werden die grundsätzlichen Anforderungen an das Artefakt definiert. Ferner werden die Ziele, welche mit dem Artefakt verfolgt werden, als auch die Adressatengruppe definiert. Zusätzlich wird ein allgemeiner Überblick über das jeweilige Thema benötigt. Klassische Forschungsmethoden in der Analysephase sind Umfragen, Fallstudien, Literaturrecherchen, Interviews oder Analysen von Informationssystemen. Anschließend wird in der Entwurfsphase das gewünschte Artefakt entwickelt. Es sind dabei die Anforderungen aus der Analyse zu berücksichtigen sowie geeignete methodische Vorgehensweisen für die Entwicklung zu wählen und zu begründen (Benner-Wickner et al., 2020, S. 7; Österle et al., 2010, S. 667).

Da es sich bei Design Science um ein empirisches Forschungsparadigma handelt, ist eine quantitative oder qualitative Evaluation der entwickelten Artefakte notwendig, um die Rigorosität zu wahren. Dabei werden die Ergebnisse, also die erstellten Artefakte, gegen die identifizierten Anforderungen abgeglichen und kontrolliert, ob diese erfüllt werden. Darüber hinaus sollten in der Evaluation die erstellten Artefakte auch in realer Umgebung und nicht ausschließlich in Testumgebungen eingesetzt werden. Dadurch wird validiert, ob sie praxistauglich sind und den gewünschten Nutzen bringen. Die Ergebnisse des Evaluationsprozesses werden so lange in die vorgelagerten Phasen Analyse und Entwurf zurückgeführt, bis die Artefakte als angemessen gut bewertet werden. Forschungsmethoden zur Evaluation der Artefakte sind u. a. Laborexperimente,

Feldexperimente oder die Prüfung durch ExpertInnen (Benner-Wickner et al., 2020, S. 7; Österle et al., 2010, S. 668).

Design Science ist an einer größtmöglichen Diffusion der erzielten Ergebnisse interessiert. Wenn ein Artefakt nach Abschluss der Evaluation als nützlich betrachtet wird, soll im Rahmen der Diffusion das entwickelte Artefakt dauerhaft verfügbar gemacht werden. Adäquate Instrumente zur Veröffentlichung der Ergebnisse sind beispielsweise wissenschaftliche Aufsätze oder Konferenzbeiträge. Bei Zusammenarbeit mit einem Praxisunternehmen können die Ergebnisse u. a. in Prozessabläufe oder Leitfäden zur Verfügung gestellt werden. Wird im Zuge des Design Science eine Software entwickelt, kann auch diese veröffentlicht werden (Benner-Wickner et al., 2020, S. 8; Österle et al., 2010, S. 668). Abbildung 19 bildet die Phasen von Design Science nach ÖSTERLE ET AL. grafisch ab.

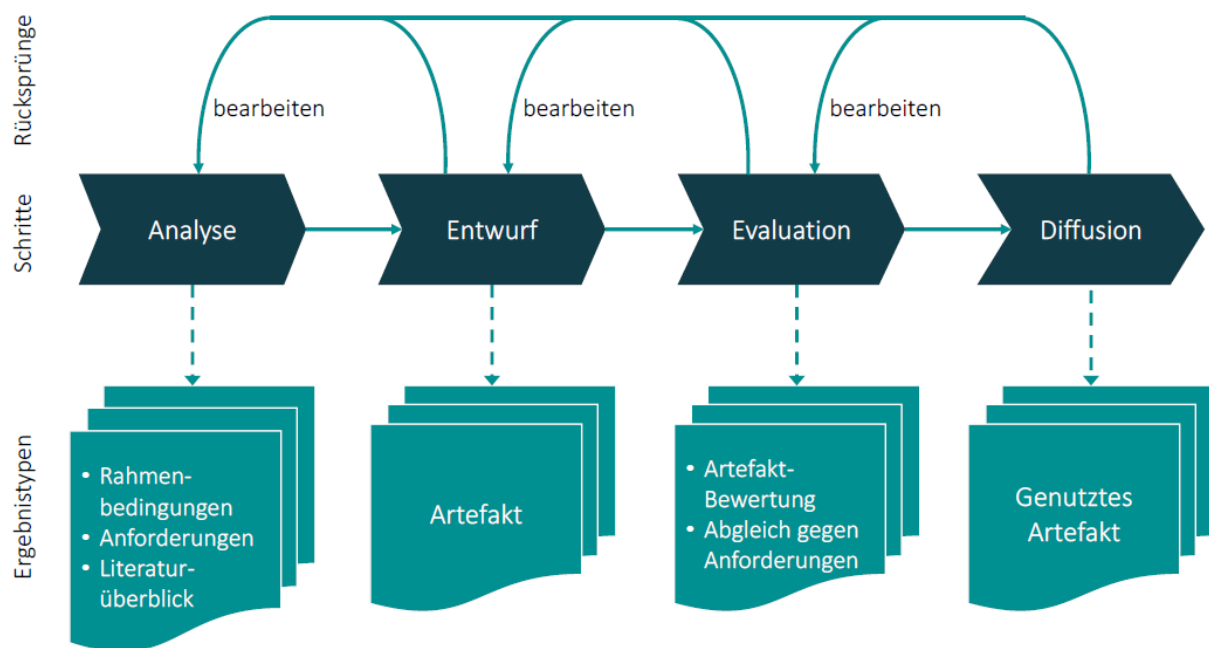


Abbildung 19: Design Science Prozess nach Österle et al. (Benner-Wickner et al., 2020, S. 7)

Nach HEVNER ET AL. sind innerhalb des Design Science Prozesses sieben Richtlinien zu beachten, welche in Tabelle 7 dargestellt werden (Hevner, March, Park & Ram, 2004, S. 83).

Richtlinie	Beschreibung
Richtlinie 1: Design as an Artifact	Im Rahmen des Design Science muss ein brauchbares Artefakt in Form eines Konstrukts, eines Modells, einer Methode oder einer Anwendung hervorgebracht werden.
Richtlinie 2: Problem Relevance	Ziel des Design Science muss es sein, technologiebasierte Lösungen für wichtige und relevante Praxisprobleme zu entwickeln.
Richtlinie 3: Design Evaluation	Der Nutzen und die Qualität eines Artefakts müssen durch adäquat durchgeführte Evaluierungsmethoden rigoros nachgewiesen werden.
Richtlinie 4: Research Contributions	Es müssen allgemeingültige Beiträge geleistet werden.
Richtlinie 5: Research Rigor	Für die Entwicklung und Evaluierung des Artefakts sind etablierte Methode Methoden anzuwenden.
Richtlinie 6: Design as a Search Process	Die Suche nach einem wirksamen Artefakt erfordert den Einsatz der verfügbaren Mittel, um die gewünschten Ziele zu erreichen und gleichzeitig den Gesetzen des Problemumfelds gerecht zu werden.
Richtlinie 7: Communication of Research	Design Science muss für technologieorientierte sowie für managementorientierte Zielgruppen effektiv präsentiert werden.

Tabelle 7: Allgemeine Design Science Richtlinien (In Anlehnung an Hevner et al., 2004, S. 83)

In Tabelle 8 wird gezeigt, wie die Richtlinien im Rahmen dieser Masterarbeit eingehalten werden.

Richtlinie	Umsetzung
Richtlinie 1: Design as an Artifact	Erstellung von Instanziierungen für das EMSControl.
Richtlinie 2: Problem Relevance	Verbesserung des obsoleten EMSControl durch Implementierung der aktuellen Marktanforderungen.
Richtlinie 3: Design Evaluation	Evaluierung der Artefakte mittels ExpertInneninterviews.
Richtlinie 4: Research Contributions	Erstellung eines Anforderungskatalogs für EDM-Softwaresysteme sowie Lessons Learned hinsichtlich der objektorientierten Programmierung.
Richtlinie 5: Research Rigor	Requirements Engineering, Wasserfallmodell, ExpertInneninterviews nach MAYRING und DRESING/PEHL.
Richtlinie 6: Design as a Search Process	Iterative Vorgehensweise durch Rückkopplung der Ergebnisse aus den ExpertInneninterviews.
Richtlinie 7: Communication of Research	Freigabe der Masterarbeit sowie Veröffentlichung der neuen Version des EMSControl im ACC Store der evon GmbH.

Tabelle 8: Design Science Richtlinien bezogen auf die Masterarbeit (In Anlehnung an Hevner et al., 2004, S. 83)

5.2 Analyse

Auf Basis der Problemstellung bei der evon GmbH hinsichtlich der obsoleten EDM-Software EMSControl wurde bereits in den bisherigen Kapiteln mit der Analysephase begonnen. Dabei wurde eine Literaturrecherche durchgeführt und ein Überblick zum Thema Energiemanagement gegeben sowie die relevantesten Rahmenbedingungen erläutert. Zudem wurden mittels Requirements Engineering die Anforderungen an eine EDM-Software identifiziert. Im Anschluss wurde der aktuelle Ist-Zustand des EMSControl erhoben. Zum Abschluss der Analysephase des Design Science Prozesses müssen die gesammelten Anforderungen mit den Ergebnissen der Ist-Analyse verglichen werden, um daraus die spezifischen Anforderungen für die Artefakte zu generieren. Bei diesem Vergleich konnte der Autor dieser Masterarbeit feststellen, dass die normativen sowie kundenspezifischen Anforderungen von dem EMSControl unzureichend erfüllt werden. Zwar werden viele Basisfunktionen einer EDM-Software vom System unterstützt, jedoch fehlen einige – für die AnwenderInnen essenzielle – Funktionen, wie u. a. nach ISO 50001 verpflichtende Kennzahlenberechnungen sowie Witterungsbereinigungen. Die Ergebnisse des Vergleichs zwischen den identifizierten Anforderungen und der aktuellen Version des EMSControl werden in Tabelle 9 dargelegt.

Nr.	Name	Aktuelle Umsetzung im EMSControl	Anforderung erfüllt
Req-1	Hardwareunabhängigkeit	Bis auf die Einschränkung, dass das System auf einem Windows 10 Betriebssystem mit ausreichend RAM-Speicher und CPU-Leistung installiert sein muss, ist eine Hardwareunabhängigkeit gegeben. Auch bei den einzubindenden Feldgeräten gibt es bis auf eine nahezu keine Einschränkungen, da auch mit Gateways gearbeitet werden kann.	Ja
Req-2	Standardisierte Bussysteme	Es werden die gängigsten Kommunikationsschnittstellen unterstützt. Die Software unterstützt zur Kommunikation mit den Hardwarezählern folgende Bussysteme: Modbus, EIB/KNX, BACnet, OPC UA, MBus, REST, MQTT, Profibus, S7TcplpLink.	Ja
Req-3	Datenimport und-export	Über SQL-Statements im SQL Management Studio oder über das bereitgestellte Export/Import-Tool.	Ja
Req-4	Kopplung an andere Datenbanken	Da SQL Server als Datenbank verwendet wird, ist eine Kopplung zu anderen Datenbanken einfach möglich.	Ja
Req-5	Fernzugriff auf Software	Zugriff auf Server oder Industrie-PC, auf welchen XAMControl installiert ist via VPN, TeamViewer etc.	Ja
Req-6	Cloud-Installation	Wird derzeit nicht unterstützt.	Nein
Req-7	Webservice	System enthält einen eigenen Webserver, um Daten über einen Browser abrufen zu können. Zugriff auch über Relay-Server außerhalb des Firmennetzwerks möglich.	Ja
Req-8	Datenerfassung	Daten können automatisch über Smart Meter oder manuell erfasst werden.	Ja
Req-9	Summen, Mittel- und Extremwerte	Summenwerte und Extremwerte vorhanden, Mittelwerte nicht vorhanden.	Nein
Req-10	Plausibilitätsprüfung	Derzeit manuelle Plausibilitätsprüfung.	Nein
Req-11	Ersatzwertbildung	Wird derzeit nicht unterstützt.	Nein
Req-12	Virtuelle Zählstellen	Erstellung von virtuellen Zählstellen möglich.	Ja
Req-13	Kostenstellenzuordnung	Wird derzeit nicht unterstützt.	Nein
Req-14	Verwaltung Zählernummern	Über Popup des jeweiligen Zählers individuell parametrierbar.	Ja
Req-15	Verteilschlüssel	Generierung von fiktiven Werten und prozentuale Verteilung der Daten möglich.	Ja
Req-16	Medium und Einheiten	Derzeit eine Auswahl von sieben Medien mit fixen Einheiten implementiert.	Nein
Req-17	Zeitsynchronisation	Verwendung von EVU-Impulsvorgabe möglich.	Ja
Req-18	Drittmengenabgrenzung	Durch Erstellung eigener (virtueller) Zähler möglich.	Ja

Req-19	Tarifmodelle	Es kann derzeit zwischen Hoch- und Niedertarif gewählt werden.	Ja
Req-20	Betriebsstunden	Sind auch die MSR-Anlagen mit Pumpen, Ventilen oder Ventilatoren im XAMControl implementiert, wo auch das EMSControl verwendet wird, können die Betriebsstunden aufgezeichnet werden.	Ja
Req-21	Zeitbasis für Lastgänge	Datenaufzeichnung kann zwischen 10, 15, 20 Minuten oder EVU-Synchronisation gewählt werden.	Ja
Req-22	Georeferenzierung	Standortangaben über Popup des jeweiligen Zählers individuell parametrierbar. Grafisch können Baumdiagramme verwendet werden. Des Weiteren können auch Grundrisse dargestellt und die Zähler am korrekten Standort abgebildet werden.	Ja
Req-23	Verwaltung von Grenzwerten	Leistungsgrenzen über Popup des jeweiligen Zählers individuell parametrierbar (Warnung oder Alarm).	Ja
Req-24	Alarmausgabe	Alarmer werden – mit oder ohne Soundausgabe – grafisch in der Visualisierung (XAMiris) dargestellt. Mail- oder SMS-Versand der Alarmer ist ebenfalls möglich.	Ja
Req-25	Generieren von Kennzahlen	Wird derzeit nicht unterstützt.	Nein
Req-26	Berechtigungsverwaltung	XAMControl bietet ein eigenes Berechtigungskonzept, welches auch im EMSControl zur Anwendung kommt.	Ja
Req-27	Filter- und Suchfunktion	In den Listendarstellungen der Zähler kann gefiltert und gesucht werden.	Ja
Req-28	MSCONS Export	Wird derzeit nicht unterstützt.	Nein
Req-29	Witterungsbereinigung	Wird derzeit nicht unterstützt.	Nein
Req-30	Online-Wetterdaten	Wird derzeit nicht unterstützt.	Nein
Req-31	Generierung von Datenpunktlisten	Hardwaredatenpunkte können in eine Excel-Datei exportiert werden.	Ja
Req-32	Rücksetzen von Zählern	Wird derzeit nicht unterstützt.	Nein
Req-33	Verwaltung von Wartungsaktivitäten	Für jeden Zähler können sogenannte Anlagenbucheinträge erstellt, bearbeitet und gelöscht werden.	Ja
Req-34	Verwaltung von Kosten	Im EMSControl existiert ein spezielles Dashboard, in welchem Energiepreise eingegeben und damit die Energiekosten berechnet werden können.	Ja
Req-35	Grafische Kennzahlenauswertung	Wird derzeit nicht unterstützt.	Nein
Req-36	Automatische Auswertungen	Über SQL Reporting Service möglich.	Ja
Req-37	Auswertungszeiträume	Drei Standardreports vorhanden, in welchen der Zeitraum frei gewählt werden kann.	Ja
Req-38	Berichtvorlagen	Drei Standardreports vorhanden, welche über den SQL Report Builder individuell angepasst werden können.	Ja
Req-39	Reporteditor	Als Reporteditor dient der kostenfreie SQL Report Builder.	Ja
Req-40	Dynamische Reportgenerierung	In speziellen Dashboards können Zählerdaten dynamisch zusammengefasst werden. Hier ist kein Export möglich. Allerdings existiert im XAMControl ein sogenannter Trendviewer. In diesen können Trenddaten dynamisch, per Drag-and-Drop angezeigt und in PDF-, Word- oder CSV-Dateien exportiert werden.	Ja
Req-41	Diagrammarten	Derzeit Balkendiagramme vorhanden.	Nein
Req-42	Dashboard	Die grafische Visualisierung kann von den AnwenderInnen mit vorhandenen Berechtigungen individuell angepasst werden.	Ja
Req-43	Dokumentenmanagement	XAMControl beinhaltet ein integriertes Dokumentenmanagement.	Ja
Req-44	Leittechnik	Entweder direkt XAMControl als Leittechniksystem oder über Schnittstellen wie OPC UA oder BACnet zu anderen Leittechniksystemen.	Ja
Req-45	Schulungen	Die evon GmbH bietet Schulungen im Hause evon als auch Schulungen vor Ort bei den KundInnen an.	Ja
Req-46	Supporthotline	Es existiert eine 24/7 Supporthotline.	Ja
Req-47	Demoversion	Für Testzwecke und als Entscheidungshilfe kann ein Demozugang beantragt werden.	Ja
Req-48	Online-Hilfe	Im sogenannten Developer-Portal der evon GmbH gibt es eine ausführliche Online-Dokumentation.	Ja
Req-49	DSGVO	DSGVO wird eingehalten.	Ja
Req-50	BAFA-förderfähig	Das EMSControl ist gelistet.	Ja
Req-51	ISO 50001	Das EMSControl ist nach ISO 50001:2011 zertifiziert, jedoch nicht nach ISO 50001:2018.	Nein

Tabelle 9: Vergleich zwischen EDM-Softwareanforderungen und EMSControl (eigene Darstellung)

Durch den Abgleich zwischen den Anforderungen und den Features des EMSControl konnten 14 Requirements identifiziert werden, welche aktuell nicht bzw. nur teilweise vom System erfüllt werden. Diese 14 Requirements wurden in einem Meeting zwischen dem Autor und dem Abteilungsleiter der Gebäudeleittechnik der evon GmbH besprochen

und es wurde definiert, welche Anforderungen im Rahmen dieser Masterarbeit umgesetzt werden sollen. Das Besprechungsprotokoll des Meetings vom 06.12.2021 befindet sich in Anhang B. In der Besprechung wurde festgehalten, dass bis auf Req-5 und Req-51 alle Anforderungen im Zuge der Masterarbeit realisiert werden. Dadurch wurden alle Anforderungen für die zu erstellenden Artefakte definiert, die Analysephase des Design Science Prozesses somit abgeschlossen und es konnte im nächsten Schritt mit der Entwicklung der Artefakte begonnen werden.

5.3 Entwurf

In diesem Kapitel wird gezeigt, wie die Artefakte zur Implementierung der identifizierten Anforderungen in das EMSControl implementiert wurden. Die Entwicklung einer Software basiert auf einem sogenannten Software Development Life Cycle (SDLC), welcher die einzelnen Phasen eines Softwareentwicklungsprozesses definiert. Da bei der evon GmbH für die Entwicklung des EMSControl das Wasserfallmodell nach ROYCE zur Anwendung kommt, wurde auch zur Erstellung der Artefakte dieses Modell angewendet. Dieses Vorgehensmodell, welches von Winston Royce 1970 konzipiert wurde, wird in der Praxis verbreitet eingesetzt, da es eines der simpelsten phasenbasierten SDLC-Modelle darstellt (Balaji & Murugaiyan, 2012, S. 26–28; Boehm, 1988, S. 63; Burghardt, 2018, S. 151–152).

Die einzelnen Phasen haben definierte Start- und Endzeitpunkte, welche mit eindeutigen Ereignissen zusammenhängen. Bei erfolgreichem Abschluss einer Phase wird mit der Nachfolgephase begonnen. Grundsätzlich erfolgt der Ablauf in Vorwärtsrichtung. Bei bestimmten, schwerwiegenden Abweichungen, wie beispielsweise neue Software- oder Systemanforderungen, wird zu einer vorherigen Phase zurückgesprungen. Das Wasserfallmodell bietet einen klaren Prozess und eine klare Struktur für die Erstellung einer Software. Durch diese lassen sich umfangreiche sowie auch kleine Softwareumsetzungen präzise planen und zuverlässig durchführen (Burghardt, 2018, S. 151).

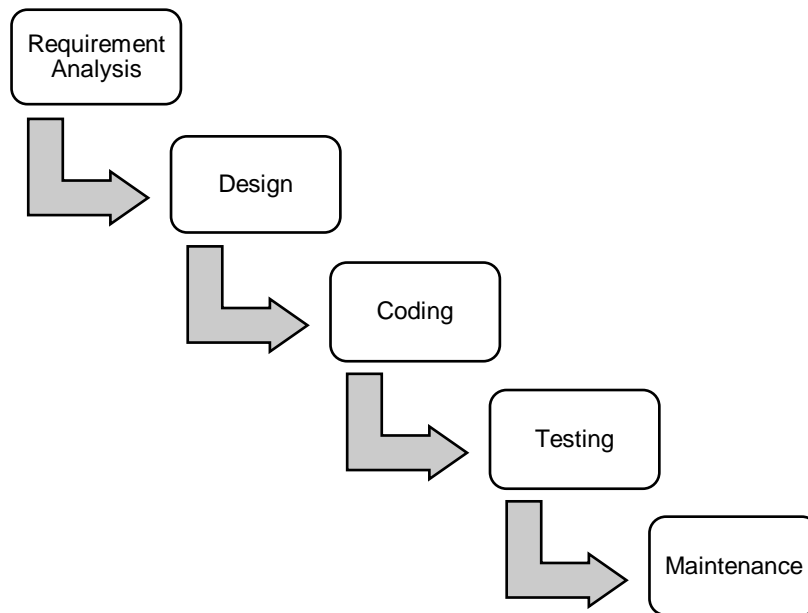


Abbildung 20: Wasserfallmodell nach Royce (In Anlehnung an Munassar & Govardhan, 2010, S. 95)

Abbildung 20 veranschaulicht die Projektphasen des Wasserfallmodells nach ROYCE. Nach diesem Vorgehensmodell wird bei einer Softwareentwicklung mit der Anforderungserhebung begonnen. Es handelt sich hierbei um eine Anforderungsanalyse des Systems und der Software. Zuerst werden die notwendigen Hardwarespezifikationen identifiziert. Des Weiteren wird untersucht, welche Funktionen die Software erfüllen muss, um den Anforderungen zu entsprechen. Anschließend erfolgt ein Grob- und Detailentwurf der Softwarearchitektur. In der Phase Coding wird der Programmcode erstellt und in der Phase Testing durch Debuggen auf Fehler kontrolliert sowie untersucht, ob die gewünschten Funktionen adäquat implementiert wurden. Wird die Software von den TesterInnen als releasefähig erachtet, wird sie freigegeben. Im Laufe der Zeit werden mithilfe von Software-Wartungen Fehler behoben und Optimierungen durchgeführt (Boehm, 1988, S. 63; Munassar & Govardhan, 2010, S. 96).

Wie bereits erwähnt, wird die Umsetzung der einzelnen Artefakte an das Wasserfallmodell nach ROYCE angelehnt. Die Softwareanforderungen an die Artefakte wurden bereits in Kapitel 5.2 erhoben und können ohne Abänderung in die Entwurfsphase übernommen werden. Da alle im Zuge der Masterarbeit entwickelten Artefakte, Instanziierungen im EMSControl sind und dieses wiederum im XAMControl integriert ist, gelten die Systemanforderungen von XAMControl. Die aktuell von der evon GmbH empfohlenen Systemanforderungen für die Verwendung von XAMControl sind in Tabelle 10 dargestellt.

Komponente	Anforderung
Betriebssystem	Windows Server 2019 64 Bit oder Windows 10 (IoT) 64 Bit
CPU	8 x 2 GHz
RAM	16 GB
Freier Festplattenspeicher	120 GB
Datenbank	SQL Server 2019 Standard
.NET Framework	4.7.2

Tabelle 10: Systemanforderungen XAMControl (In Anlehnung an Evon, 2021)

Da es sich bei der Realisierung der Anforderungen um mehrere – auf den Softwareumfang bezogen – kleine Artefakte handelt, wird das Software-Design vernachlässigt. Es werden für die einzelnen Artefakte keine Architektur-Topologien sowie Unified Modeling Language (UML) Diagramme erstellt, sondern die Umsetzung mittels natürlichsprachigem Text in Prosaform dargelegt. Der konkrete Vorgang in der Phase Coding wird in den jeweiligen Unterkapiteln der einzelnen Artefakte erklärt, da sich die Softwareerstellung je nach Anforderung stark unterscheidet.

Die Phase Testing des Wasserfallmodells erfolgt einerseits durch Debuggen des vorhandenen Codes. Dabei werden Compilefehler, Laufzeitfehler und etwaige grafische Visualisierungsfehler behoben. Für eine adäquate Simulation in der Testumgebung werden fiktive Energiedaten durch einfache mathematische Operationen generiert. Im Zuge des Design Science wird ebenfalls eine Evaluation der Artefakte durch vier ExpertInnen unter realen Bedingungen durchgeführt. Nach Freigabe der Artefakte durch die Stakeholder werden diese in den ACC Store geladen und somit für alle KundInnen der evon GmbH zur Verfügung gestellt.

Um den Umfang der Masterarbeit einzuschränken, wird die Phase Maintenance außer Acht gelassen und in weiterer Folge durch den Autor im Zuge seiner beruflichen Tätigkeit durchgeführt. Nachfolgend wird die Umsetzung der Anforderungen für das Erreichen der Marktkonformität des EMSControl in einzelnen Unterkapiteln erläutert. Die dazugehörigen Programmcodes werden im Anhang C dargestellt.

5.3.1 Online-Wetterdaten

Für die Witterungsbereinigungen der Energiedaten werden die Außentemperaturen zu den jeweiligen Aufzeichnungszeitpunkten benötigt. Diese Daten können über Hardwaretemperatursensoren wie u. a. PT1000 Sensoren an Gebäudefassaden erhoben und in das EMSControl über entsprechende Feldbussysteme übernommen werden. Ist dies bauseits nicht gewünscht oder realisierbar, soll es den AnwenderInnen des EMSControl ermöglicht werden, Temperaturen über Online-Wetterdaten in das System übernehmen zu können.

Online-Wetterdaten können über sogenannte Application Programming Interfaces (API) abgerufen werden. Eine API ist eine Programmierschnittstelle, welche es ermöglicht, zwei unabhängige Softwarekomponenten miteinander zu verknüpfen, um dadurch Informationen auszutauschen. Dabei stellt eine Software die API zur Verfügung, worauf die andere Software zugreifen kann. Idealerweise wird bei der Bereitstellung einer API eine ausführliche Dokumentation veröffentlicht, welche beschreibt, wie eine Kommunikation mit der Gegenstelle aufgebaut werden kann (Geißler & Ostler, 2018; Luber, 2017; Slate, 2019). Durch Anwendung einer API besteht die Möglichkeit des Outsourcings von Programmierarbeiten. Hierbei werden Teilbereiche einer Software mit überschaubarem Aufwand ausgelagert, wodurch sich die Entwickler Zeit sparen, da sie ihre Applikation mit bereits vorhandenen Tools verbinden können. Dadurch wird die Notwendigkeit zur Erstellung von Funktionen reduziert, welche von anderen Unternehmen bereits entwickelt wurden. APIs werden vor allem verwendet, um mit Webdiensten interagieren zu können (Geißler & Ostler, 2018; Luber, 2017). Knapp 70 Prozent der Web-APIs verwenden aufgrund der Skalierbarkeit den Architekturstil Representational State Transfer (REST). Dabei wird für die Zugriffe auf das HTTP Kommunikationsprotokoll zurückgegriffen, um mittels PUT-, GET-, POST- und DELETE-Methoden auf Daten zugreifen zu können. Die Repräsentation der angeforderten Ressource erfolgt beispielsweise in den Formaten HTML, JSON oder XML (Hunsaker, 2015; Richards, 2006, S. 633–637).

Nach einer Vergleichsrecherche zur Auswahl einer geeigneten API für das Auslesen von Wetterdaten wurde vom Autor die OpenWeatherMap API von OpenWeather ausgewählt. Die API kann für kommerzielle Zwecke verwendet werden. Zudem existiert eine kostenfreie Version, welche für den Anwendungsfall alle notwendigen Daten und Informationen bereitstellt. Dabei werden sowohl aktuelle Wetterdaten als auch ein 5-Tages-Forecast in 3-Stunden-Schritten zur Verfügung gestellt. Es ist allerdings zu

beachten, dass bei der freien Variante die Anfragen pro Tag auf 1.000 begrenzt sind. Soll in Zukunft dieses Softwarekonzept weiter skaliert werden, muss auf einen kostenpflichtigen Account gewechselt werden. Beim kostenpflichtigen Account kann einerseits eine höhere Anzahl an Anfragen durchgeführt werden. Andererseits besteht auch die Möglichkeit detailliertere Wetterdaten abzurufen, welche allerdings im konkreten Anwendungsfall nicht benötigt werden (OpenWeather, 2021a).

Die Schnittstelle zum Webservice wird als REST API zur Verfügung gestellt. Die Anfragen an den Service werden über das HTTP-Protokoll mit der GET-Methode durchgeführt und die Antwort der API erfolgt im JSON-Format. Für eine gültige Anfrage muss zuerst eine Registrierung bei OpenWeather durchgeführt werden, um anschließend den sogenannten API Key zu erhalten. Durch diesen Schlüssel werden die anfragenden UserInnen eindeutig identifiziert und die Berechtigungen überprüft. Dadurch kann ein Missbrauch der betreffenden API verhindert werden. Ferner muss für eine korrekte Anfrage der Ort, von welchem die Wetterdaten abgerufen werden sollen, in Klartext übermittelt werden (OpenWeather, 2021b). Der verwendete API Key wurde für Testzwecke auf OpenWeatherMap erstellt. Mit dieser Anfrage wird der 5-Tage-Wetterforecast abgefragt. Der Wetterforecast liefert neben den aktuellen Temperaturen auch Vorhersagewerte. Diese sind zwar für die Witterungsbereinigung nicht relevant, es wurde allerdings mit den Stakeholdern in der Besprechung vom 19.10.2021 definiert, dass diese ebenfalls implementiert werden, da sie für MSR-Anlagen hilfreich sind. Ferner ist zu erwähnen, dass bei der URL für den HTTP-Request der Parameter „units“ hinzugefügt wurde, um die Temperaturdaten in °C zu erhalten (OpenWeather, 2021c).

Damit diese Online-Wetterdaten zur weiteren Verwendung im EMSControl verwendet werden können, wurde ein eigenes ACC entwickelt. Die Programmierung erfolgte im XAMControl dabei über C#-Methoden. Es wurde darauf geachtet, dass die Methoden, in welchen die HTTP-Requests durchgeführt werden, asynchron programmiert wurden, um den Realtime-Zyklus der Soft-SPS nicht zu beeinflussen und eine adäquate Performance des Systems zu gewährleisten. Über einen Trigger werden im 15 Minuten-Intervall die Daten abgefragt und automatisch über die Trendkonfiguration im XAMControl archiviert. Der Ort sowie der eindeutige API Key werden im ParameterPanel von den BedienerInnen eingetragen. Der Quellcodeauszug für das OpenWatherMap ACC befindet sich in Anhang C. Abbildung 21 zeigt eine Instanz des ACCs.

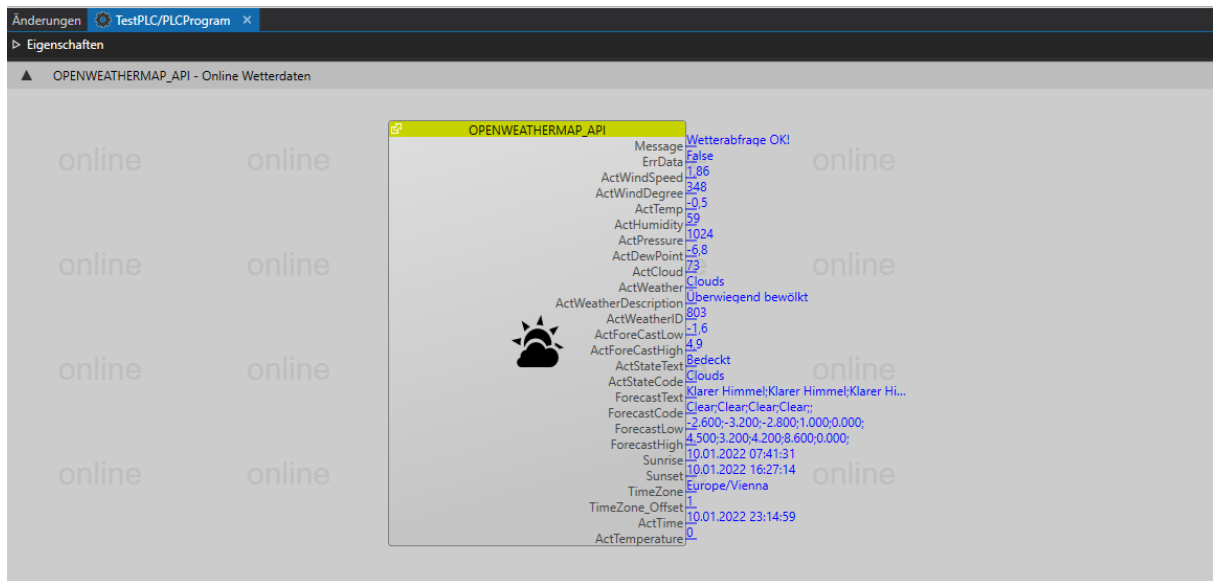


Abbildung 21: Instanz des Wetter ACCs (eigene Darstellung)

Abbildung 22 und Abbildung 23 stellen die vom Autor entworfenen grafischen Oberflächen der Wetterdaten dar.

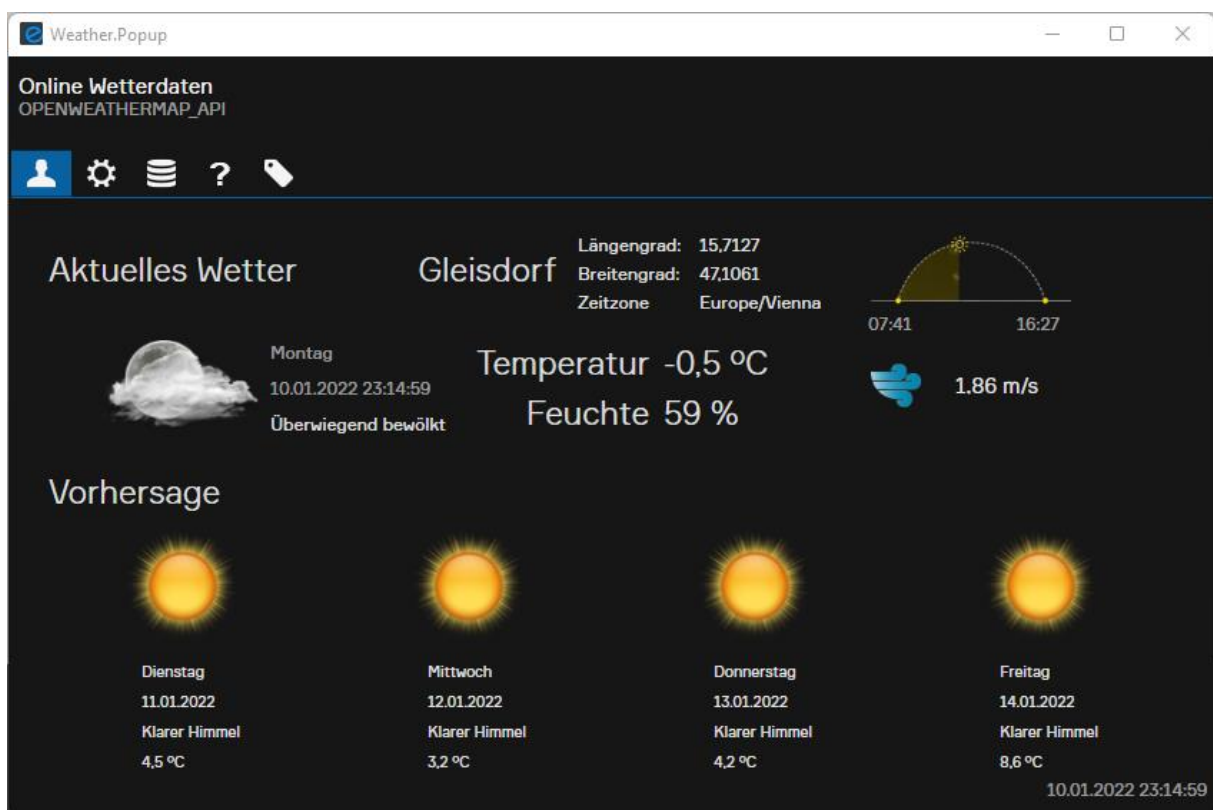


Abbildung 22: OperatorPanel des Wetter ACCs (eigene Darstellung)

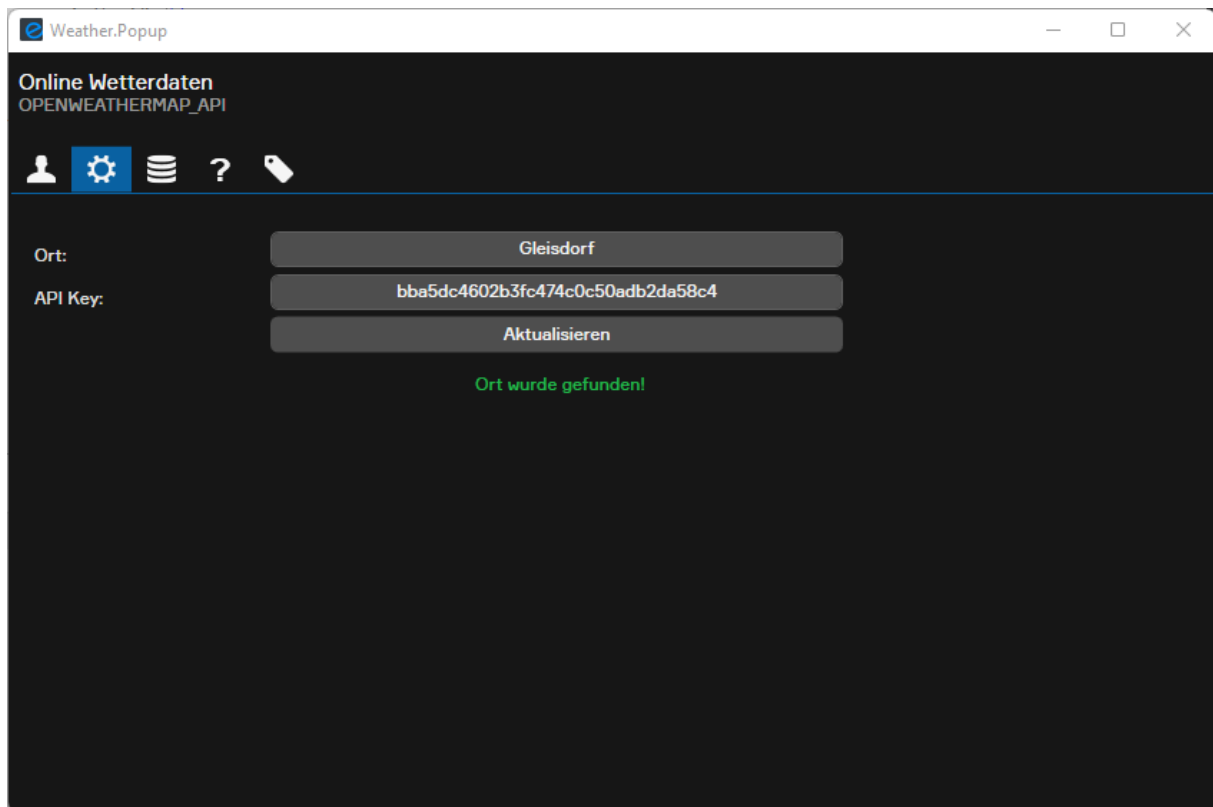


Abbildung 23: ParameterPanel des Wetter ACCs (eigene Darstellung)

5.3.2 Witterungsbereinigung

Die Witterung hat einen wesentlichen Einfluss auf die Höhe des Wärme- bzw. Kältebedarfs. Aufgrund dieser Feststellung können aufgezeichnete Verbrauchswerte nur bedingt miteinander verglichen werden. Um die Entwicklung des Energiebedarfs adäquat analysieren zu können, müssen die Werte über Klimakorrekturfaktoren korrigiert werden. Eine solche Normierung wird im Fachjargon als Witterungsbereinigung bezeichnet (DWD, o. D.).

Für die Berechnung der Klimakorrekturfaktoren werden sogenannte Gradtage verwendet. Diese stellen die Differenz zwischen der Raumtemperatur und der Außentemperatur dar. Heizgradtage werden berechnet, wenn die mittlere Außentemperatur niedriger als eine Grenztemperatur ist. In Österreich erfolgt die Berechnung der Heizgradtage anhand der Vorgaben der ÖNORM H 7500-3. Die Grenztemperatur ist hier mit 12 °C und die Raumtemperatur mit 20 °C definiert. In Deutschland werden die Heizgradtage anhand der VDI 2067 berechnet, welche den Heizgrenzwert mit 15 °C festlegt. Je höher der Wert der summierten Heizgradtage in einer betrachteten Periode ist, desto kälter war es während dieser Periode (Prettenthaler et al., 2007, S. 10–11). Die Berechnung der Heizgradtage wird in Formel 1 angegeben.

$$HGT = \sum_1^n (t_r - t_a) \text{ für } t_a < t_g$$

Formel 1: Berechnung Heizgradtage (eigene Darstellung)

Wobei t_r der angenommenen Raumtemperatur, t_a der Tagesmitteltemperatur, t_g der Grenztemperatur und n der Anzahl der Heiztage im Auswertzeitraum entsprechen.

Die Berechnung der Kühlgradtage erfolgt nach dem gleichen Prinzip, nur dass in diesem Fall die Außentemperatur den Grenzwert überschreiten muss. In Österreich liegt der Grenzwert hierbei bei 18,3 °C (Pretenthaler et al., 2007, S. 11–14). Die Berechnung der Kühlgradtage wird in Formel 2 angegeben.

$$KGT = \sum_1^n (t_r - t_a) \text{ für } t_a > t_g$$

Formel 2: Berechnung Kühlgradtage (eigene Darstellung)

Zur Veranschaulichung erfolgt demonstrativ eine Beispielberechnung der Heizgradtage über eine Periode von 5 Tagen sowie die Berechnung des Klimakorrekturefaktors. Die angestrebte Raumtemperatur wird mit 20 °C und die Heizgrenztemperatur mit 12 °C nach den Vorgaben der ÖNORM H 7500-3 festgelegt. Die berechneten Heizgradtage werden in Tabelle 11 dargestellt.

Tag	Minimaltemperatur	Maximaltemperatur	Tagesdurchschnitt	HGT
1	7,0 °C	18,0 °C	12,5 °C	0,0 °C
2	9,0 °C	22,0 °C	15,5 °C	0,0 °C
3	3,0 °C	13,0 °C	8,0 °C	12,0 °C
4	-2,0 °C	6,0 °C	2,0 °C	18,0 °C
5	0,0 °C	8,0 °C	4,0 °C	16,0 °C
Summe				46,0 °C

Tabelle 11: Beispielberechnung Heizgradtage (eigene Darstellung)

Somit ergibt sich für den berechneten Zeitraum von 5 Tagen eine Heizgradtagszahl von 46,0 °C. Damit können die klimatischen Verhältnisse verschiedenster Standorte oder Zeitbereiche verglichen und eine Klimabereinigung des Energieverbrauchs durchgeführt werden. Liegt er Referenzwert der Heizgradtage für diesen Zeitraum bei 50 °C, so kann ausgesagt werden, dass die Tage im Vergleich zum Referenzjahr relativ warm waren. Der Klimakorrekturefaktor berechnet sich durch Division der beiden Werte, wobei die Heizgradtage des Referenzjahres im Zähler und jene des betrachteten Jahres im Nenner stehen (DWD, o. D.).

$$\text{Klimakorrekturenfaktor} = \frac{HGT_{\text{Referenz}}}{HGT_{\text{Jahr}}}$$

Formel 3: Berechnung Klimakorrekturenfaktor (eigene Darstellung)

Für das oben genannte Beispiel würde die Berechnung einen Klimakorrekturenfaktor von 1,087 (= 50 °C / 46 °C) ergeben. Da das aktuelle Jahr wärmer war als das Referenzjahr, fiel der Heizbedarf geringer aus und der Verbrauch muss durch den Faktor 1,087 nach oben korrigiert werden, um entsprechende Vergleiche durchführen zu können.

Für die Berechnung der Gradtage im EMSControl wurde ein eigenes ACC entwickelt. Diesem ACC wird als Eingangsvariable die Außentemperatur übergeben, welche entweder über einen Hardwaretemperatursensor oder über die Online-Wetterdatenabfrage aufgezeichnet wird. Über FBD werden die minimale und maximale Außentemperatur des Tages definiert. Über C#-Methoden erfolgt die mathematische Berechnung der Heiz- und Kühlgradtage sowie die Archivierung der Daten in die Datenbank. Als zugehörige grafische Elemente wurde einerseits ein Popup mit einem OperatorPanel erstellt, in welchem die berechneten Werte abgebildet werden. Zudem können Reports geöffnet werden, um die Verbrauchswerte eines beliebigen Zählers zusammen mit den Heiz- oder Kühlgradtagen analysieren zu können. Auch ein Export der Werte ist über die Reports möglich, sodass mit vorhandenen Referenzwerten der Klimakorrekturenfaktor berechnet werden kann. Der Faktor kann wiederum bei der Generierung der Kennzahlen verwendet werden, was in Kapitel 5.3.11 näher erläutert wird. Ferner wurde ein ParameterPanel erstellt, in welchem die Benutzenden die Grenzwerte sowie die angestrebte Raumtemperatur für die Berechnungen parametrieren können. Der Quellcodeauszug für das Berechnen der Gradtage sowie ein demonstrativer SQL Query für den Jahresreport der Heizgradtage befinden sich in Anhang C. Nachfolgend werden die erstellten Panels und der Report dargestellt.

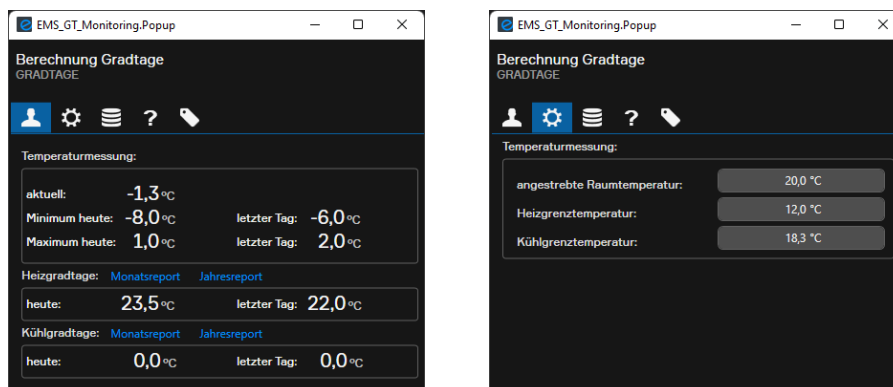


Abbildung 24: Popup des Gradtage ACCs (eigene Darstellung)

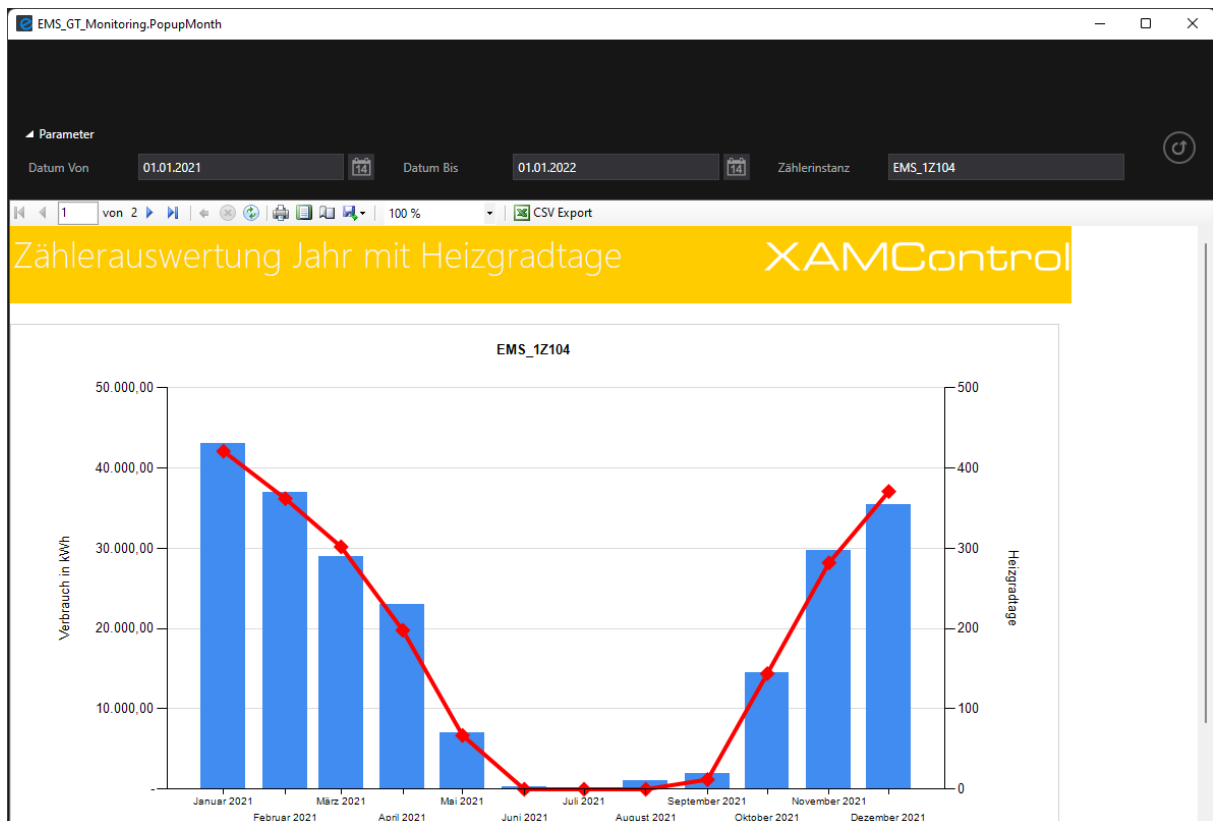


Abbildung 25: Report Heizgradtage (eigene Darstellung)

5.3.3 MSCONS

Zum elektronischen Geschäftsdatenaustausch wurde von den Vereinten Nationen ein branchenübergreifender Nachrichtenstandard namens Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) geschaffen. Durch den Standard werden Geschäftsprozesse zwischen Unternehmen optimiert und können effizienter durchgeführt werden. EDIFACT wurde 1986 veröffentlicht und besteht aus ca. 200 einzelnen Nachrichtentypen und wird von rund 300.000 Unternehmen angewendet (Liegler, 2021; TUP, 2015). Nach dem Beschluss BK-06-009 vom 11.07.2006 der Beschlusskammer der deutschen Bundesnetzagentur ist bei den Geschäftsprozessen zur Anbahnung und zur Abwicklung der Netznutzung bei der Belieferung von LetztverbraucherInnen EDIFACT bei der Datenübertragung anzuwenden. Konkret wird unter Ziffer 3 gefordert, dass für den Bericht über die Lieferung von Daten zu Energiemengen der Nachrichtentyp Metered Services Consumption Report Message (MSCONS) angewendet wird (Bundesnetzagentur, 2006). Somit ist es für die Nutzenden des EMSControl von Vorteil, wenn die Energiedaten direkt in diesem Nachrichtentyp exportiert werden können. Auf Basis des MSCONS Anwendungshandbuches der Bundesnetzagentur wurde ein automatischer Export von Verbrauchsdaten der Energiezähler im EMSControl erstellt (Bundesnetzagentur, 2021b).

Um den Umfang dieser Masterarbeit einzuschränken, wird auf eine detaillierte Erklärung des Aufbaus von MSCONS verzichtet.

Die Umsetzung des Exports wurde über sogenannte Interfaces gelöst. Jedem Energiezähler im EMSControl wird ein Interface zugewiesen. Das Interface beinhaltet Variablen, welche für den MSCONS-Export relevant sind. U. a. sind das Verbrauchsdaten, Zählernummern, Zählerbeschreibungen und OBIS-Kennzahlen. Die Verbräuche werden über eine C#-Methode einem Double Array zugewiesen. Der dazugehörige Quellcodeauszug für die Zuweisung der Intervallverbräuche befindet sich in Anhang C. Die anderen Variablen sind als Parameter ausgeführt und müssen von den NutzerInnen entsprechend mit Werten befüllt werden. Dies geschieht über das ParameterPanel des jeweiligen Zählers. Abbildung 26 zeigt das in den Zählern implementierte Interface mit den beinhalteten Variablen. Abbildung 27 stellt das Konfigurationspopup dar.

NAME	DEKL	TYPE	WERT	BEMERKUNG	PRM	RET	S...	RO...	EXT	IO	TREIBER	ARR ANZAHL	ZUSÄTZLICHE TYPE DEFINITION	X-ATTRIBUT
Values_Interval	PLC	DoubleArr...	init values...	Verbräuche Intervall								0		
Values_Hour	PLC	DoubleArr...	init values...	Verbräuche Stunde								0		
Values_Day	PLC	DoubleArr...	init values...	Verbräuche Tag								0		
Values_Month	PLC	DoubleArr...	init values...	Verbräuche Monat								0		
Unit	PLC	String		Einheit des Mediums										
Status	PLC	Int	0	Status des Wertes										
OBIS	PLC	String		OBIS Kennzahl										
CounterDescription	PLC	String		Zählerpunktbeschreibung										
CounterNumber	PLC	String		Zählernummer										
Activate_Interval	PLC	String		Aktiviere Export Intervall										
Activate_Hour	PLC	String		Aktiviere Export Stunde										
Activate_Day	PLC	String		Aktiviere Export Tag										
Activate_Month	PLC	String		Aktiviere Export Monat										

Abbildung 26: Erstelltes Interface für MSCONS Export (eigene Darstellung)

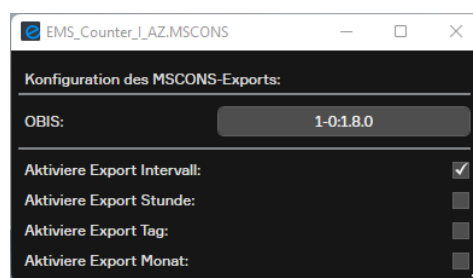


Abbildung 27: Erstelltes Konfigurationspopup für MSCONS Export (eigene Darstellung)

Ferner wurde ein ACC entwickelt, welches den Export der Zähler durchführt. Über eine GlobalInterfaceList werden die Daten der jeweiligen Energiezähler ausgelesen, daraus eine Textdatei im MSCONS-Format generiert und auf einem beliebigen Pfad abgespeichert. Der Quellcodeauszug für das Generieren der TXT-Datei im MSCONS-Format befindet sich in Anhang C. Abbildung 28 zeigt demonstrativ eine generierte MSCONS-Textdatei.

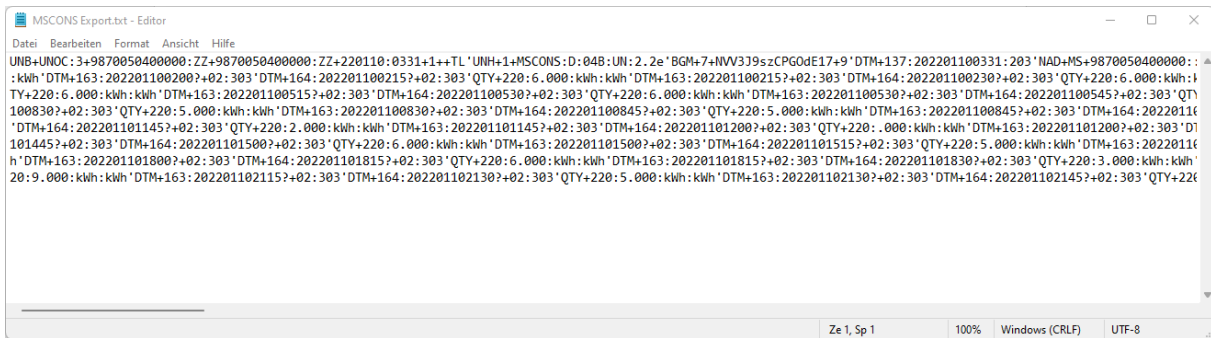


Abbildung 28: Beispiel MSCONS-Textdatei (eigene Darstellung)

Um das exportierte File und dahingehend die Programmierung auf Plausibilität überprüfen zu können, wurde die Open Source Software EDIFACT Viewer verwendet. Diese Software liest Textdateien im MSCONS-Format ein und stellt die Informationen grafisch dar. Es konnte festgestellt werden, dass die im XAMControl exportierten Textdateien problemlos von der Software interpretiert werden können und das MSCONS-Format somit richtig umgesetzt wurde. In Abbildung 29 werden die Daten der importierten Textdatei aus Abbildung 28 dargestellt. Die Software bildet dabei einen Mittelwert der Verbrauchswerte und stellt grafisch ausschließlich die Abweichungen über das Liniendiagramm dar. Des Weiteren werden die Werte in einer Tabelle abgebildet.

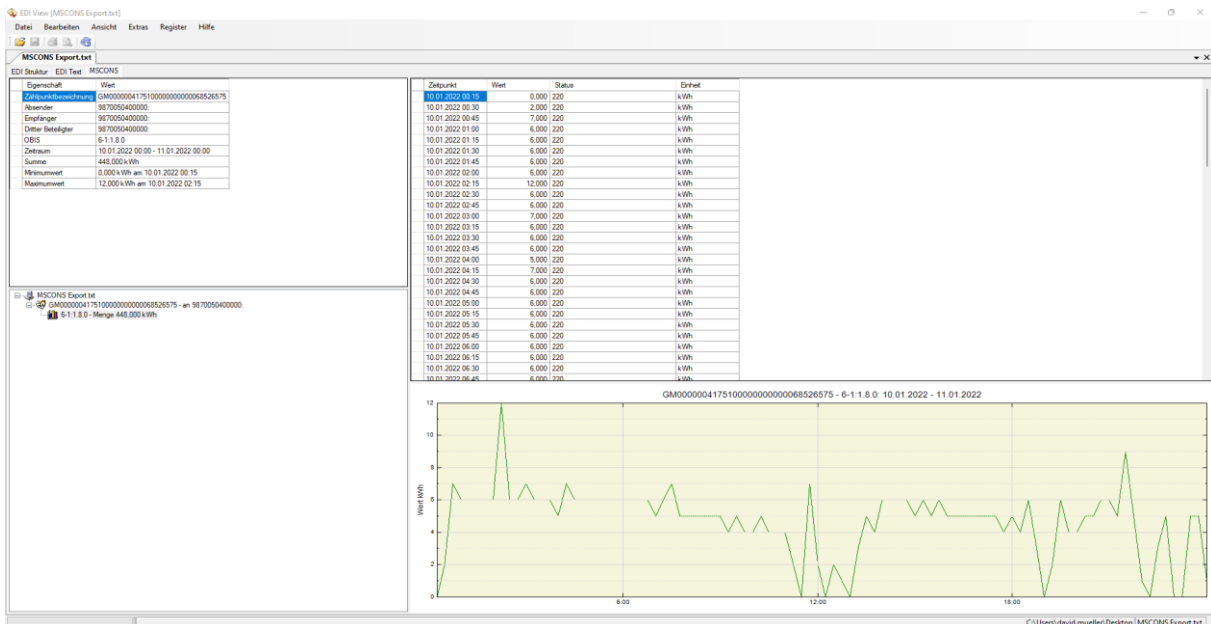


Abbildung 29: Import einer generierten MSCONS-Textdatei in den EDIFACT Viewer (eigene Darstellung)

5.3.4 Zählerreset

In der aktuellen Version des Zähler ACCs im EMSControl können die Zähler nur bedingt zurückgesetzt werden. Es werden hierbei ausschließlich die Livewerte durch Drücken eines Buttons genullt und die archivierten Energiedaten in der SQL-Datenbank bleiben bestehen. Dies ist für die BenutzerInnen unzufriedenstellend, da ein Reset aller Werte nur umständlich über direkten Eingriff in die Datenbank durchzuführen ist. Aus diesem Grund wurde im Zähler ACC eine C#-Methode für das Zurücksetzen des Zählers erstellt und es werden mithilfe eines SQL-Datenbank-Statements die Archivdaten gelöscht sowie die Live-Werte genullt. Der Datenbankzugriff wurde wiederum asynchron programmiert, um die Performance des Systems nicht zu gefährden. Der Quellcodeauszug für das Zurücksetzen der Zählerwerte befindet sich in Anhang C. Im Popup wurde im ParameterPanel ein Button zum Auslösen der Resetfunktion hinzugefügt.

5.3.5 Parametrierung der Einheiten

Derzeit kann im Popup des Zählers das Medium über ein Dropdown-Feld ausgewählt werden. Die Auswahl hat Einfluss auf die grafische Darstellung des Zählers in der Visualisierung, den Filtermöglichkeiten als auch auf die Einheit für die Daten. In Tabelle 12 werden die aktuellen Auswahlmöglichkeiten der Medien sowie die dazugehörigen Einheiten eines Zählers dargestellt.

Medium	Einheit für aktuelle Leistung	Einheit für gesamten Verbrauch
Gas	m ³ /h	m ³
Wärme	m ³ /h	m ³
Wasser	m ³ /h	m ³
Kraftstoff	l/h	l
Strom	kW	kWh
Wärmemenge	kW	kWh
Kältemenge	kW	kWh
Luft	m ³ /h	m ³

Tabelle 12: Medien und Einheiten im EMSControl (eigene Darstellung)

Es wurde bei der evon GmbH als Feedback angegeben, dass es gut wäre, wenn die Einheit von den Benutzenden selbst bestimmt werden könnte, da die Einheiten oft branchen- bzw. anlagenabhängig sind. U. a. werden für Wasserzähler standardmäßig die Einheiten m³ für den Gesamtverbrauch bzw. m³/h für den aktuellen Verbrauch verwendet. Jedoch wird beispielsweise in der Lebensmittelindustrie laut einem Stakeholder in l bzw. l/h gemessen. Aus diesem Grund wurde der Zähler im EMSControl adaptiert und ein Parameterfeld im

Popup hinzugefügt, sodass zwar die gleichen Default-Werte für die Einheiten hinterlegt sind, diese von den Benutzenden allerdings bei Bedarf abgeändert werden können. Nachfolgend wird der relevante Teilausschnitt des abgeänderten ParameterPanels grafisch abgebildet.

Standort:	Halle 1
Bereich:	HV
Name:	Warmwasser
Bezeichnung:	1Z102
Nummer:	3
Messung:	
Betrachtungszeitraum für Verbrauch:	15 Minuten
Zählerfaktor:	1,000 l/Impuls
Zählerwert Niedertarif:	0,000 K S Ü
Zählerwert Hochtarif:	0,000 K S Ü
Zeitverzögerung Kommunikation OK:	10 sec
Medium:	Wasser
Einheit für Verbrauch:	l
Einheit für aktuelle Leistung:	l/h

Abbildung 30: Parametrierfelder zur Einheitendefinition (eigene Darstellung)

5.3.6 Kostenstellenzuordnung

Im Rahmen des internen Rechnungswesens können die angefallenen Gemeinkosten – im Vergleich zu den Einzelkosten – den Kostenträgern, sprich Produkten, nicht direkt zugerechnet werden. Aus diesem Grund werden mithilfe der Kostenstellenrechnung die identifizierten Gemeinkosten, wie zum Beispiel Strom- oder Wasserkosten, sogenannten Kostenstellen verursachungsgerecht zugeordnet. Eine Kostenstelle stellt dabei einen Ort dar, an dem die Kosten entstehen. Kostenstellen können u. a. funktional oder räumlich gegliedert werden. Ferner wird zwischen Kostenstellen-Einzelkosten und Kostenstellen-Gemeinkosten unterschieden. Kostenstellen-Einzelkosten können Kostenstellen direkt zugerechnet werden, wohingegen Kostenstellen-Gemeinkosten nur indirekt mit einem Verteilungsschlüssel zurechenbar sind. Die angefallenen Kosten der Kostenstellen dienen in weiterer Folge zur Berechnung der Zuschlagssätze der Gemeinkosten und dahingehend zur verursachungsgerechten Kalkulation der Kostenträger nach Inanspruchnahme der Kostenstellen. Daneben liefern die Kostenstellen relevante Informationen aus den einzelnen Betriebsbereichen und ermöglichen eine wirksame Kostenkontrolle (Hering, 2015, S. 23–26; Nickenig, 2017, S. 73–75).

EDM-Softwaresysteme mit der Möglichkeit zur Definition von Kostenstellen bei den jeweiligen Zählern haben den Vorteil, dass Summenbildungen der einzelnen Zählerwerte, welche eine gemeinsame Kostenstelle besitzen, nicht per Hand durchgeführt werden müssen. Das Ziel in der EDM-Software ist, dass pro Zähler eine bzw. mehrere

Kostenstellen mit Verteilungsschlüsseln definiert werden können. Zudem muss den Benutzenden ein komfortabler Export der Verbrauchsdaten pro Kostenstelle ermöglicht werden. Der Autor dieser Masterarbeit hat zur Umsetzung dieser Anforderung ein Interface erstellt, in welchem fünf Kostenstellen und fünf zugehörige Umlageschlüssel parametriert werden können. Eine Erweiterung dieser Anzahl ist bei Bedarf problemlos möglich. Ferner wurden die Parametrierfelder in das Popup des EMS-Zählers hinzugefügt. Anschließend wurde ein eigenes ACC für die Summierung der Zählerwerte pro Kostenstelle erstellt. Für jede verwendete Kostenstelle im EMSControl wird eine Instanziierung von diesem ACC vorgenommen. Da hier ebenfalls als Parameter die Kostenstellenummer eingegeben wird, können über das Interface alle Verbrauchswerte dieser Kostenstelle summiert und in die SQL-Datenbank gespeichert werden. Ferner wurde berücksichtigt, dass einer Kostenstelle Verbrauchswerte von Zählern mit verschiedenen Medien zugerechnet werden können. Für die Visualisierung wurde ein Popup mit einem OperatorPanel erstellt, welches alle Verbrauchsdaten aller Medien der jeweiligen Kostenstelle darstellt. In Abbildung 31 werden der relevante Teilausschnitt des angepassten ParameterPanels sowie das neu erstellte Konfigurationspopup des Zähler ACCs dargestellt. Wie nachfolgend zu sehen ist, können für jeden Zähler bis zu 5 Kostenstellen sowie deren Verteilungsschlüssel definiert werden.

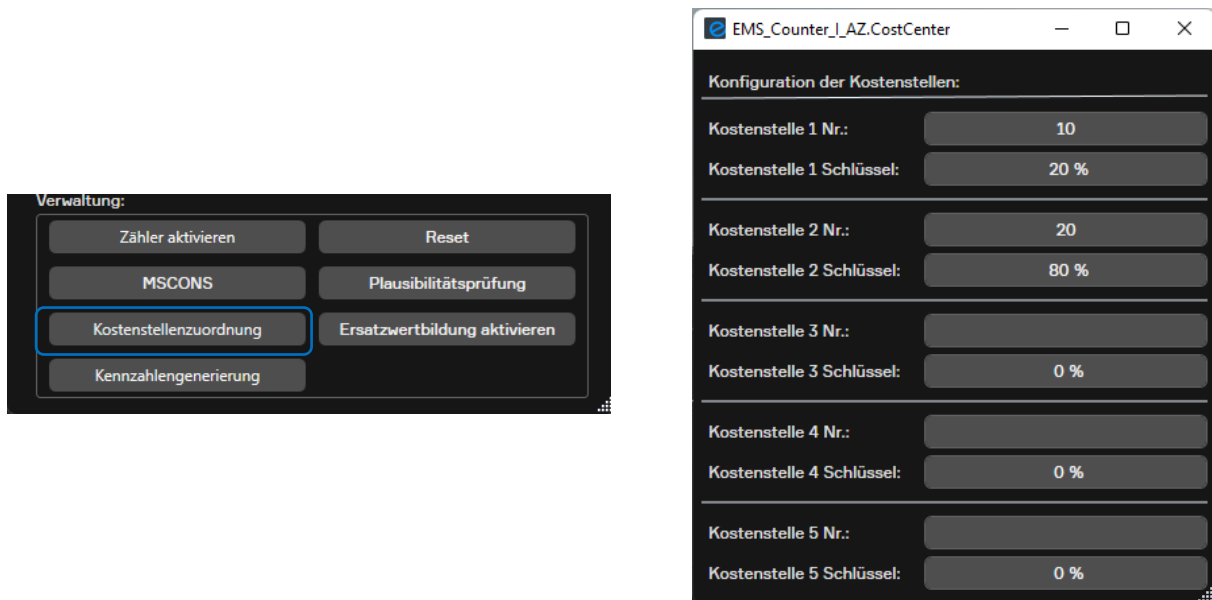


Abbildung 31: Parametrierfelder zur Kostenstellenkonfiguration (eigene Darstellung)

Abbildung 32 zeigt demonstrativ eine Kostenstelle mit den zugerechneten Verbrauchswerten. Es ist zu erkennen, dass dieser Kostenstelle Strom- als auch Wasserverbräuche von den Zählern zugerechnet wurden.

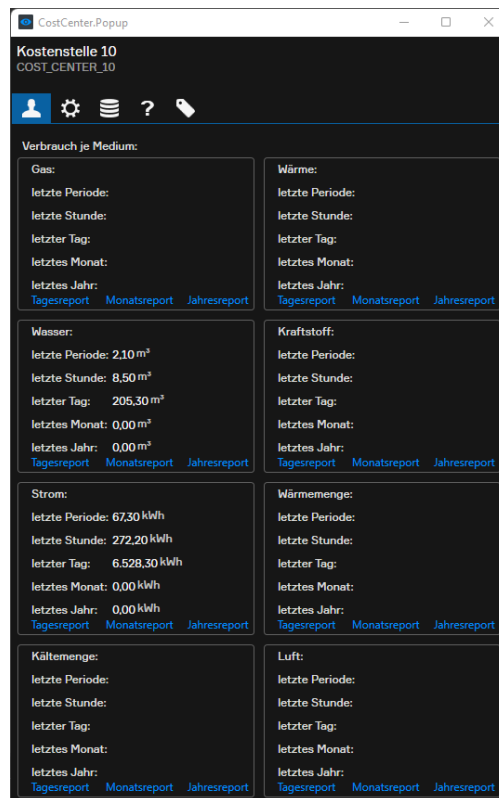


Abbildung 32: Popup des Kostenstellen ACCs (eigene Darstellung)

In Anhang C wird der Quellcodeauszug für die Berechnung der Verbrauchswerte der Kostenstellen dargestellt. Der Auszug beinhaltet ausschließlich die Berechnung der Intervallwerte für die Medien Gas und Luft, da die Berechnung der weiteren Werte ident erfolgt.

Hinsichtlich der Exportmöglichkeiten wurden die bereits vorhandenen Tages-, Monats- und Jahresreports aus den Zählern herangezogen und im Microsoft SQL Report Builder entsprechend abgeändert, sodass nun die Werte der jeweiligen Kostenstelle angezeigt werden. Über das Popup der Kostenstelle kann der jeweilige Report geöffnet werden. Ein Export der Daten in eine CSV-Datei ist hierbei ebenfalls möglich, sodass es einfach möglich ist, die Werte zur Berechnung der Zuschlagssätze und zur internen Leistungsverrechnung sowie zur Analyse der einzelnen Bereiche im Unternehmen zu verwenden. Abbildung 33 zeigt einen Tagesreport der Kostenstelle 10.

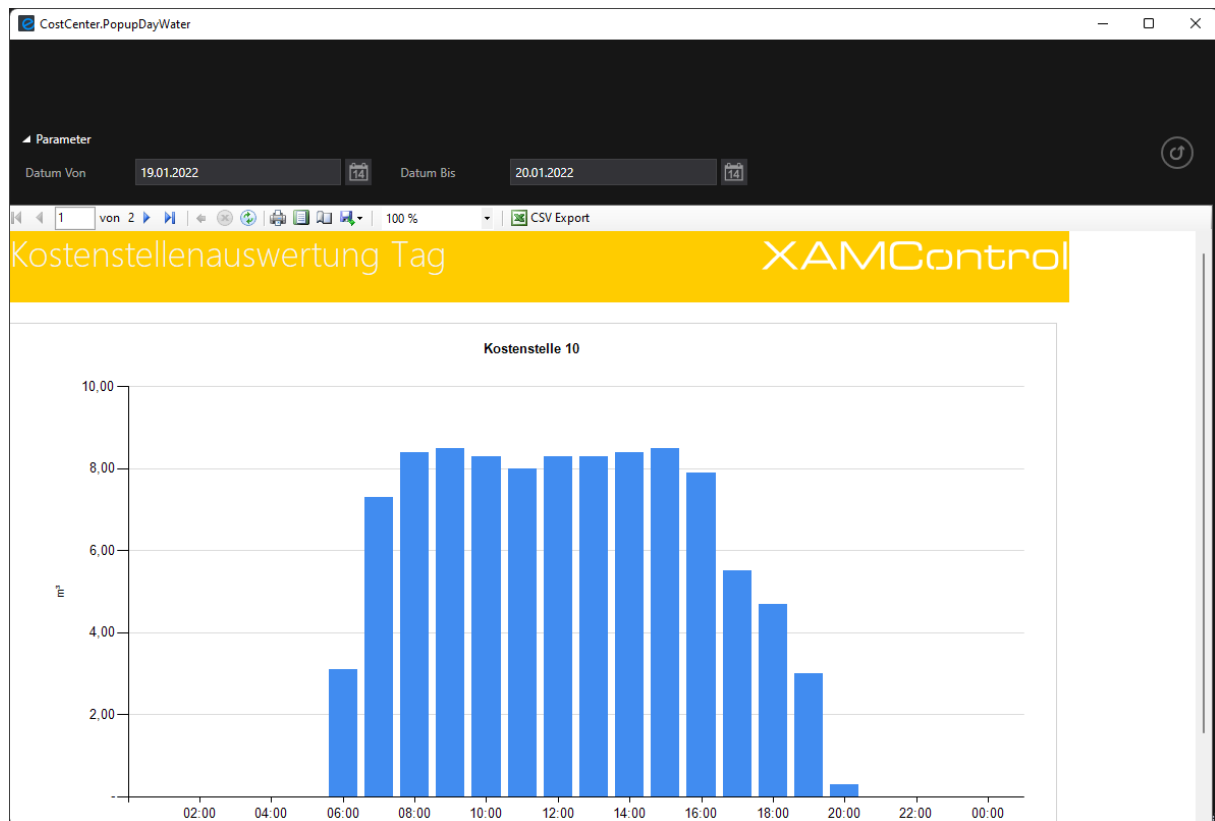


Abbildung 33: Tagesreport des Kostenstellen ACCs (eigene Darstellung)

5.3.7 Liniendiagramm

Die verfügbaren Reports im EMSControl waren in der bisherigen Version auf Balkendiagramme beschränkt. Aus den Anforderungen geht hervor, dass die Lastgänge auch mittels Liniendiagrammen dargestellt werden sollen. Aus diesem Grund wurden die drei Standardreports für Tages-, Monats- und Jahresauswertungen im Microsoft SQL Report Builder angepasst. Es wurde ein Liniendiagramm und ein weiterer Parameter für den Diagrammtyp hinzugefügt. Über ein Dropdown-Feld kann von den BenutzerInnen der Diagrammtyp ausgewählt werden. Damit nun der jeweilige Diagrammtyp angezeigt wird, wurde über eine SQL-Expression auf die Sichtbarkeitseigenschaft der Diagramme zugegriffen. Abbildung 34 und Abbildung 35 zeigen den neu angelegten Parameter „Diagrammtyp“ sowie die Expression zum Aktivieren der Sichtbarkeit des Liniendiagramms.

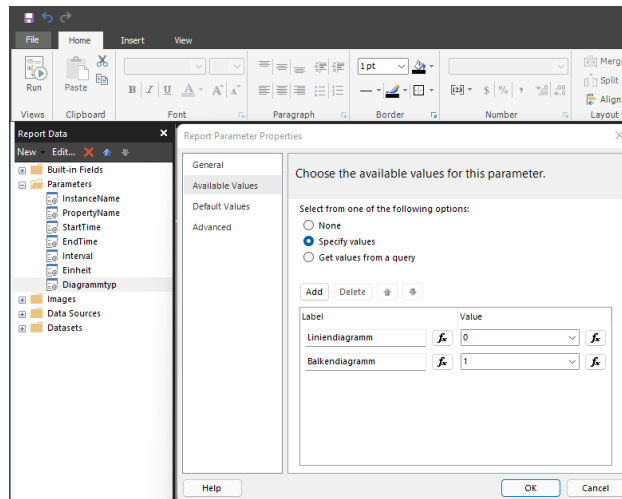


Abbildung 34: Parameter für dynamische Diagrammumschaltung (eigene Darstellung)

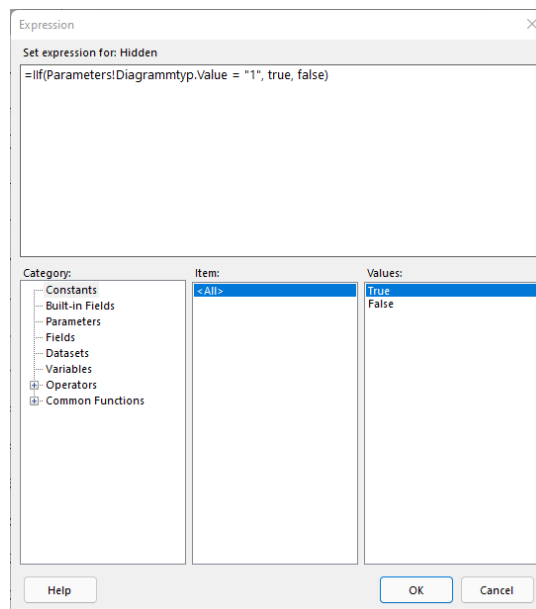


Abbildung 35: SQL Report Expression für dynamische Diagrammumschaltung (eigene Darstellung)

Weiters hat der Autor in den Tagesreports einen Parameter zur dynamischen Umschaltung der Verbrauchsdaten hinzugefügt. Es kann nun per Dropdown-Feld zwischen Intervall- und Stundenverbräuchen gewechselt werden. Die Umsetzung erfolgte ebenfalls über eine SQL-Expression. Abbildung 36 zeigt den über die Visualisierungsumgebung XAMIris geöffneten Report mit den neuen Auswahlfeldern inklusive Liniendiagramm.

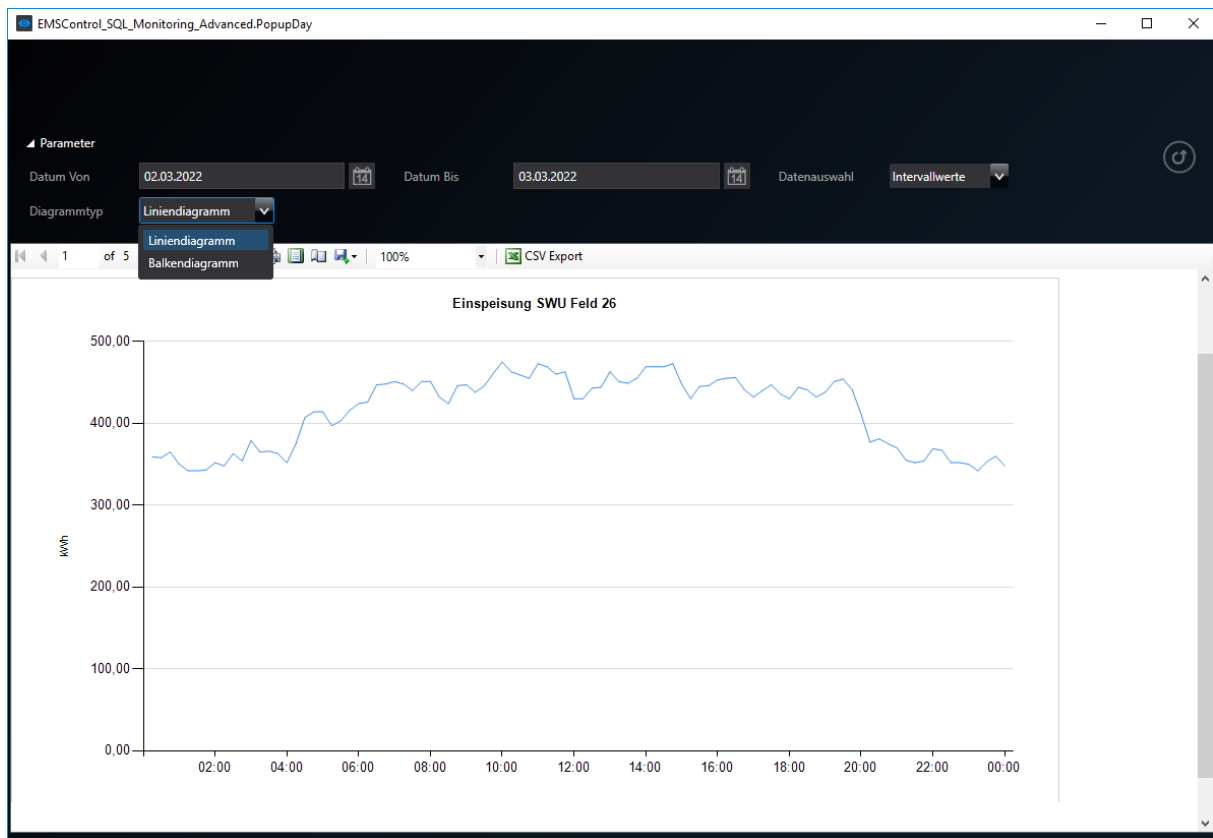


Abbildung 36: SQL Report mit dynamischer Diagrammumschaltung (eigene Darstellung)

5.3.8 Mittelwerte

Das BAFA fordert von den EDM-Systemen die Darstellung von Energiedaten in Form von Summen-, Mittel- und Extremwerten. Da die Zähler im EMSControl derzeit nur Summenbildungen durchführen sowie Extremwerte berechnen, liegt hier ein Handlungsbedarf vor. Aus diesem Grund wurde in das Programm des Zähler ACCs eingegriffen und Mittelwertbildungen der Verbrauchsberechnungen erstellt. Hierzu wurden C#-Methoden programmiert, welche auf die Archivdaten der SQL-Datenbank zugreifen und daraus die jeweiligen Mittelwerte bilden. Dabei werden Mittelwerte über die Intervallwerte des letzten Tages, Stundenwerte der letzten sieben Tage, Tageswerte des letzten Monats und Monatswerte des letzten Jahres berechnet. Demonstrativ zeigt der Quellcodeauszug in Anhang C die Mittelwertbildung der Intervallwerte der letzten Stunde.

Grafisch werden die berechneten Werte im OperatorPanel des Zählers angezeigt. Dieses wurde daher entsprechend angepasst. Abbildung 37 zeigt den Teilausschnitt des neuen OperatorPanels mit den Mittelwerten.

aktuell:	1.072,148 kW	
Periode:	1.071,952 kW	letzte Periode: 1.080,000 kW
Max Stunde:	1.076,486 kW	Max Tag: 1.076,537 kW
Max Monat:	1.076,537 kW	Max Jahr: 1.076,537 kW
Avg Interval:	267,000 kWh	Avg Stunde: 1.080,000 kWh
Avg Tag:	6.650,000 kWh	Avg Monat: 200.150,000 kWh

Abbildung 37: Grafische Darstellung der berechneten Mittelwerte (eigene Darstellung)

5.3.9 Automatische Ersatzwertbildung

Zur adäquaten Analyse der Energiedaten ist eine Datenintegrität essenziell. Datenintegrität bedeutet, dass die archivierten Daten vollständig, korrekt und widerspruchsfrei sein müssen (Appenzeller, 2021, S. 237). Da Smart Meter ihre Daten an die EDM-Software über diverse Schnittstellen übertragen, kann es bei Ausfällen des Netzwerks vorkommen, dass für den Zeitraum des Ausfalls keine Daten an die Software übermittelt werden. Zwar erkennen die Zähler des EMSControl einen Kommunikationsausfall, stoppen dadurch die Archivierung der Daten und vermeiden somit fehlerhafte Einträge, es entstehen dadurch allerdings Lücken in der Datenbank. Im Zeitraum des Netzwerkausfalls zählen die Smart Meter den Verbrauch der angeschlossenen Aggregate weiter und der absolute Zählerstand wird erhöht. Sobald wieder eine aufrechte Verbindung zur Software hergestellt ist, wird in der aktuellen Version des EMSControl der neue Zählerstand mit dem alten Zählerstand vor dem Ausfall verglichen und es entsteht durch die mathematische Operation der Subtraktion ein sogenannter Peak, welchen Abbildung 38 demonstrativ darstellt.

Datum	Uhrzeit	Verbrauch
19.01.2022	21:30:00	266,0 kWh
19.01.2022	21:15:00	266,0 kWh
19.01.2022	21:00:00	265,0 kWh
19.01.2022	20:45:00	263,0 kWh
19.01.2022	20:30:00	275,0 kWh
19.01.2022	20:15:00	269,0 kWh
19.01.2022	20:00:00	1360,0 kWh
19.01.2022	18:45:00	265,0 kWh
19.01.2022	18:30:00	270,0 kWh
19.01.2022	18:15:00	265,0 kWh
19.01.2022	18:00:00	263,0 kWh
19.01.2022	17:45:00	269,0 kWh
19.01.2022	17:30:00	267,0 kWh
19.01.2022	17:15:00	271,0 kWh
19.01.2022	17:00:00	270,0 kWh

Abbildung 38: Peak in den Energiedaten aufgrund eines Netzwerkausfalls (eigene Darstellung)

Um die Datenintegrität zu wahren, muss versucht werden, den Peak zu vermeiden und die fehlerhaften Lücken zu schließen. Dafür wurde das Zähler ACC entsprechend abgeändert. Durch ein SQL SELECT-Statement in einer C#-Methode wird abgefragt, welchen Zeitstempel der letzte Eintrag in der Datenbank aufweist. Dabei wird dieser Zeitstempel mit der aktuellen Zeit verglichen und es können Abweichungen festgestellt werden. Bevor der entstandene Peak in die Datenbank gespeichert wird, wird sein Wert durch die Anzahl der fehlenden Einträge dividiert, sodass eine gleichmäßige Aufteilung der Energiedaten stattfindet. Im beispielhaften Fall aus Abbildung 38 werden die 1360 m³ durch 5 dividiert, da 5 Einträge zwischen 18:45 und 20:00 fehlen, da auch der Peak um 20:00 noch nicht in die Datenbank gespeichert wurde. Nach der Überarbeitung des ACCs sehen die Datenbankeinträge nach dem Netzwerkausfall wie folgt aus.

Datum	Uhrzeit	Verbrauch
19.01.2022	21:30:00	266,0 kWh
19.01.2022	21:15:00	266,0 kWh
19.01.2022	21:00:00	265,0 kWh
19.01.2022	20:45:00	263,0 kWh
19.01.2022	20:30:00	275,0 kWh
19.01.2022	20:15:00	269,0 kWh
19.01.2022	20:00:00	272,0 kWh
19.01.2022	19:45:00	272,0 kWh
19.01.2022	19:30:00	272,0 kWh
19.01.2022	19:15:00	272,0 kWh
19.01.2022	19:00:00	272,0 kWh
19.01.2022	18:45:00	265,0 kWh
19.01.2022	18:30:00	270,0 kWh
19.01.2022	18:15:00	265,0 kWh

Abbildung 39: Ersatzwertbildung der Energiedaten nach einem Netzwerkausfall (eigene Darstellung)

Da es vorkommen kann, dass ein Netzwerkausfall genau dann auftritt, wenn die Stunden-, Tages- oder Monatswerte in die Datenbank gespeichert werden, wurde die Ersatzwertbildung auch bei diesen Werten implementiert. Des Weiteren wurde die Änderung so durchgeführt, dass die Funktion der automatischen Ersatzwertbildung für jeden Zähler über das ParameterPanel individuell aktiviert oder deaktiviert werden kann. Wird eine Korrektur durchgeführt, wird dies – aufgrund der Nachvollziehbarkeit der Werte – über eine Meldung in der Alarmliste der Visualisierungsumgebung XAMIRIS als auch über einen Logeintrag im XAMTraceViewer angezeigt. Demonstrativ wird in Anhang C der Quellcodeauszug für die Ersatzwertbildung der Intervallwerte dargestellt.

5.3.10 Automatische Plausibilitätsprüfung

Lastspitzen sind durch einen kurzfristigen starken Anstieg der nachgefragten Leistung nachvollziehbar, wenn es beispielsweise zu einem synchronen Hochfahren von großen Generatoren kommt. Allerdings kann es auch aufgrund von technischen Problemen bei den Smart Metern oder der Kommunikationsschnittstelle zur Aufzeichnung von fehlerhaften Werten kommen. Liegt eine hohe Abweichung zwischen den aufgezeichneten Energiedaten und den durchschnittlichen Werten vor, muss der Zähler genauer analysiert werden. In der aktuellen Version des EMSControl ist eine automatische Plausibilitätsprüfung nicht möglich. Derzeit kann ausschließlich manuell über einen Button im ParameterPanel des Zählers ein Eintrag erstellt werden, welcher besagt, dass dieser Zähler überprüft werden sollte. Der Autor dieser Masterarbeit hat sich zum Ziel gesetzt, dass die Zähler ACCs die Verbrauchswerte automatisch auf Plausibilität prüfen und bei festgestellten Abweichungen zu den Mittelwerten einen Logeintrag erstellen bzw. eine Warnmeldung ausgeben.

Für die Umsetzung konnte auf die bereits vom Autor programmierte Mittelwertbildung der Zählerdaten zurückgegriffen werden. Es wurde im ParameterPanel des Zählers ein Button implementiert, mit welchem ein Konfigurationspopup geöffnet werden kann. In diesem wird die automatische Plausibilitätsprüfung aktiviert und die akzeptierte Toleranz der Abweichung eingetragen. Des Weiteren gibt es die Möglichkeit, über das standardmäßige Zeitschaltprogramm von XAMControl, Betriebszeiten zu definieren. Dadurch kann verhindert werden, dass außerhalb der Betriebszeiten die Plausibilitätsprüfung anschlägt, wenn der Verbrauchswert null ist. Nachfolgend werden die neu erstellten Parametrierfelder dargestellt.

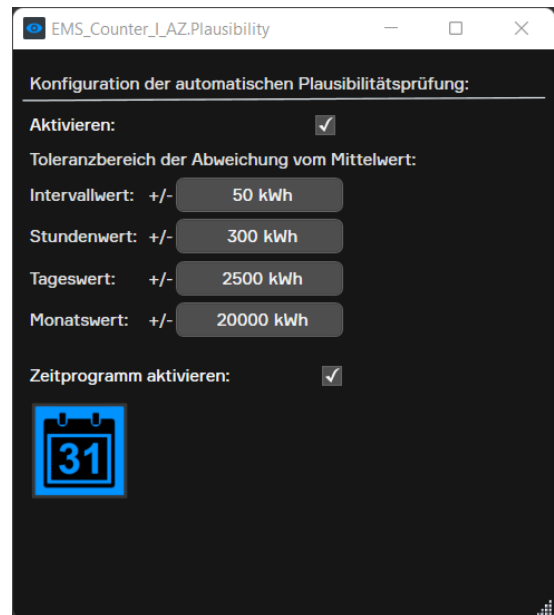
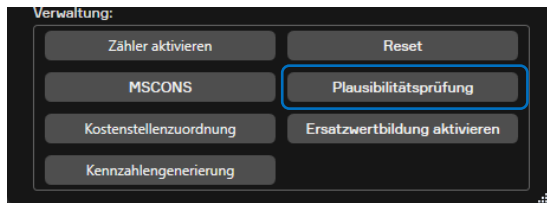


Abbildung 40: Parametrierfelder zur Konfiguration der Plausibilitätsprüfung (eigene Darstellung)

Ist der neu berechnete Verbrauchswert höher bzw. niedriger als der Mittelwert multipliziert bzw. dividiert durch den definierten Faktor, so wird der Verbrauchswert zwar in die Datenbank gespeichert, allerdings eine Warnmeldung in der Visualisierungsumgebung XAMiris ausgegeben und zusätzlich ein Logeintrag im XAMTraceViewer generiert. Die automatische Plausibilitätsprüfung wurde bei dem Zähler ACC über C#-Methoden implementiert. In Anhang C befindet sich demonstrativ der Quellcodeauszug für die Plausibilitätsprüfung der Intervalldaten. Abbildung 41 zeigt einen Logeintrag der Plausibilitätsprüfung.

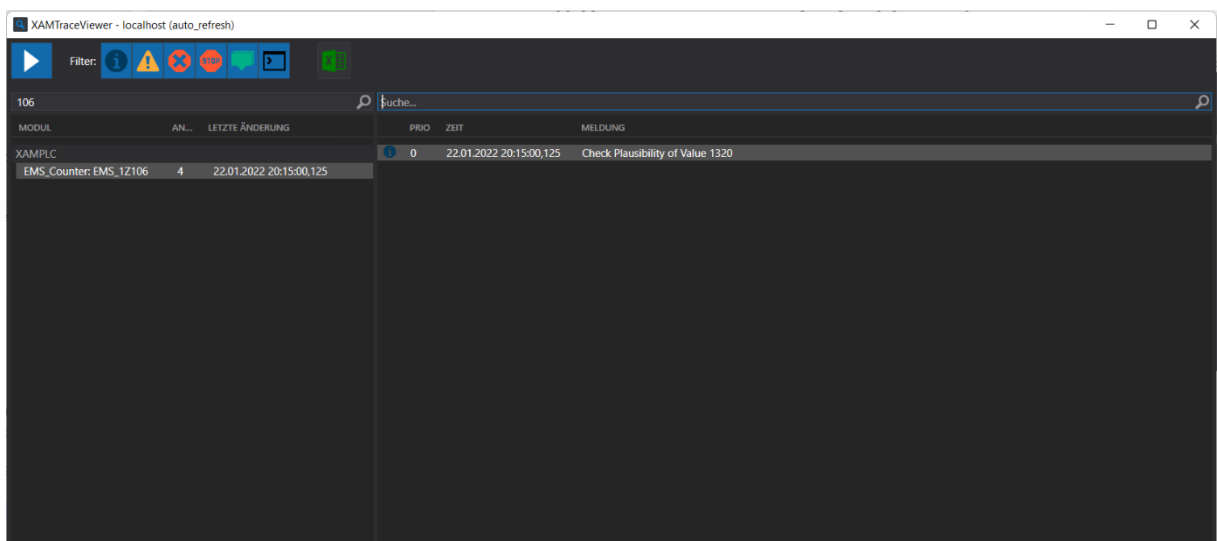


Abbildung 41: Logeintrag der Plausibilitätsprüfung (eigene Darstellung)

5.3.11 Kennzahlengenerierung und -darstellung

Gemäß der ISO 50001 müssen – wie in Kapitel 2.5.2 bereits erwähnt – geeignete EnPIs festgelegt werden, um die energetische Leistung zu messen. Durch die EnPIs sollen Unternehmen dabei unterstützt werden, die Ziele zu erreichen sowie Probleme und Auswirkungen von Effizienzmaßnahmen zu erkennen. Auch energetische Zielvorgaben können von Unternehmen in Form von EnPIs dargestellt werden. Die Kennzahlen können in Form von Verhältniszahlen, wie kWh/produzierte Einheiten oder m³/Anzahl der MitarbeiterInnen oder durch statistische Modelle wie linearer Regression ausgedrückt werden, bei welchen die Abweichungen zwischen prognostizierten und tatsächlichen Verbrauchswerten miteinander verglichen werden (BMLRT, 2018).

Zur Umsetzung dieser Anforderung im EMSControl wurde die Möglichkeit geschaffen, die Kennzahlen durch Integration von Bezugsgrößen generieren zu können. Dazu können beliebige Messwerte, wie zum Beispiel Produktionseinheiten oder Umsatzzahlen über einen CSV-Importer in die Datenbank eingespielt werden. Des Weiteren hat der Autor die Möglichkeit geschaffen, die importierten Daten wiederum exportieren zu können, sodass im Anschluss eine Abänderung problemlos möglich ist. Für die Umsetzung wurde ein Konfigurationspopup erstellt, in welchem die Art der Kennzahl definiert wird. Es kann hierbei zwischen drei Berechnungsarten gewählt werden. Wird die „konstante Berechnung“ ausgewählt, so werden die Verbrauchsdaten durch einen spezifischen Wert dividiert oder multipliziert. Bei Auswahl von „Messwerte“ werden die Verbräuche durch den Messwert des jeweiligen Zeitstempels dividiert oder multipliziert. Eine Multiplikation der Verbrauchswerte ist beispielsweise bei den Klimakorrekturfaktoren aus Kapitel 5.3.2 notwendig. Wird „Regression“ gewählt, muss die einfache lineare Regressionsgleichung eingegeben werden. Im Anschluss werden auf Basis der importierten Messwerte die prognostizierten Verbrauchswerte berechnet. Ferner kann im Konfigurationspopup über Tabs zwischen der Kennzahlengenerierung für Intervall-, Stunden-, Tages- und Monatswerten gewechselt werden. Abbildung 42 zeigt das Konfigurationspopup für die Kennzahlengenerierung.

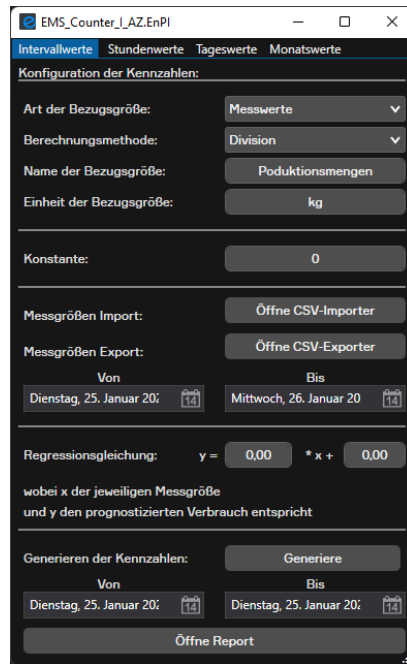


Abbildung 42: Popup für Kennzahlengenerierung (eigene Darstellung)

Zur Demonstration der Funktion wurde die Berechnungsart „Messwerte“ ausgewählt, sodass die aufgezeichneten Zählerverbräuche durch die jeweilige Messgröße desselben Zeitstempels dividiert werden. Es wurden dafür fiktive Messgrößen in das System über den programmierten CSV-Importer importiert. Einen Teilausschnitt der CSV-Datei sowie die importierten Daten in die SQL-Datenbank wird nachfolgend dargestellt.

	A	B	C
1	Value	DateTime	
2	6	26.01.2022 00:00:00 +01:00	
3	6	25.01.2022 23:45:00 +01:00	
4	6	25.01.2022 23:30:00 +01:00	
5	7	25.01.2022 23:15:00 +01:00	
6	7	25.01.2022 23:00:00 +01:00	
7	6	25.01.2022 22:45:00 +01:00	
8	6	25.01.2022 22:30:00 +01:00	
9	7	25.01.2022 22:15:00 +01:00	
10	6	25.01.2022 22:00:00 +01:00	
11	6	25.01.2022 21:45:00 +01:00	
12	6	25.01.2022 21:30:00 +01:00	
13	6	25.01.2022 21:15:00 +01:00	
14	6	25.01.2022 21:00:00 +01:00	
15	6	25.01.2022 20:45:00 +01:00	
16	6	25.01.2022 20:30:00 +01:00	
17	7	25.01.2022 20:15:00 +01:00	
18	6	25.01.2022 20:00:00 +01:00	
19	6	25.01.2022 19:45:00 +01:00	
20	6	25.01.2022 19:30:00 +01:00	
21	7	25.01.2022 19:15:00 +01:00	
22	6	25.01.2022 19:00:00 +01:00	
23	7	25.01.2022 18:45:00 +01:00	
24	7	25.01.2022 18:30:00 +01:00	
25	6	25.01.2022 18:15:00 +01:00	

	Intervalverbräuche	Messgröße	DateTime
1	273	6	2022-01-26 00:00:00.0000000 +01:00
2	268	6	2022-01-25 23:45:00.0000000 +01:00
3	266	6	2022-01-25 23:30:00.0000000 +01:00
4	268	7	2022-01-25 23:15:00.0000000 +01:00
5	275	7	2022-01-25 23:00:00.0000000 +01:00
6	270	6	2022-01-25 22:45:00.0000000 +01:00
7	274	6	2022-01-25 22:30:00.0000000 +01:00
8	263	7	2022-01-25 22:15:00.0000000 +01:00
9	269	6	2022-01-25 22:00:00.0000000 +01:00
10	273	6	2022-01-25 21:45:00.0000000 +01:00
11	265	6	2022-01-25 21:30:00.0000000 +01:00
12	266	6	2022-01-25 21:15:00.0000000 +01:00
13	274	6	2022-01-25 21:00:00.0000000 +01:00
14	268	6	2022-01-25 20:45:00.0000000 +01:00
15	263	6	2022-01-25 20:30:00.0000000 +01:00
16	269	7	2022-01-25 20:15:00.0000000 +01:00
17	266	6	2022-01-25 20:00:00.0000000 +01:00
18	267	6	2022-01-25 19:45:00.0000000 +01:00
19	266	6	2022-01-25 19:30:00.0000000 +01:00
20	275	7	2022-01-25 19:15:00.0000000 +01:00
21	271	6	2022-01-25 19:00:00.0000000 +01:00
22	271	7	2022-01-25 18:45:00.0000000 +01:00
23	264	7	2022-01-25 18:30:00.0000000 +01:00

Query executed successfully.

Abbildung 43: CSV-Messgrößen-Import (eigene Darstellung)

Durch das Generieren der Kennzahlen werden diese in die Datenbank gespeichert und können über einen Report grafisch angezeigt werden. Als Reportvorlage wurde der bereits im Rahmen dieser Masterarbeit erstellte Report mit Liniendiagramm herangezogen und entsprechend adaptiert. Abbildung 44 stellt den Report der EnPIs auf Basis der importierten Messwerte dar.

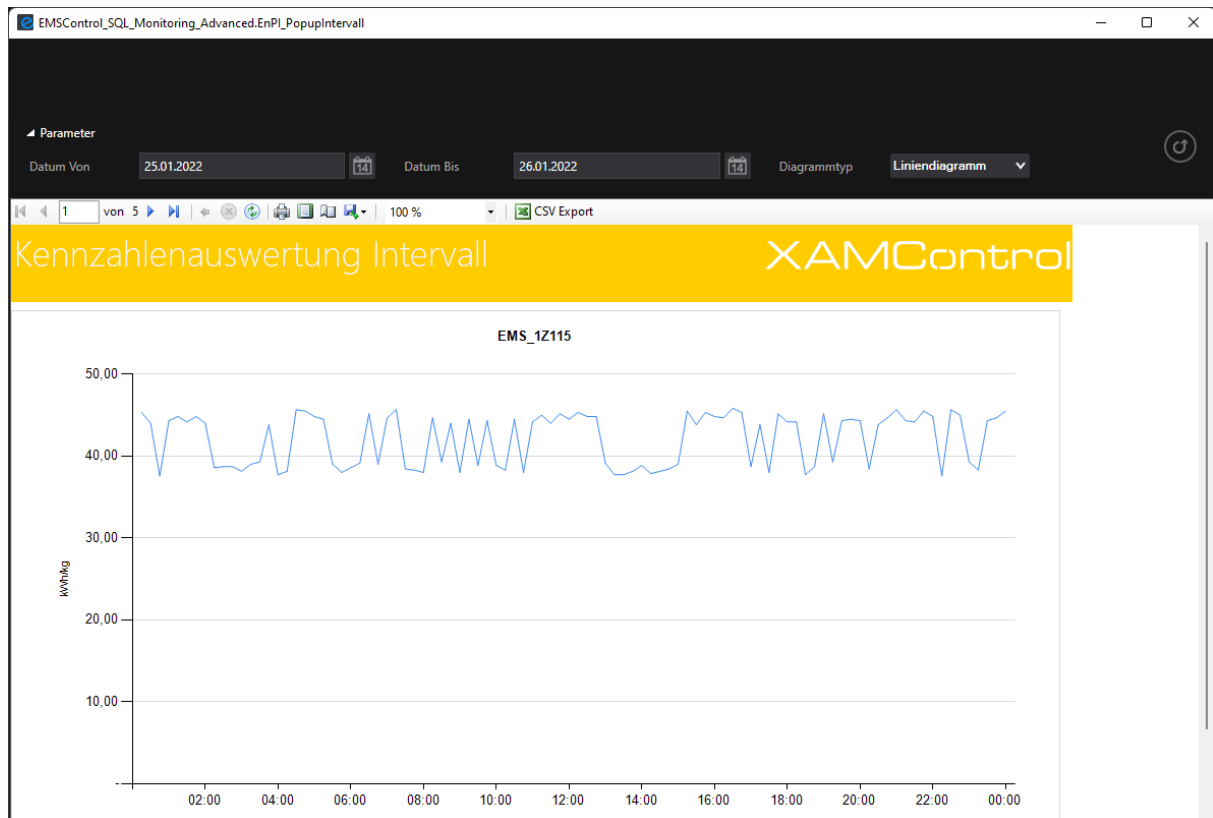


Abbildung 44: Report EnPI (eigene Darstellung)

Zur weiteren Veranschaulichung wird die Funktionsweise der einfachen linearen Regression demonstriert. Hierbei wird – unter der Annahme eines linearen Zusammenhangs – versucht, Werte einer abhängigen Variable mithilfe einer unabhängigen Variable vorherzusagen. Die Vorhersage ist dabei umso genauer, je größer der Zusammenhang zwischen der abhängigen und der unabhängigen Variable ist. Im Fall des EnMS kann versucht werden, anhand von einer statistischen linearen Regressionsanalyse der Verbrauchsdaten in Verbindung mit den Bezugsgrößen einen linearen Zusammenhang zu identifizieren. Beispielsweise ist es durchaus möglich, dass der Energieverbrauch umso höher ist, je mehr Produkte in einem Zeitraum produziert werden, da die Maschinen mit einem höheren Durchsatz bzw. einer höheren Drehzahl arbeiten. Grafisch wird der Zusammenhang zwischen der abhängigen und der unabhängigen Variable in einem sogenannten Streudiagramm dargestellt. Ziel der Regressionsanalyse ist es nun, jene Gerade zu bestimmen, welche den linearen

Zusammenhang am besten beschreibt. Dabei greift die Regressionsanalyse auf die Methode der kleinsten Quadrate zurück und versucht die Modellparameter a und b der Gerade so zu verändern, dass die Summe der quadrierten Abweichungen minimiert wird (Frost, 2018, S. 1–6; Kohn & Öztürk, 2017, S. 107–111; Weiß, 2019, S. 63–65). Die Formel der Regressionsgerade wird nachfolgend dargestellt.

$$\widehat{y}_k = b * x_k + a$$

Formel 4: Regressionsgerade (In Anlehnung an Frost, 2018, S. 9)

Wobei y_k der vorhergesagte Wert der abhängigen Variable, b die Steigung der Regressionsgeraden, x_k der Wert der unabhängigen Variable und a den Schnittpunkt der Geraden mit der y -Achse entsprechen.

Ferner ist zu erwähnen, dass die Regressionsgleichung zur Berechnung der prognostizierten y -Werte nur dann angewendet werden soll, wenn das Bestimmtheitsmaß ausreichend hoch ist. Diese Kennzahl beschreibt, welcher Anteil an der Varianz der abhängigen Variable durch die unabhängige Variable erklärt werden kann. Der Wert des Bestimmtheitsmaßes liegt stets zwischen 0 und 1. Ein Wert von 0,1 würde beispielsweise aussagen, dass ausschließlich 10 Prozent der Varianz der abhängigen Variable durch die gewählte unabhängige Variable erklärt werden kann. Der Vorhersagewert würde demnach ungenau ausfallen. Andersrum kann bei einem Bestimmtheitsmaß von 0,9 die abhängige Variable mithilfe der unabhängigen Variable gut vorhergesagt werden (Kohn & Öztürk, 2017, S. 129–131; Weiß, 2019, S. 65).

Um die Berechnung der prognostizierten Werte im EMSControl demonstrieren zu können, wurden fiktive Daten anhand von Intervallverbräuchen und Bezugswerten herangezogen. Um die Regressionsgleichung zu erhalten, wurde in Microsoft Excel ein Streudiagramm erstellt und die Regressionsgerade eingezeichnet, welches in Abbildung 45 dargestellt wird. Das dazugehörige Bestimmtheitsmaß beträgt 0,84. Somit können 84 Prozent der Varianz der Intervallverbräuche mit der Anzahl der Produktionseinheiten erklärt werden.

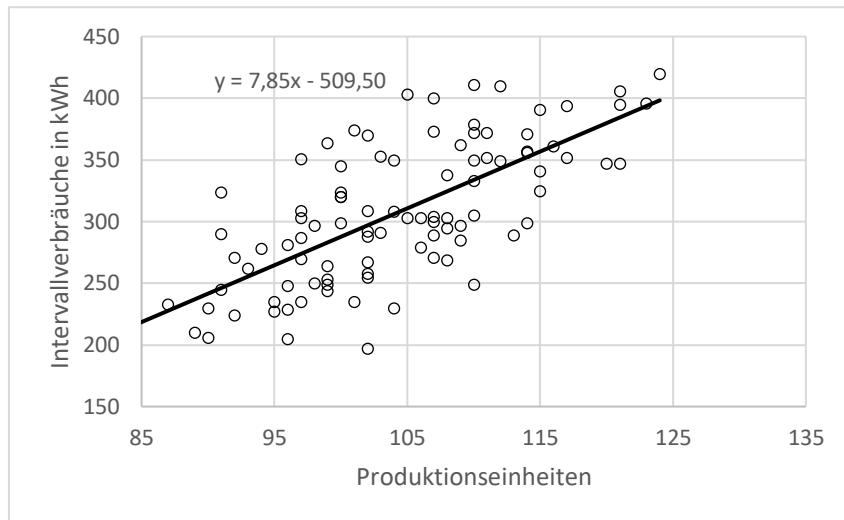


Abbildung 45: Streudiagramm mit Regressionsgerade (eigene Darstellung)

Die Modellparameter b (7,85) und a (-509,50) können nun im Konfigurationspopup eingetragen werden. Beim Generieren der EnPIs werden diese Parameter sowie die importierten Messwerte herangezogen und die y-Werte, sprich Intervallverbräuche, prognostiziert und in die SQL-Datenbank gespeichert.

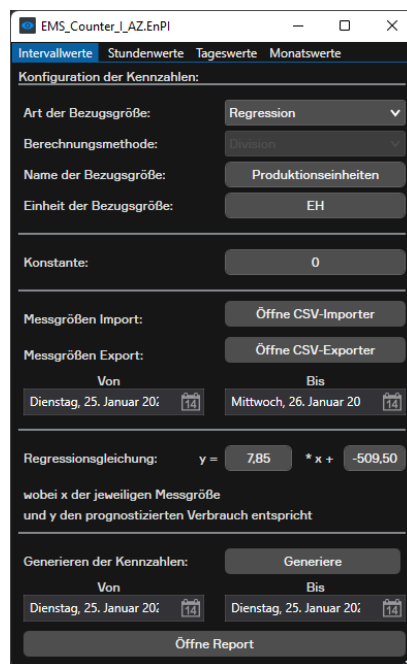


Abbildung 46: Regressionsgleichung im Konfigurationspopup (eigene Darstellung)

Es wurde ein eigener Report erstellt, welcher in einem Liniendiagramm die prognostizierten und die tatsächlichen Verbräuche gegenüberstellt. Demonstrativ wird dieser Report mit den fiktiven Daten in Abbildung 47 dargestellt. Wie in diesem Beispiel zu sehen ist, stimmen die prognostizierten und tatsächlichen Verbrauchswerte nahezu überein.

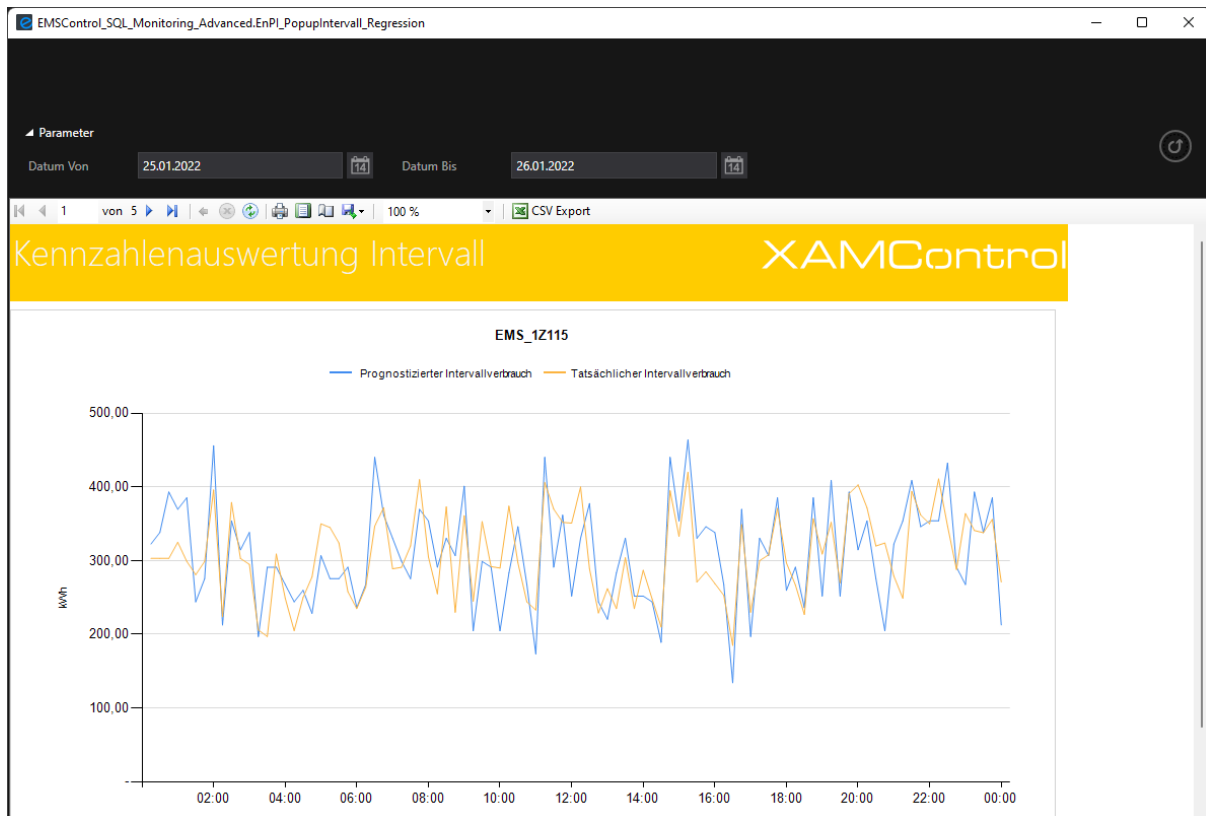


Abbildung 47: Report EnPI Regression (eigene Darstellung)

In Anhang C werden die Quellcodeauszüge der jeweiligen C#-Methoden zum Import und Export der Messgrößen, zur Berechnung der EnPIs und zur Auswahl des Reports dargestellt.

5.4 Evaluation

Ein wesentlicher Aspekt im Design Science ist die adäquate Evaluierung der erstellten Artefakte. Dabei wird geprüft, ob die Artefakte auch außerhalb der Simulationsumgebungen funktionieren und demnach die Anforderungen praxistauglich erfüllen (Baskerville et al., 2018, S. 362; Benner-Wickner et al., 2020, S. 7–8). Da es sich bei den entwickelten Artefakten um Instanziierungen handelt, hat sich der Autor als Evaluierungsmethode aufgrund von Empfehlungen aus der Literatur für die

ExpertInneninterviews entschieden (Sonnenberg & vom Brocke, 2012, S. 393–395). Nachfolgend werden die einzelnen Schritte und Ergebnisse der Interviews erläutert.

5.4.1 Methodische Vorgehensweise

Ziel der ExpertInneninterviews war die Verifikation der erstellten Artefakte, um festzustellen, ob diese für die Praxis geeignet sind und veröffentlicht werden können. Dazu wurden die vier Stakeholder, welche bereits beim Requirements Engineering als ExpertInnen involviert waren, für die Befragung ausgewählt. Da diese Personen im Bereich des EnMS eine hohe Kompetenz aufweisen und das EMSControl in ihrer täglichen Arbeit verwenden, werden sie vom Autor als ideale InterviewpartnerInnen angesehen. Die Befragungen wurden anhand eines Interviewleitfadens durchgeführt, welcher aus 12 Fragen bestand. Dieser wird in Anhang D dargestellt. Die Interviews wurden aufgrund der geografischen Entfernung zwischen Autor und ExpertInnen online durchgeführt. Die Transkription der Interviews erfolgte nach inhaltlich-semantischer Form. Dabei wird wörtlich, allerdings nicht lautsprachlich transkribiert. Dialekte werden ins Hochdeutsche übersetzt und Wortverschleifungen dem Schriftdeutsch angenähert. Zudem werden Stottern, abgebrochene Sätze, Wortdoppelungen und Rezeptionssignale nicht transkribiert (Dresing & Pehl, 2018, S. 21–23). Die transkribierten Interviews befinden sich in Anhang D. Die qualitative Inhaltsanalyse wurde an den Prozess nach MAYRING angelehnt. Zudem wurden die ExpertInnen angehalten, ihre Meinungen zu den jeweiligen Artefakten kurz und prägnant wiederzugeben. Aus diesem Grund erfolgen die Paraphrasierung und Generalisierung der relevanten Textstellen in einem Schritt. Da sich das Ziel der Interviews ausschließlich auf die Validierung der Artefakte bezogen hat, erfolgt die Kategorisierung der Ergebnisse nach dem jeweiligen Artefakt (Mayring, 2015, S. 70).

5.4.2 Qualitative Auswertung der ExpertInneninterviews

In diesem Abschnitt werden die Interviews qualitativ untersucht, um daraus die Qualität der entwickelten Artefakte ableiten zu können. Die transkribierten Interviews wurden generalisiert, um damit die Kernaussagen der ExpertInnen zusammenzufassen. Diese Ergebnisse werden in der nachfolgenden Tabelle 13 dargestellt. Die Zeilen der Tabelle entsprechen der jeweiligen Frage aus dem Interviewleitfaden. In den Spalten befinden sich die Kernaussagen der ExpertInnen.

Frage Nr.	Interview 1	Interview 2	Interview 3	Interview 4	Zusammengefasst
1	Adäquat umgesetzt	Adäquat umgesetzt; Idee für Regelungen auf Basis der Vorhersagewerte; Meeting gewünscht	Funktion gegeben, aber Probleme mit Schreibweise des Ortes (nicht beeinflussbar)	Adäquat umgesetzt	Adäquat umgesetzt
2	Nicht getestet, daher kein Feedback	Adäquat umgesetzt	Nicht getestet, daher kein Feedback	Nicht getestet, daher kein Feedback	Adäquat umgesetzt allerdings wenig getestet
3	Adäquat umgesetzt	Bis dato adäquat umgesetzt; Vergleich mit Onlinedaten in ein paar Wochen	Bis dato adäquat umgesetzt; Umstieg von kostenpflichtigen Daten, wenn die Werte beider Varianten ähnlich sind	Adäquat umgesetzt	Adäquat umgesetzt; weiteres Feedback in ein paar Wochen
4	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
5	Adäquat umgesetzt	Nicht getestet, daher kein Feedback	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
6	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
7	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
8	Funktion gegeben, aber Zeitbereich sollte beliebig definierbar sein	Funktion gegeben, aber Zeitbereich sollte beliebig definierbar sein	Adäquat umgesetzt	Funktion gegeben, aber Zeitbereich sollte beliebig definierbar sein	Funktion gegeben; Zeitbereich beliebig definierbar machen
9	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
10	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt	Adäquat umgesetzt
11	Adäquat umgesetzt; abhängig von Mittelwerten, daher wichtig, dass diese angepasst werden können	Adäquat umgesetzt	Adäquat umgesetzt; Funktion sollte nur eingesetzt werden, wenn Werte nicht zu sehr schwanken	Adäquat umgesetzt; Aktivieren und Deaktivieren sehr wichtig	Adäquat umgesetzt
12	ISO Zertifizierung zeitnah durchführen	ISO Zertifizierung zeitnah durchführen	Keine Anmerkungen	ISO Zertifizierung zeitnah durchführen	ISO Zertifizierung zeitnah durchführen

Tabelle 13: Ergebnisse aus den ExpertInneninterviews (eigene Darstellung)

Wie Tabelle 13 zeigt, wurde die Umsetzung der Anforderungen von den ExpertInnen größtenteils als praxistauglich befunden. Eine Funktion sollte allerdings angepasst werden, um die Praxistauglichkeit zu wahren. Der Zeitbereich der Mittelwertbildung muss für die Benutzenden frei definierbar sein. Ebenfalls sehr wichtig ist den ExpertInnen eine aktuelle ISO 50001:2018 Zertifizierung. Diese sollte unbedingt noch im Jahr 2022 durchgeführt werden.

5.4.3 Weitere Iteration

Damit die neue Version des EMSControl veröffentlicht werden kann, müssen die Erkenntnisse aus der Evaluation der Artefakte wiederum in den iterativen Prozess des Design Science eingepflegt werden. Die Erkenntnisse stellen demnach wieder Anforderungen dar, welche in der Entwurfsphase in das System implementiert werden.

Wie bereits in Kapitel 5.2 erwähnt, ist die ISO 50001:2018 Zertifizierung nicht Teil dieser Masterarbeit. Somit wird ausschließlich die, in den Interviews identifizierte Anforderung für die Mittelwertbildung umgesetzt, sodass die Zeitbereiche für die Berechnungen von den AnwenderInnen frei definierbar sind. Auf Basis dieser neuen Anforderung hat der Autor das Zähler ACC angepasst. Es wurde ein Konfigurationspopup erstellt, in welchem für die Berechnung der Mittelwerte die jeweiligen Zeitbereiche parametrisiert werden können. Demonstrativ wird nachfolgend das Konfigurationspopup für die Mittelwertbildung der Intervallverbräuche abgebildet.

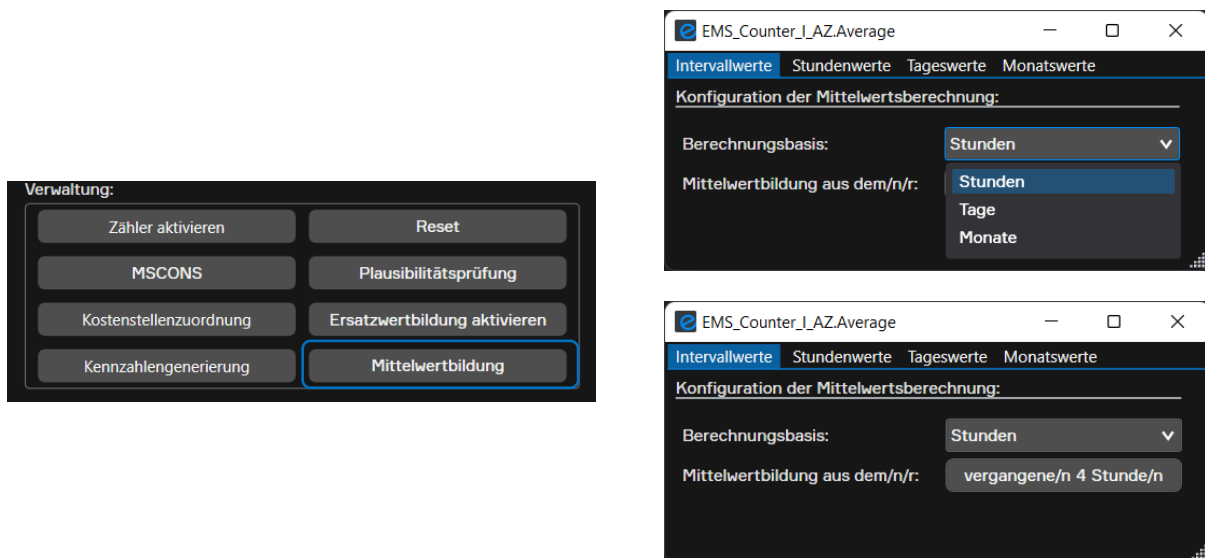


Abbildung 48: Parametrierfelder zur Konfiguration der Mittelwertbildung (eigene Darstellung)

Wie zu erkennen ist, kann bei der Mittelwertberechnung der Intervallverbräuche die Zeitspanne in Stunden, Tagen und Monaten angegeben werden. Je nach Auswahl des Modus wird das Suffix innerhalb des Eingabefeldes angepasst. Die Auswahlmöglichkeit ist bei den Tages- und Monatswerten nicht ident, da beispielsweise eine Mittelwertbildung von Tagesdaten über vergangene Stunden naturgemäß nicht möglich ist. Für die Berechnung der Werte wurden die C#-Methoden des Programmteils angepasst. Der dazugehörige Quellcodeauszug für die Intervalldaten befindet sich demonstrativ in Anhang C. Aufgrund der klaren Formulierung der Anforderung, der einfachen Implementierung der Funktion und des erfolgreichen Tests in der Simulationsumgebung, wird von einer weiteren qualitativen Evaluation dieser Änderung abgesehen. Da die Berechnung in der vorherigen Variante bei den ExpertInnen reibungslos verlief, kann davon ausgegangen werden, dass auch nach dieser Abänderung die Funktion weiterhin gegeben ist. Nach Abschluss der Anpassung der Mittelwertbildung ist die adäquate Umsetzung aller Anforderungen abgeschlossen und es kann zum letzten Schritt im Design Science, der Diffusion, übergegangen werden.

5.5 Diffusion

5.5.1 Veröffentlichung des neuen EMSControl

Die Diffusions-Phase des Design Science Prozesses nach ÖSTERLE ET AL. zielt u. a. darauf ab, eine dauerhafte Nutzung der entworfenen Artefakte zu gewährleisten (Benner-Wickner et al., 2020, S. 8; Österle et al., 2010, S. 668). Da im Rahmen dieser Masterarbeit bereits veröffentlichte Softwareprogramme adaptiert wurden, ist eine Veröffentlichung ohne Weiteres möglich. Da der Autor über Upload-Berechtigungen im ACC Store der evon GmbH verfügt, können die überarbeiteten ACCs in den Store geladen und somit allen KundInnen mit entsprechendem Zugang verfügbar gemacht werden. Durch Download auf Seite der Stakeholder werden die Instanzen der Energiezähler aufgrund der objektorientierten Programmierung automatisch aktualisiert. Damit ist gewährleistet, dass die Stakeholder kein aufwändiges Updateszenario durchlaufen müssen, um zu den neuen Features zu gelangen. In Abbildung 49 wird der Upload-Prozess auszugsweise dargestellt.

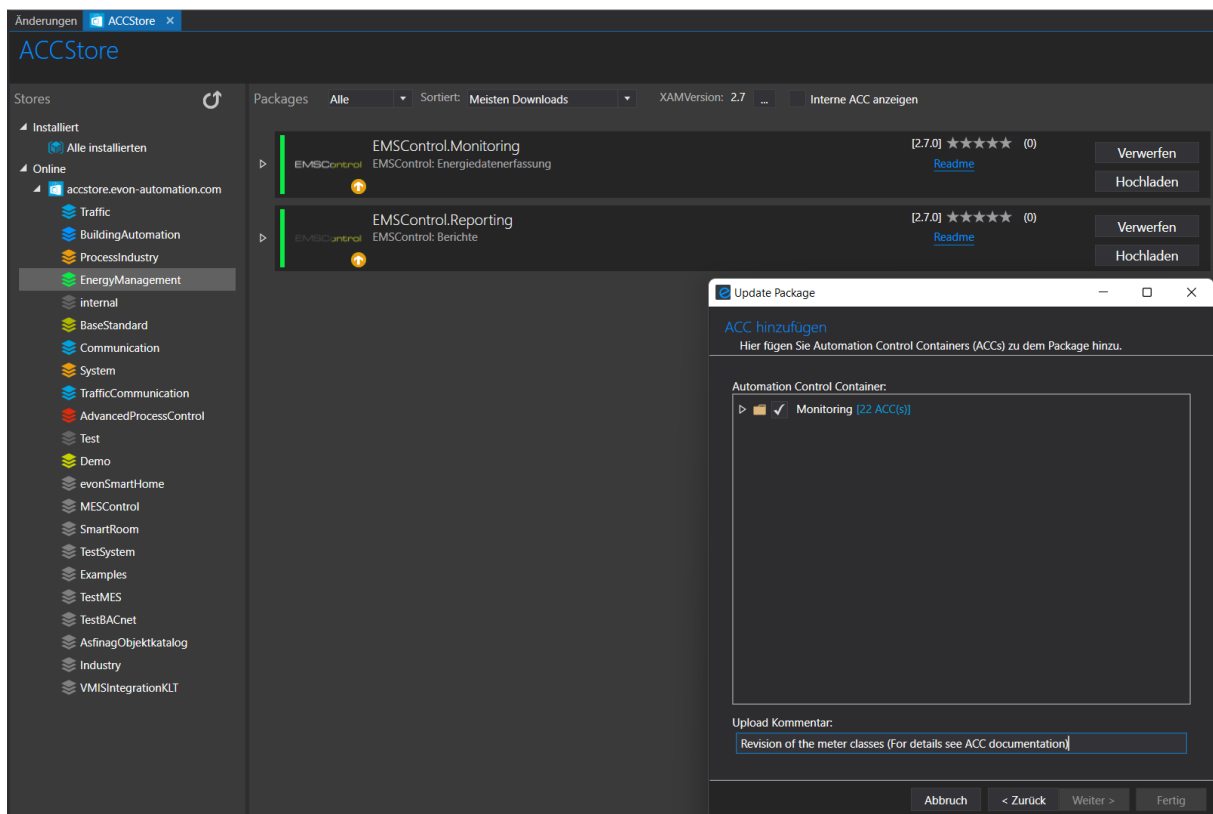


Abbildung 49: Upload der ACCs in den ACC Store (eigene Darstellung)

5.5.2 Veröffentlichung der aktuellen Marktanforderungen

Zudem wird eine Diffusion durch die Masterarbeit selbst erreicht, da die identifizierten Marktanforderungen an eine EDM-Software übersichtlich zusammengefasst veröffentlicht werden. Die Anforderungen wurden mittels einem Requirements Engineering Prozess identifiziert und mit internen und externen ExpertInnen geprüft. Dadurch ist die Qualität der ermittelten Anforderungen gewährleistet und es kann ausgesagt werden, dass die in dieser Masterarbeit dargelegten Marktanforderungen dem Stand März 2022 entsprechen.

5.5.3 Lessons Learned

Darüber hinaus wurden einige Lessons Learned durch Bearbeitung des Praxisproblems aufgedeckt, welche nachfolgend erläutert werden.

Es wurde festgestellt, dass die Ermittlung der aktuellen Anforderungen an eine EDM-Software auf Basis von Sekundärforschung schwierig ist, da zu diesem Thema wenig Quellen zu finden sind. In den Gesetzestexten und Normen werden zwar Anforderungen definiert, allerdings sind diese stets allgemein gehalten und es können daraus keine Funktionen abgeleitet werden, welche eine EDM-Software unterstützen soll. Der von der TÜV Süd AG übermittelte Kriterienkatalog ist nicht frei verfügbar und wurde ausnahmsweise für die Bearbeitung der Masterarbeit übermittelt. Erst wenn eine Zertifizierung nach ISO 50001 begonnen wird, bekommen die Softwarehersteller den Katalog übermittelt und wissen, welche Anforderungen ihre Software unterstützen muss. Nach Meinung des Autors stellt dies aus programmiertechnischer Sicht ein großes Problem dar, da die Software aufwendig adaptiert werden muss, sofern Kriterien des Katalogs nicht erfüllt sind. Als Lessons Learned kann hierbei somit festgehalten werden, dass Softwarehersteller, welche eine EDM-Software entwickeln möchten, gleich zu Beginn eine ISO Zertifizierung bei der TÜV Süd AG anfragen sollten. Dadurch erhalten sie den aktuellen Kriterienkatalog und wissen, welche Anforderungen sie in der Software umsetzen müssen. Da der Kriterienkatalog stets geändert und auf die aktuellen Normen angepasst wird, sind die identifizierten Anforderungen dieser Masterarbeit eine Momentaufnahme und können nicht verallgemeinert werden.

Zudem wurde der Vorteil der Anwendung des .NET Frameworks sowie der objektorientierten Programmierung in einer Software aufgezeigt. Durch das .NET Framework im XAMControl konnte die Online-Wetterdatenabfrage oder das Importieren von Daten aus CSV-Dateien ohne Probleme umgesetzt werden, da auf verschiedenste

Werkzeuge der Frameworks, wie beispielsweise die Hochsprache C# und dessen Klassen zurückgegriffen werden konnte. Da das .NET Framework weit verbreitet ist, gibt es im Internet unzählige Anleitungen und Lösungsansätze für Probleme, welche bei der Erstellung der Programmcodes entstehen. Dies war nach Meinung des Autors ein großer Vorteil während der Entwicklung der Artefakte. Hinsichtlich der objektorientierten Programmierung kann ausgesagt werden, dass grafische oder programmtechnische Änderungen ausschließlich in der jeweiligen Klasse durchgeführt werden müssen. Dadurch werden alle Instanzen dieser Klasse automatisch aktualisiert, was Updateprozesse auf bereits bestehenden Systemen vereinfacht. Dies konnte auch im Rahmen der Tests der ExpertInnen ausgenutzt werden. Der Autor hat die neuen Klassen den ExpertInnen per Mail bzw. Wettransfer übermittelt. Diese konnten wiederum die Klassen einfach austauschen und alle ihre Zählerinstanzen wurden upgedatet. Dies hat den gesamten Aufwand für die Evaluierung der Artefakte stark reduziert.

Ferner sind Reports in EDM-Softwares ein wichtiger Bestandteil für Analysezwecke. Daher ist es wichtig, dass diese für die individuellen Bedürfnisse der Stakeholder ohne hohen Aufwand erstellt oder angepasst werden können. Hierbei konnte im Rahmen der Masterarbeit aufgezeigt werden, dass bei Verwendung einer Microsoft SQL-Datenbank eine Adaptierung der Reports mithilfe des Microsoft SQL Report Builders einfach möglich ist. Da auch Microsoft SQL weit verbreitet ist, gibt es auch hier gute Dokumentationen und Anleitungen, welche bei der Erstellung der Reports nützlich sind.

6 Resümee

6.1 Zusammenfassung

Im Rahmen dieser Masterarbeit wurde untersucht, wie ein marktkonformes Modell einer EDM-Software anhand der obsoleten Software EMSControl aussehen kann. Um diese Frage beantworten zu können, wurden zuerst die zentralen Begrifflichkeiten, die Ziele und Motive sowie die gesetzlichen Rahmenbedingungen hinsichtlich des Themas Energiemanagement erläutert und die Prozesse eines EnMS nach ISO 50001 definiert. Zudem wurde das Energiedatenmanagement vom Energiemanagement abgegrenzt. Hierbei wurde geklärt, was die zentrale Aufgabe eines EDM-Systems ist und wo es im sogenannten PDCA-Zyklus angesiedelt ist.

Damit ein marktkonformes Modell entwickelt werden konnte, zielte eine Subforschungsfrage darauf ab, zu ermitteln, welche rechtlichen und kundenspezifischen Marktanforderungen für EDM-Softwares aktuell existieren. Dies wurde mittels einem Requirements Engineering Prozess beantwortet, indem die entsprechenden Anforderungen identifiziert wurden. Dazu wurde auf alle drei Anforderungsquellen des Requirements Engineerings zurückgegriffen. Es wurden Dokumente in Form von Gesetzestexten, Normen und Kriterienkatalogen herangezogen. Ferner wurde ein Marktspiegel von bereits in Betrieb befindlichen Systemen verwendet. Zudem wurden Feedbacks von einigen Stakeholdern der evon GmbH als Anforderungen aufgenommen. Die identifizierten Requirements wurden entsprechend dokumentiert und zusammen mit einem Teamleiter der evon GmbH sowie vier Stakeholdern geprüft. Im letzten Schritt des Requirements Engineering wurde die Dokumentation der Anforderungen, für alle MitarbeiterInnen zugänglich, in einem eigens dafür eingerichteten Microsoft Teams-Kanal abgelegt.

Im Anschluss an die Anforderungserhebung wurde eine Ist-Analyse der bestehenden EDM-Software der evon GmbH durchgeführt. Dabei wurde die Softwarearchitektur und der allgemeine Systemaufbau des EMSControl erläutert. Anschließend wurden die derzeitig implementierten Funktionen aufgezeigt, um daraus feststellen zu können, welche Anforderungen aus dem Requirements Engineering nicht in der Software umgesetzt sind.

Nach Abschluss der Ist-Analyse war es das Ziel, die Differenzen zwischen der aktuellen, obsoleten EDM-Software und den Marktanforderungen zu schließen. Hierbei stellte sich

die Subforschungsfrage, in welcher Form die Anforderungen in das EMSControl implementiert werden können. Dafür wurde auf das gestaltungsorientierte Forschungsparadigma der Wirtschaftsinformatik namens Design Science zurückgegriffen. Dieses Paradigma zielt darauf ab, Praxisprobleme durch die Entwicklung von sogenannten Artefakten zu lösen. Die im Rahmen der Masterarbeit erstellten Artefakte waren Instanziierungen, welche Teillösungen einer gesamten Software darstellen. Die Anforderungen für die Entwicklung der Artefakte stammten aus dem Requirements Engineering und der Ist-Analyse. In Abstimmung mit dem Teamleiter der Gebäudeautomation der evon GmbH wurden 12 Anforderungen definiert, welche im Zuge dieser Masterarbeit umgesetzt werden sollten. Die Programmierung der Artefakte für das EMSControl erfolgte nach dem Wasserfallmodell nach ROYCE in der Automatisierungslösung XAMControl.

Nach der Entwicklung der Artefakte stellte sich die Frage, ob mithilfe von qualitativen ExpertInneninterviews eine Praxistauglichkeit bzw. Marktkonformität bestätigt werden kann. Als ExpertInnen wurden vier Stakeholder der evon GmbH herangezogen. Bis auf die Mittelwertbildung der Verbrauchsdaten sowie die aktuelle ISO 50001:2018 Zertifizierung, zeigten die Ergebnisse der Interviews, dass laut den Stakeholdern die Artefakte adäquat entwickelt wurden. Die Anmerkungen hinsichtlich der Mittelwertbildung sind wiederum in dem iterativen Design Science Prozess eingeflossen. Demnach wurden die neuen Anforderungen aufgrund der Anmerkungen ermittelt und das Artefakt in der Entwurfsphase überarbeitet. Von einer weiteren Überprüfung mittels Interviews wurde abgesehen, da die neue Anforderung an die Mittelwertbildung klar formuliert war und somit exakt umgesetzt werden konnte. Zusammen mit dem Teamleiter wurde bereits in der Analysephase vereinbart, dass die ISO 50001:2018 Zertifizierung nicht Teil dieser Masterarbeit ist.

Die Diffusion des Design Science Prozesses wurde erreicht, indem die Artefakte als neue Version des EMSControl in den ACC Store geladen und somit für alle KundInnen veröffentlicht wurden. Zudem wird die Diffusion durch die Veröffentlichung der Masterarbeit mit der inkludierten Dokumentation der Marktanforderungen für EDM-Softwares erreicht. Ferner entstanden durch die Bearbeitung des Praxisproblems der evon GmbH Lessons Learned, welche ebenfalls durch die Masterarbeit veröffentlicht werden.

Zusammenfassend kann ausgesagt werden, dass – mit Stand März 2022 – in einer EDM-Software 51 rechtliche bzw. kundenspezifische Anforderungen umgesetzt sein müssen, um eine Marktkonformität gewährleisten zu können. Es wurde am Beispiel des

EMSCControl zeigt, wie ein Modell einer EDM-Software aussehen kann und durch welche Vorgehensweise im Rahmen der objektorientierten Programmierung, eine Implementierung von nicht vorhandenen Anforderungen möglich ist. Somit kann diese Arbeit durch die Zusammenführung von theoretischen Inhalten mit der praktischen Umsetzung als Referenz für alle Hersteller von EDM-Software verwendet werden.

6.2 Kritische Reflexion und Ausblick

Nach Meinung des Autors sollte der Produktname EMSControl hinterfragt werden, da dieser Name als „EnergiemanagementControl“ assoziiert werden könnte. Im Rahmen der Masterarbeit konnte eindeutig das EDM vom EnMS abgegrenzt werden. Demnach ist eine EDM-Software ausschließlich für die Aufzeichnung und managementtaugliche Aufbereitung von Energiedaten zuständig und befindet sich im Check-Abschnitt des PDCA-Zyklus. Aus diesem Grund legt der Autor dem Unternehmen eine Umbenennung des Produktes nahe. Eine mögliche Produktbezeichnung wäre beispielsweise „EDMControl“. Zudem wird kritisch angemerkt, dass die Kriterien für die ISO 50001 Zertifizierung einer EDM-Software nicht öffentlich zugänglich sind und erst zu Beginn einer Zertifizierung übermittelt werden. Dadurch ist es für Hersteller einer EDM-Software schwierig, schon zu Beginn der Entwicklung zu wissen, welche Funktionen ihre neue Software unterstützen muss.

Aufgrund der vom Autor dieser Masterarbeit durchgeführten Anpassungen im EMSControl wird im Jahr 2022 eine Neu-Zertifizierung nach ISO 50001:2018 für EDM-Systeme angestrebt. Diese muss nach Meinung des Autors unbedingt zeitnahe erfolgen, da ansonsten das aktuell bei den Stakeholdern eingesetzte EMSControl mit Fremdsystemen ersetzt wird bzw. keine weiteren Neuaufträge aufgrund der fehlenden Zertifizierung akquiriert werden können. Des Weiteren besteht die Möglichkeit, dass das BAFA das EMSControl aus der Liste der förderfähigen Softwaresysteme entfernt, wenn diese eine Neuerhebung ihrer gelisteten Systeme durchführt. Erfolgt eine positive Neu-Zertifizierung, wird die Zukunft zeigen, ob der Relaunch des EMSControl aufgrund der implementierten Marktanforderungen erfolgreich war oder das Produkt vom Markt ausscheidet.

Zudem wird sich zeigen, wie sich allgemein das Thema Energiemanagement hinsichtlich politischer Entscheidungen verändern wird und ob dadurch EDM-Systeme in naher Zukunft einen noch höheren Stellenwert in den Unternehmen aufweisen.

Literaturverzeichnis

- Ahmad, Muhammad/Mourshed, Monjur/Mundow, David/Sisinni, Mario/Rezgui, Yacine (2016). Building energy metering and environmental monitoring – A state-of-the-art review and directions for future research. *Energy and Buildings*, 120, S. 85–102.
- Appenzeller, Heinz (2021). *Kostenoptimierte Anwendungsentwicklung. Reduzierung der Entwicklungskosten durch Trennung der Datenverwaltungs- und Fachfunktionen*, Wiesbaden: Springer Fachmedien.
- Aumayr, Klaus J. (2017). *Erfolgreiches Produktmanagement. Tool-Box für das professionelle Produktmanagement und Produktmarketing*, 5. Aufl., Wiesbaden: Springer Fachmedien.
- BAFA (2020). Modul 3 - MSR, Sensorik und Energiemanagement-Software. Technische Mindestanforderungen. Online: https://eskap.de/site/assets/files/1205/eew_modul3_ems_merkblatt_tma_2020.pdf (Abruf am 07.10.2021).
- Balaji, Srinivasan/Murugaiyan, Sundararajan (2012). Waterfall vs V-Model Vs Agile: A comparative Study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), S. 26–29.
- Baskerville, Richard/Baiyere, Abayomi/Gregor, Shirley/Hevner, Alan/Rossi, Matti (2018). Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems*, 19(5), S. 358–376.
- Benner-Wickner, Marian/Kneuper, Ralf/Schlömer, Inga (2020). Leitfaden für die Nutzung von Design Science Research in Abschlussarbeiten. *IUBH Discussion Papers - IT & Engineering*, 2, S. 1–10.
- BfEE – Bundesstelle für Energieeffizienz beim Bundesamt für Wirtschaft und Ausfuhrkontrolle (2018). Energiemanagement. Online: https://www.bfee-online.de/BfEE/DE/Energiedienstleistungen/Energiemanagement/energiemanagement_node.html (Abruf am 18.05.2021).

BMK – Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie (o. D.). Energieeffizienzgesetz-Neu, Übergangsregelungen vom Energieeffizienzgesetz (alt) zum „Energieeffizienzgesetz-Neu“, Online: <https://www.bmk.gv.at/themen/energie/effizienz/recht/effizienzgesetz.html> (Abruf am 23.07.2021).

BMK – Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie (2021). Das Übereinkommen von Paris. Online: https://www.oesterreich.gv.at/themen/bauen_wohnen_und_umwelt/klimaschutz/1/Seite.1000325.html (Abruf am 17.07.2021).

BMLRT – Bundesministerium für Landwirtschaft, Regionen und Tourismus (2018): Nachweis der Verbesserung der energiebezogenen Leistung. Guideline zur Umsetzung des Ziels der ISO 50001, die energiebezogene Leistung laufend zu verbessern. Online: <https://www.klimaaktiv.at/dam/jcr:0ae25523-abff-41d3-ad7f-7c9393b3c8c6/Guideline%20Nachweis%20der%20Verbesserung%20der%20energiebezogene%20Leistung%20-%20barrierefrei.pdf> (Abruf am 23.01.2022).

BMWi – Bundesministerium für Wirtschaft und Technologie (2010). Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung. Online: https://www.bmwi.de/Redaktion/DE/Downloads/E/energiekonzept-2010.pdf?__blob=publicationFile&v=5 (Abruf am 26.07.2021).

Boehm, Barry W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, 21(5), S. 61–72.

Brauner, Günther (2014). Das europäische Energieeffizienzgesetz und seine Umsetzung. *e & i Elektrotechnik und Informationstechnik*, 131(4), S. 114–118.

Bränzel, Juliane/Engelmann, Dirk/Geilhausen, Marko/Schulze, Olaf (2019). *Energiemanagement. Für Fachkräfte, Beauftragte und Manager*, 2. Aufl., Wiesbaden: Springer Fachmedien.

Brugger-Gebhardt, Simone/Jungblut, Günter (2019). *Die DIN EN ISO 50001:2018 verstehen. Die Norm sicher interpretieren und sinnvoll umsetzen*, Wiesbaden: Springer Fachmedien.

- Bundesnetzagentur (2006). Festlegung zu den Geschäftsprozessen und Datenformaten. Online: https://www.bundesnetzagentur.de/DE/Beschlusskammern/1_GZ/BK6-GZ/2006/BK6-06-009/BK6-06-009_Beschluss_download.pdf?__blob=publicationFile&v=5 (Abruf am 11.01.2022).
- Bundesnetzagentur (2021a). EEG-Umlage. Online: https://www.bundesnetzagentur.de/SharedDocs/A_Z/E/EEG_Umlage.html (Abruf am 26.07.2021).
- Bundesnetzagentur (2021b). MSCONS Anwendungshandbuch. Online: https://www.bundesnetzagentur.de/DE/Beschlusskammern/BK06/BK6_83_Zug_Mess/835_mitteilungen_datenformate/Mitteilung_21/EDIFACT-Konsultationsdokumente/MSCONS_AHB_3_1.pdf?__blob=publicationFile&v=1 (Abruf am 11.01.2022).
- Burghardt, Manfred (2018). *Projektmanagement. Leitfaden für die Planung, Überwachung und Steuerung von Projekten*, 10. Aufl., Erlangen: Publicis Publishing Verlag.
- Danish Energy Agency (2014). Energy Policy Toolkit on Energy Efficiency in Industries. Experiences from Denmark. Online: <https://c2e2.unepdtu.org/wp-content/uploads/sites/3/2016/03/energy-efficiency-in-industriesfinal-1.pdf> (Abruf am 03.09.2021).
- Dehli, Martin (2020). *Energieeffizienz in Industrie, Dienstleistung und Gewerbe*, Wiesbaden: Springer Fachmedien.
- Deng, Qi/Ji, Shaobo (2018). A Review of Design Science Research in Information Systems. Concept, Process, Outcome, and Evaluation. *Pacific Asia Journal of the Association for Information Systems*, 10(1), S. 1–36.
- Drenkelfort Gregor/Pröhl Thorsten/Erek Koray/Behrendt Frank/Zarnekow Rüdiger (2013). Energiemonitoring im IKT-Umfeld Standards und Trends. In Jorge Marx Gómez/ Corinna Lang/Volker Wohlgemuth (Hrsg.), *IT-gestütztes Ressourcen- und Energiemanagement*, Heidelberg: Springer Verlag, S. 81–91.
- Dresing, Thorsten/Pehl, Thorsten (2018). *Praxisbuch Interview, Transkription & Analyse. Anleitungen und Regelsysteme für qualitativ Forschende*, 8. Aufl., Marburg: Eigenverlag.

- Dröge, Susanne (2015). Das Pariser Abkommen 2015: Weichenstellung für das Klimaregime. *SWP-Studie 19/2015*, Berlin: Stiftung Wissenschaft und Politik - SWP- Deutsches Institut für Internationale Politik und Sicherheit.
- DWD (o. D.). Klimafaktoren (KF) für Energieverbrauchsangabe. Online: <https://www.dwd.de/DE/leistungen/klimafaktoren/klimafaktoren.html> (Abruf am 12.01.2022).
- Energieagentur NRW (2021). EMS Marktspiegel. Online: <https://www.energieagentur.nrw/tool/ems-marktspiegel/> (Abruf am 08.10.2021).
- Europäische Kommission (2021). Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates zur Energieeffizienz (Neufassung). Online: https://eur-lex.europa.eu/resource.html?uri=cellar:a214c850-e574-11eb-a1a5-01aa75ed71a1.0006.02/DOC_1&format=PDF (Abruf am 20.11.2021).
- Evon (2021). XAMControl. Online: <https://developer.evon-automation.com/wp-content/uploads/Documentation/2.7.7/DE/articles/XAMiris/Einfuehrung.html> (Abruf am 14.10.2021).
- Frauchinger, Daniel (2017). Anwendungen von Design Science Research in der Praxis. In Edy Portmann (Hrsg.), *Wirtschaftsinformatik in Theorie und Praxis. Festschrift zu Ehren von Prof. Dr. Andreas Meier*, Wiesbaden: Springer Fachmedien, S. 107–118.
- Frost, Irasianty (2018). *Einfache lineare Regression*, Wiesbaden: Springer Fachmedien.
- Geilhausen, Marko/Bränzel, Juliane/Engelmann, Dirk/Schulze, Olaf (2015). *Energiemanagement. Für Fachkräfte, Beauftragte und Manager*, Wiesbaden: Springer Fachmedien.
- Geilhausen, Marko (2015). *Kompakter Leitfaden für Energiemanager. Energiemanagementsysteme nach ISO 50001*, Wiesbaden: Springer Fachmedien.
- Geilhausen, Marko (2020). *Kompakter Leitfaden für Energiemanager. Energiemanagementsysteme nach DIN EN ISO 50001:2018*, 2. Aufl., Wiesbaden: Springer Fachmedien.
- Geißler, Otto/Ostler, Ulrike (2018). Was ist ein Application-Programming-Interface (API)?. Online: <https://www.datacenter-insider.de/was-ist-ein-application-programming-interface-api-a-735797/> (Abruf am 19.12.2021).

- Golovatchev, Julius (2020). Digitaler Zwilling für die Realisierung Utility 4.0. In Oliver D. Doleski (Hrsg.), *Realisierung Utility 4.0 Band 2: Praxis der digitalen Energiewirtschaft vom Vertrieb bis zu innovativen Energy Services*, Wiesbaden: Springer Verlag, S. 593–609.
- Gonzalez-Perez, Maria Alejandra (2016). Climate Change and the 2030 Corporate Agenda for Sustainable Development. *Advances in Sustainability and Environmental Justice*, 19, S. 1–6.
- Hering, Ekbart (2015). *Kostenrechnung und Kostenmanagement für Ingenieure*, Wiesbaden: Springer Fachmedien.
- Hevner, Alan/March, Salvatore/Park, Jinsoo/Ram, Sudha (2004). Design science in information systems research. *MIS Quarterly*, 28(1), S. 75–105.
- Hunsaker, Claire (2015). REST VS SOAP: When Is REST Better?. Online: <https://stormpath.com/blog/rest-vs-soap> (Abruf am 19.12.2021).
- IEA – International Energy Agency (2020). CO2 Emissions From Fuel Combustion 2020. Online: <https://de.statista.com/statistik/daten/studie/167957/umfrage/verteilung-der-co-emissionen-weltweit-nach-bereich/#professional> (Abruf am 27.01.2021).
- ISO – International Organization for Standardization (2015). *Energiemanagementsysteme – Anforderungen mit Anleitung zur Anwendung (ISO 50001:2011)*. Deutsche Fassung.
- ISO – International Organization for Standardization (2018). *Energy management systems — Requirements with guidance for use (ISO Standard No. 50001:2018)*. English Version.
- Janitza (o. D.). Multifunktionaler Netzanalysator mit Ethernet und RCM UMG 96RM-E. Online: <https://www.janitza.de/umg-96rm-e.html> (Abruf am 04.10.2021).
- Janson, Matthias (2020). Es wird wärmer auf der Erde. Online: <https://de.statista.com/infografik/4708/abweichung-von-der-globalen-durchschnittstemperatur/> (Abruf am 10.05.2021).
- Johannesson, Paul/Perjons, Erik (2014). *An Introduction to Design Science*, Cham: Springer International Publishing.

- Jovanović, Bojana/Filipović, Jovan/Bakić, Vukman (2017). Energy management system implementation in Serbian manufacturing – Plan-Do-Check-Act cycle approach. *Journal of Cleaner Production*, 162, S. 1144–1156.
- Kappler, Cornelia (2014). Energie und Kosten sparen mit einem IT-basierten Energiecontrollingsystem. In Ronald Gleich (Hrsg.), *Energiecontrolling. Energiekosten systematisch steuern und senken*. München: Haufe Verlag, S. 225–242.
- Kohn, Wolfgang/Öztürk, Riza (2017). *Statistik für Ökonomen: Datenanalyse mit R und SPSS*, Heidelberg: Springer Verlag.
- Köppl, Angela/Schleicher, Stefan (2021). *Systemische Perspektiven zur Energieeffizienz. Unterstützende Analysen für die Novellierung des österreichischen Energieeffizienzgesetzes*, Wien: WIFO – Österreichisches Institut für Wirtschaftsforschung.
- Liegl, Philipp (2021). Aufbau einer EDIFACT-Datei. Online: <https://ecosio.com/de/blog/aufbau-einer-edifact-datei/> (Abruf am 11.01.2022).
- Luber, Stefan (2017). Was ist eine API?. Online: <https://www.dev-insider.de/was-ist-eine-api-a-583923/> (Abruf am 19.12.2021).
- Maalem, Sourour/Zarour, Nacereddine (2016). Challenge of validation in requirements engineering. *Journal of Innovation in Digital Ecosystems*, 3(1), S. 15–21.
- March, Salvatore/Smith, Gerald (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4), S. 251–266.
- Masek, Pavel/Hudec, David/Krejci, Jan/Ometov, Aleksandr/Hosek, Jiri/Samouylov, Konstantin (2018). Communication Capabilities of Wireless M-BUS: Remote Metering Within SmartGrid Infrastructure. In Mizuho Hoshi/Shinnosuke Seki (Hrsg.), *Developments in Language Theory Lecture Notes in Computer Science*. Cham: Springer International Publishing, S. 31–42.
- Matyokurehwa, Kanos/Mavetera, Nehemiah/Jokonya, Osden (2017). Requirements Engineering Techniques. A Systematic Literature Review. *International Journal of Soft Computing and Engineering*, 7(1), S. 14–20.

- Mayring, Philipp (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken*, 12. Aufl., Basel: Beltz Verlag.
- Mijač, Marko (2019). Evaluation of Design Science instantiation artifacts in Software engineering research. *Central European Conference on Information and Intelligent Systems. Proceedings. 30th CECIS*, Varaždin, Croatia, 2.–4. Oktober 2019, S. 313–321. University of Zagreb, Faculty of Organization and Informatics Varaždin.
- Mudaliar, Mani Dheeraj/Sivakumar, N. (2020). IoT based real time energy monitoring system using Raspberry Pi. *Internet of Things*, 12, 100292.
- Munassar, Nabil Mohammed Ali/Govardhan, Ali (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues*, 7(5), S. 94–101.
- Nickenig, Karin (2017). *Grundkurs Kosten- und Leistungsrechnung*, Wiesbaden: Springer Fachmedien.
- OpenWeather (2021a). Pricing. Online: <https://openweathermap.org/price> (Abruf am 19.12.2021).
- OpenWeather (2021b). How to start. Online: <https://openweathermap.org/appid> (Abruf am 19.12.2021).
- OpenWeather (2021c). 5 day weather forecast. Online: <https://openweathermap.org/forecast5> (Abruf am 19.12.2021).
- Österle, Hubert/Becker, Jörg/Frank, Ulrich/Hess, Thomas/Karagiannis, Dimitris/Krcmar, Helmut/Loos, Peter/Mertens, Peter/Oberweis, Andreas/Sinz, Elmar J. (2010). Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung*, 62(6), S. 664–672.
- Paetsch, Frauke/Eberlein, Armin/Maurer, Frank (2003). Requirements engineering and agile software development. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linz, Austria, 11. Juni 2003, S. 308–313. IEEE.
- Pohl, Klaus (2010). *Requirements Engineering. Fundamentals, Principles, and Techniques*, Heidelberg: Springer Verlag.

- Pohl, Klaus/Rupp, Chris (2015). *Basiswissen Requirements Engineering. Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*, 4. Aufl., Heidelberg: dpunkt Verlag.
- Posch, Wolfgang (2011). *Ganzheitliches Energiemanagement für Industriebetriebe*, Wiesbaden: Springer Fachmedien.
- Prettenthaler, Franz/ Gobiet, Andreas/Habsburg-Lothringen, Clemens/Steinacker, Reinhold/Töglhofer, Christoph/Türk, Andreas (2007). Auswirkungen des Klimawandels auf Heiz- und Kühlenergiebedarf in Österreich. Online: https://startclim.at/fileadmin/user_upload/StartClim2006/StCl06F.pdf (Abruf am 19.12.2021).
- Richards, Robert (2006). *Pro PHP XML and Web Services*, Berkeley: Apress.
- Robra-Bissantz, Susanne/Strahringer, Susanne (2020). Wirtschaftsinformatik-Forschung für die Praxis. *HMD Praxis der Wirtschaftsinformatik*, 57(2), S. 162–188.
- Roden, Golo (2008). *Auf der Fährte von C#*, Heidelberg: Springer Verlag.
- Rowley, Jennifer/Slack, Frances (2004). Conducting a literature review. *Management Research News*, 27(6), S. 31–39.
- Rupp, Chris/Simon, Matthias/Hocker, Florian (2009). Requirements Engineering und Management. *HMD Praxis der Wirtschaftsinformatik*, 46(3), S. 94–103.
- Rupp, Chris/Schüpferling, Dirk (2014). Anforderungsermittlung – Hellsehen für Fortgeschrittene. In Chris Rupp/die SOPHISTen (Hrsg.), *Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil*, 5. Aufl., München: Carl Hanser Verlag, S. 89–121.
- Schellong, Wolfgang (2016). *Analyse und Optimierung von Energieverbundsystemen*, Heidelberg: Springer Verlag.
- Schmitt, Robert/Günther, Sebastian (2014). *Industrielles Energiemanagement*, München: Carl Hanser Verlag.

- Schödwell, Björn/Drenkelfort, Gregor/Erek, Koray/Zarnekow, Rüdiger/Behrendt, Frank (2012). Auf dem Weg zu einem ganzheitlichen, quantitativen Bewertungsansatz für Energiemonitoring-Systeme in Rechenzentren. In Ursula Goltz/Marcus Magnor/Hans-Jürgen Appelrath/Herbert Matthies/Wolf-Tilo Balke/Lars Wolf (Hrsg.), *INFORMATIK 2012*, Bonn: Gesellschaft für Informatik e.V., S. 189–203.
- Schweinfurth, Holger (2020). Energiedatenmanagement – EDMS, 18 Big Data, Smart Data. In Oliver D. Doleski (Hrsg.), *Realisierung Utility 4.0 Band 2. Praxis der digitalen Energiewirtschaft vom Vertrieb bis zu innovativen Energy Services*, Wiesbaden: Springer Verlag, S. 227–239.
- Slate, Andrew (2019). Die Programmierschnittstelle (API - Application Programming Interface) erklärt. Online: <https://www.wrike.com/de/blog/programmierschnittstelle-api-erklart/> (Abruf am 19.12.2021).
- Sonnenberg, Christian/vom Brocke, Jan (2012). Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In Ken Peffers/Marcus Rothenberger/Bill Kuechler (Hrsg.), *Design Science Research in Information Systems. Advances in Theory and Practice*, Heidelberg: Springer Verlag, S. 381–397.
- Statistik Austria (2020). Gesamtenergiebilanz Österreich 1970 bis 2019. Online: https://www.statistik.at/web_de/statistiken/energie_umwelt_innovation_mobilitaet/energie_und_umwelt/energie/energiebilanzen/index.html (Abruf am 04.09.2021).
- Thommen, Jean-Paul/Achleitner, Ann-Kristin, Gilbert, Dirk Ulrich/Hachmeister, Dirk/Kaiser, Gernot (2017). *Allgemeine Betriebswirtschaftslehre. Umfassende Einführung aus managementorientierter Sicht*, 8. Aufl., Wiesbaden: Springer Fachmedien.
- Tiedemann, Jan (2021). Digitalisierung der industriellen Dinge. Worauf es bei der Verbindung von IT und OT ankommt. *Datenschutz und Datensicherheit - DuD*, 45(2), S. 83–89.
- Trentler, Klaus (2013). Energiemanagement – Make the most of your energy. *e & i Elektrotechnik und Informationstechnik*, 130(8), 273–274.

- TUP (2015). EDifact: Standard für Electronic Data Interchange for administration, commerce and transport. Online: <https://logistikknowhow.com/codiertechnik/edifact-standard-fuer-electronic-data-interchange-for-administration-commerce-and-transport/> (Abruf am 11.01.2022).
- TÜV (2021). Zertifiziertes Energiedatenmanagement. Online: <https://www.tuvsud.com/de-de/branchen/produzierende-industrie/anlagenbau-betrieb/energieeffizienz/energiedatenmanagement-zertifizierung> (Abruf am 05.10.2021).
- Umweltbundesamt (2020). Klimaschutzbericht 2020. Online: <https://www.umweltbundesamt.at/fileadmin/site/publikationen/rep0738.pdf> (Abruf am 02.02.2021).
- UNFCCC (1997). Kyoto Protocol to the United Nations Framework Convention on Climate Change adopted at COP3 in Kyoto, Japan, on 11 December 1997. Online: <https://unfccc.int/resource/docs/cop3/07a01.pdf> (Abruf am 15.07.2021).
- Venable, John/Pries-Heje, Jan/Baskerville, Richard (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, 25(1), S. 77–89.
- Vresk, Tomislav/Cavrak, Igor (2016). Architecture of an interoperable IoT platform based on microservices. *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 30. Mai–3. Juni 2016, S. 1196–1201. IEEE
- Weber, Eva (2014). Prozessbedingte CO₂-Emissionen in Bayern. *Bayern in Zahlen*, 4, S. 247–250.
- Weiß, Christel (2019). *Basiswissen Medizinische Statistik*, Heidelberg: Springer Verlag.
- Windsperger, Andreas/Windsperger, Bernhard/Schörner, Georg (2014). *Umsetzung der Energieeffizienzrichtlinie in Österreich. Analyse des Energiedienstleistungsmarktes auf Potentiale der Weiterentwicklung*. St. Pölten: Institut für Industrielle Ökologie.
- Winter, Robert (2008). Design science research in Europe. *European Journal of Information Systems*, 17(5), S. 470–475.

Xiao, Yu/Watson, Maria (2019). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, 39(1), S. 93–112.

Zabel, Hans-Ulrich (2012). Nachhaltigkeitsorientiertes Energiemanagementsystem – Anforderungen, Potentiale und Inhalte. *uwf UmweltWirtschaftsForum*, 20(1), S. 89–94.

Abbildungsverzeichnis

Abbildung 1: Sektorale Verteilung der weltweiten energiebedingten CO ₂ -Emissionen im Jahr 2018	1
Abbildung 2: Phasen des Produktlebenszyklus	4
Abbildung 3: Aufbau eines EnMS nach ISO 50001	17
Abbildung 4: Prozesskomponenten des EDM.....	25
Abbildung 5: Multifunktionsmessgerät UMG 96 RM-E	26
Abbildung 6: Beispiel eines Stromlastgangs	28
Abbildung 7: Topologie eines EDM-Systems.....	28
Abbildung 8: Systemarchitektur XAMControl.....	41
Abbildung 9: ACC Store und Janitza Smart Meter Klassen im XAMControl	44
Abbildung 10: Beispiel Panel.....	45
Abbildung 11: Beispiel Popup mit OperatorPanel	45
Abbildung 12: Beispiel PLCProgram.....	46
Abbildung 13: Architektur einer Modbus Datenübertragung	48
Abbildung 14: Beispiel EMSControl PAC Sentron 3200 Instanz	48
Abbildung 15: Beispiel EMSControl Zähler Instanz.....	49
Abbildung 16: Beispiel EMSControl Zähler Popup.....	50
Abbildung 17: Beispiel EMSControl Listendarstellung mit Filterfunktion.....	50
Abbildung 18: Beispiel EMSControl Monatsreport	51
Abbildung 19: Design Science Prozess nach Österle et al.	55
Abbildung 20: Wasserfallmodell nach Royce	60
Abbildung 21: Instanz des Wetter ACCs.....	64

Abbildung 22: OperatorPanel des Wetter ACCs	64
Abbildung 23: ParameterPanel des Wetter ACCs	65
Abbildung 24: Popup des Gradtage ACCs	67
Abbildung 25: Report Heizgradtage	68
Abbildung 26: Erstelltes Interface für MSCONS Export.....	69
Abbildung 27: Erstelltes Konfigurationspopup für MSCONS Export.....	69
Abbildung 28: Beispiel MSCONS-Textdatei	70
Abbildung 29: Import einer generierten MSCONS-Textdatei in den EDIFACT Viewer..	70
Abbildung 30: Parametrierfelder zur Einheitendefinition	72
Abbildung 31: Parametrierfelder zur Kostenstellenkonfiguration.....	73
Abbildung 32: Popup des Kostenstellen ACCs.....	74
Abbildung 33: Tagesreport des Kostenstellen ACCs.....	75
Abbildung 34: Parameter für dynamische Diagrammumschaltung.....	76
Abbildung 35: SQL Report Expression für dynamische Diagrammumschaltung.....	76
Abbildung 36: SQL Report mit dynamischer Diagrammumschaltung	77
Abbildung 37: Grafische Darstellung der berechneten Mittelwerte	78
Abbildung 38: Peak in den Energiedaten aufgrund eines Netzwerkausfalls.....	78
Abbildung 39: Ersatzwertbildung der Energiedaten nach einem Netzwerkausfall	79
Abbildung 40: Parametrierfelder zur Konfiguration der Plausibilitätsprüfung	81
Abbildung 41: Logeintrag der Plausibilitätsprüfung	81
Abbildung 42: Popup für Kennzahlengenerierung	83
Abbildung 43: CSV-Messgrößen-Import.....	83
Abbildung 44: Report EnPI.....	84

Abbildung 45: Streudiagramm mit Regressionsgerade	86
Abbildung 46: Regressionsgleichung im Konfigurationspopup	86
Abbildung 47: Report EnPI Regression.....	87
Abbildung 48: Parametrierfelder zur Konfiguration der Mittelwertbildung	90
Abbildung 49: Upload der ACCs in den ACC Store	91

Tabellenverzeichnis

Tabelle 1: Mindestanforderungen an eine EDM-Software nach ISO 50001	32
Tabelle 2: Mindestanforderungen an eine EDM-Software nach dem BAFA.....	32
Tabelle 3: Anforderungen an eine EDM-Software nach der Energieagentur NRW	34
Tabelle 4: Kundenfeedbacks der evon GmbH	35
Tabelle 5: Anforderungsdokumentation einer EDM-Software	37
Tabelle 6: Systemmodule XAMControl	43
Tabelle 7: Allgemeine Design Science Richtlinien	56
Tabelle 8: Design Science Richtlinien bezogen auf die Masterarbeit.....	56
Tabelle 9: Vergleich zwischen EDM-Softwareanforderungen und EMSControl	58
Tabelle 10: Systemanforderungen XAMControl	61
Tabelle 11: Beispielberechnung Heizgradtage.....	66
Tabelle 12: Medien und Einheiten im EMSControl.....	71
Tabelle 13: Ergebnisse aus den ExpertInneninterviews	89

Formelverzeichnis

Formel 1: Berechnung Heizgradtage.....	66
Formel 2: Berechnung Kühlgradtage.....	66
Formel 3: Berechnung Klimakorrekurfaktor.....	67
Formel 4: Regressionsgerade.....	85

Quellcodeverzeichnis

Quellcodeausschnitt 1: Abfrage Koordinaten.....	120
Quellcodeausschnitt 2: Abfrage Wetterdaten	123
Quellcodeausschnitt 3: Berechnung der Min-Max-Werte für Gradtagberechnung.....	123
Quellcodeausschnitt 4: Trigger bilden und Zuweisung der Gradtagewerte	124
Quellcodeausschnitt 5: Berechnung der Gradtagewerte	125
Quellcodeausschnitt 6: Archivierung der Tageswerte	125
Quellcodeausschnitt 7: Archivierung der Monatswerte.....	126
Quellcodeausschnitt 8: SQL Query für Report der Gradtage.....	126
Quellcodeausschnitt 9: Zuweisung Zählerwerte für MSCONS-Export	127
Quellcodeausschnitt 10: MSCONS-Export.....	128
Quellcodeausschnitt 11: Rücksetzen der Zählerwerte.....	129
Quellcodeausschnitt 12: Berechnung der Kostenstellenverbräuche	131
Quellcodeausschnitt 13: Berechnung des Mittelwerts der Intervalle der letzten Stunde	132
Quellcodeausschnitt 14: Ersatzwertbildung von Intervallwerten.....	134
Quellcodeausschnitt 15: Instanziierung Zeitschaltprogramm.....	134

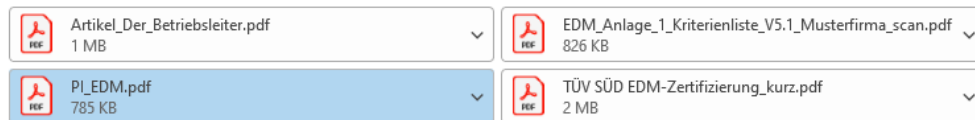
Quellcodeausschnitt 16: Plausibilitätsprüfung von Intervallwerten	134
Quellcodeausschnitt 17: CSV-Import von Messgrößen für EnPI-Berechnung.....	138
Quellcodeausschnitt 18: CSV-Export von Messgrößen für EnPI-Berechnung.....	140
Quellcodeausschnitt 19: EnPI-Berechnung.....	142
Quellcodeausschnitt 20: Reportauswahl mit Click-Event	142
Quellcodeausschnitt 21: Dynamische Anpassung vom Suffix des Parametrierfeldes für den Zeitbereich.....	143
Quellcodeausschnitt 22: Berechnung des Mittelwerts der Intervalle des definierten Zeitbereichs	144

Abkürzungsverzeichnis

ACC	Automation Control Container
API	Application Programming Interface
BAFA	Bundesamt für Wirtschaft und Ausfuhrkontrolle
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EDL-G	Gesetz über Energiedienstleistungen und andere Energieeffizienzmaßnahmen
EMD	Energiedatenmanagement
EEffG	Energieeffizienzgesetz
EEG	Erneuerbare-Energien-Gesetz
EnEff-RL	Energieeffizienzrichtlinie 2012/27/EU
EnergieStG	Energiesteuergesetz
EnSaG	Energiedatensammelgesetz
EMAS	Eco-Management and Audit Scheme
EnMS	Energiemanagementsystem
EnB	Energetische Ausgangsbasen
EnPI	Energy Performance Indicator
FMEA	Fehlermöglichkeits- und Einflussanalyse
FBD	Function Block Diagram
FVP	Fortlaufender Verbesserungsprozess
GUI	Graphical User Interface
ISO	Internationale Organisation für Normung
KMU	Kleine und mittlere Unternehmen
MSCONS	Metered Services Consumption Report Message
PDCA	Plan-Do-Check-Act
RIS	Rechtsinformationssystem
RPZ	Risikoprioritätszahl
SCADA	Supervisory Control and Data Acquisition
SDLC	Software Development Life Cycle
SEU	Significant Energy Use
SPS	Speicherprogrammierbare Steuerung
StromStG	Stromsteuergesetz
UML	Unified Modeling Language

Anhang A

In diesem Anhang befindet sich die E-Mail der TÜV Süd AG an den Autor.



Sehr geehrter Herr Müller,

im Anhang finden Sie einige Beschreibungen und die Kriterienliste für die TÜV SÜD Zertifizierung von Energiedatenmanagementsystemen. Bitte studieren Sie die Dokumentationen und melden Sie sich danach für spezielle Fragen.

Der Kriterienkatalog darf nicht weitergegeben oder allgemein veröffentlicht werden. Für Ihre Masterarbeit können Sie die Dokumente nutzen. Dabei bitte die Quellenangabe beachten.

Zur Zeit bis ich noch bis Mitte November im Außendienst, danach können wir noch einmal miteinander telefonieren.

Viel Erfolg.

Mit freundlichen Grüßen / Kind regards
Steffen Kügler

[Informationen Energieeffizienz](#)

Sachverständiger Energiesysteme/ Lead-Auditor
Geschäftsstelle Dresden
Abteilung Anlagensicherheit
Energiesysteme
Phone: +49 351 4202-328
Mobile: +49 160 70 44 521
Fax: +48 351 4202-356
Email: steffen.kuegler@tuvsud.com
www.tuvsud.com/de-is



TÜV SÜD Industrie Service GmbH
Drescherhäuser 5d
01159 Dresden
Germany

Quelle: Kügler, Steffen (2021). EDM-Kriterienkatalog, Persönliche E-Mail, 27.10.2021.

Anhang B

In diesem Anhang sind die durchgeführten Besprechungsprotokolle enthalten.

Protokoll 1:

EINBERUFEN VON	David Müller
DATUM	19.10.2021 09:30 bis 10:00
ORT	Online via Microsoft Teams
BESPRECHUNGSART	Prüfung und Abstimmung der identifizierten Anforderungen
BESPRECHUNGSLEITER/IN	David Müller
PROTOKOLLFÜHRER/IN	David Müller
TEILNEHMER/IN	<ul style="list-style-type: none">• evon GmbH: David Müller, Teamleiter GLT• Stakeholder 1: Teamleiter Technische IT eines Salzwerks• Stakeholder 2: Teamleiter Instandhaltung Elektro, GLT und Energiemanagement eines Energieversorgers• Stakeholder 3: Teamleiter eines Glücksspielunternehmens• Stakeholder 4: Teamleiter Electrical Engineer Maintenance eines Herstellers von Halbfabrikaten aus Kupfer und Kupferlegierungen

Agenda:

Ziel dieses Meeting war, die von Herrn Müller identifizierten Anforderungen an eine EDM-Software mit Stakeholder der evon GmbH sowie mit dem Teamleiter der GLT auf Vollständigkeit zu validieren.

Ergebnisse aus der Besprechung:

Herr Müller übermittelte vorab die ausgearbeitete Anforderungsdokumentation an die beteiligten Personen. In der Besprechung wurde die Dokumentation über das Screensharing von Microsoft Teams geöffnet, um die identifizierten Anforderungen gemeinsam zu besprechen. Alle Personen waren mit der Ausarbeitung von Herrn Müller zufrieden und haben diese als gut, vollständig und die Beschreibungen als verständlich befunden.

Stakeholder 2 betonte, dass für sein Unternehmen (Energieversorgungsunternehmen) der MSCONS-Export von großer Bedeutung ist und dieser Herr Müller bei der Implementierung in das EMSControl für Fragen sehr gerne zur Verfügung steht, da das MSCONS Format unübersichtlich ist und Laien in diesem Gebiet Schwierigkeiten haben könnten, dieses zu verstehen. Zudem wurde betont, dass eine rückwirkende Aufzeichnung bei Ausfällen sowie eine aktuelle ISO 50001:2018 Zertifizierung für die Stakeholder sehr relevant sind. Weiters wurde erwähnt, dass bei den Online-Wetterdaten – neben den aktuellen Temperaturen – auch Vorhersagewerte abgefragt werden sollten, da diese im XAMControl für andere Zwecke, wie beispielsweise zur Lüftungssteuerung, verwendet werden könnten. Somit müssen nicht zwei unterschiedliche ACCs entwickelt werden.

Es wurde vereinbart, dass nachdem Herr Müller die Ist-Analyse der Software durchgeführt hat, zusammen mit dem Teamleiter der evon GmbH vereinbart wird, welche Anforderungen in das System implementiert werden müssen. Anschließend beginnt Herr Müller mit der Programmierung und kann bei Fragen hinsichtlich MSCONS auf Stakeholder 2 zurückgreifen.

Protokoll 2:

EINBERUFEN VON	David Müller
DATUM	06.12.2021 13:00 bis 14:00
ORT	evon GmbH
BESPRECHUNGSART	Abstimmung der umzusetzenden Anforderungen
BESPRECHUNGSLEITER/IN	David Müller
PROTOKOLLFÜHRER/IN	David Müller
TEILNEHMER/IN	<ul style="list-style-type: none">• evon GmbH: David Müller, Teamleiter GLT

Agenda:

Diese Besprechung wurde als Folgebesprechung vom 19.10.2021 gehalten. Dabei sollte zusammen mit dem Teamleiter der Gebäudeleittechnik geklärt werden, welche Requirements von Herrn Müller im Rahmen der Masterarbeit umgesetzt werden.

Ergebnisse aus der Besprechung:

Herr Müller hat im Zuge der Ist-Analyse festgestellt, dass 14 rechtliche bzw. kundenspezifische Anforderungen in der aktuellen Software nicht umgesetzt sind. Es wurde definiert, dass im Rahmen der Masterarbeit von Herrn Müller 12 Anforderungen umgesetzt werden. Req-6 Cloud-Installation kann technisch von Herrn Müllern nicht realisiert werden, da dafür auch das XAMControl Cloudfähig sein muss und dies derzeit nicht gegeben ist. Derzeit gibt es noch keine Ambitionen das XAMControl als SaaS anzubieten. Die Anforderung wurde allerdings als Feedback an die Geschäftsführung bzw. Entwicklungsabteilung der evon GmbH weitergegeben. Ferner kann Req-51 ISO 50001:2018 Zertifizierung im Zuge der Masterarbeit nicht umgesetzt werden, da dies aufgrund der zeitlichen Beschränkung der Arbeit nicht realisierbar ist. Die Zertifizierung ist allerdings ab Sommer 2022 geplant. Es wurde vereinbart, dass Herr Müller im nächsten Schritt mit der Umsetzung der Anforderungen und Erstellung der Artefakte beginnt. Sind die internen Tests abgeschlossen, werden die Artefakte an die vier Stakeholder aus Besprechung 1 übermittelt, damit diese praxisbezogene Tests unter realen Bedingungen durchführen können. Im Anschluss werden Interviews mit den vier Stakeholdern durchgeführt und evaluiert, ob die Artefakte praxistauglich sind und die Anforderungen korrekt umgesetzt wurden.

Anhang C

In diesem Anhang sind die Ausschnitte der Quellcodes dargestellt.

OpenWeatherMap API:

In den folgenden Methoden werden die Längen und Breitengrade des eingegebenen Orts abgefragt, um im Anschluss die Wetterdaten zu erhalten.

```
1. public class Coord {
2.     public double lon { get; set; }
3.     public double lat { get; set; }
4. }
5.
6. public class Main {
7.     public double temp { get; set; }
8.     public double feels_like { get; set; }
9.     public double temp_min { get; set; }
10.    public double temp_max { get; set; }
11.    public int pressure { get; set; }
12.    public int humidity { get; set; }
13. }
14.
15. public class Wind {
16.     public double speed { get; set; }
17.     public int deg { get; set; }
18. }
19.
20. public class Clouds {
21.     public int all { get; set; }
22. }
23.
24. public class Sys {
25.     public int type { get; set; }
26.     public int id { get; set; }
27.     public string country { get; set; }
28.     public int sunrise { get; set; }
29.     public int sunset { get; set; }
30. }
31.
32. public class Data {
33.     public Coord coord { get; set; }
34.     public List<Weather> weather { get; set; }
35.     public string @base { get; set; }
36.     public Main main { get; set; }
37.     public int visibility { get; set; }
38.     public Wind wind { get; set; }
39.     public Clouds clouds { get; set; }
40.     public int dt { get; set; }
41.     public Sys sys { get; set; }
42.     public int timezone { get; set; }
43.     public int id { get; set; }
44.     public string name { get; set; }
45.     public int cod { get; set; }
46. }
47.
48. bool Active_Coordinates = false;
49. string JsonResponse_Coordinates = string.Empty;
50.
51. void GetLatLong()
52. {
53.
54.
55.     if (this.Trigger)
```

```

56.  {
57.    Trigger = false;
58.    GetMyCoordinates();
59.  }
60. }
61.
62. private async void GetMyCoordinates()
63. {
64.
65.     if (Active_Coordinates)
66.         return;
67.
68.     try
69.     {
70.         Active_Coordinates = true;
71.         await GetMyCoordinatesFromAsync();
72.     }
73.     catch (Exception ex)
74.     {
75.         XAMLIB.TraceViewer.Trace(this.InstanceName, ex.ToString(),
76.             XAMCommon.Trace.TracePrio.ERROR);
77.     }
78.
79.     finally
80.     {
81.         Active_Coordinates = false;
82.     }
83. }
84.
85. private System.Threading.Tasks.Task GetMyCoordinatesFromAsync()
86. {
87.     System.Threading.Tasks.Task task = System.Threading.Tasks.Task.Run(() =>
88.     {
89.         try
90.         {
91.             if (AppKey != "" && City != "")
92.             {
93.                 string url =
94.                     string.Format("https://api.openweathermap.org/data/2.5/weather?q={0}&appid={1}",
95.                         this.City, this.AppKey);
96.
97.                 System.Net.HttpWebRequest request = (System.Net.HttpWebRequest)
98.                     System.Net.WebRequest.Create(url);
99.
100.                 request.Method = "GET";
101.                 request.PreAuthenticate = true;
102.                 request.Timeout = 10000;
103.
104.                 using(var response = (System.Net.HttpWebResponse) request.GetResponse()) {
105.
106.                     if (response.StatusCode != System.Net.HttpStatusCode.OK)
107.                     {
108.                         Message = String.Format("Request failed. Received HTTP {0}",
109.                             response.StatusCode);
110.                     }
111.
112.                     using(System.IO.Stream dataStream = response.GetResponseStream())
113.                     {
114.                         System.IO.StreamReader reader = new System.IO.StreamReader(dataStream);
115.                         string responseFromServer = reader.ReadToEnd();
116.                         JsonResponse_Coordinates = (responseFromServer);
117.                         Message = String.Format("Wetterabfrage OK");
118.                         LastAccess = DateTime.Now;
119.                         this.ErrData = false;
120.                     }
121.
122.                     Data values = Newtonsoft.Json.JsonConvert.DeserializeObject < Data >
123.                         (JsonResponse_Coordinates);
124.
125.
126.                     this.Latitude = values.coord.lat.ToString();

```

```

127.         this.Longitude = values.coord.lon.ToString();
128.         CityRequestOK = true;
129.
130.         Trigger_Data = true;
131.     }
132. }
133. }
134.
135. catch (Exception ex)
136. {
137.     this.Message = "Wetterdaten nicht aktuell!";
138.     this.ErrData = true;
139.     XAMLIB.TraceViewer.Trace("Online-Wetter: " + this.InstanceName, ex.ToString(),
140.     XAMCommon.Trace.TracePrio.ERROR);
141.     CityRequestOK = false;
142.     Longitude = "";
143.     Latitude = "";
144.     Trigger_Data = false;
145. }
146.
147.     return;
148. });
149.
150.     return task;
151. }

```

Quellcodeausschnitt 1: Abfrage Koordinaten (eigene Darstellung)

In den folgenden Methoden werden die Wetterdaten des eingegebenen Orts abgefragt.

```

1.     public class WeatherData    {
2.         public int id { get; set; }
3.         public string main { get; set; }
4.         public string description { get; set; }
5.         public string icon { get; set; }
6.     }
7.
8.     public class Current    {
9.         public int dt { get; set; }
10.        public int sunrise { get; set; }
11.        public int sunset { get; set; }
12.        public double temp { get; set; }
13.        public double feels_like { get; set; }
14.        public int pressure { get; set; }
15.        public int humidity { get; set; }
16.        public double dew_point { get; set; }
17.        public double uvi { get; set; }
18.        public int clouds { get; set; }
19.        public int visibility { get; set; }
20.        public double wind_speed { get; set; }
21.        public int wind_deg { get; set; }
22.        public List<WeatherData> weather { get; set; }
23.    }
24.
25.    public class Temp    {
26.        public double day { get; set; }
27.        public double min { get; set; }
28.        public double max { get; set; }
29.        public double night { get; set; }
30.        public double eve { get; set; }
31.        public double morn { get; set; }
32.    }
33.
34.    public class FeelsLike    {
35.        public double day { get; set; }
36.        public double night { get; set; }
37.        public double eve { get; set; }
38.        public double morn { get; set; }

```

```

39.     }
40.
41.     public class Daily    {
42.         public int dt { get; set; }
43.         public int sunrise { get; set; }
44.         public int sunset { get; set; }
45.         public Temp temp { get; set; }
46.         public FeelsLike feels_like { get; set; }
47.         public int pressure { get; set; }
48.         public int humidity { get; set; }
49.         public double dew_point { get; set; }
50.         public double wind_speed { get; set; }
51.         public int wind_deg { get; set; }
52.         public List<WeatherData> weather { get; set; }
53.         public int clouds { get; set; }
54.         public double pop { get; set; }
55.         public double uvi { get; set; }
56.     }
57.
58.     public class Root    {
59.         public double lat { get; set; }
60.         public double lon { get; set; }
61.         public string timezone { get; set; }
62.         public int timezone_offset { get; set; }
63.         public Current current { get; set; }
64.         public List<Daily> daily { get; set; }
65.     }
66.
67. string JsonResponse_Data = string.Empty;
68. bool Active_Data;
69.
70. void GetValues()
71. {
72.     if (this.Trigger_Data)
73.     {
74.         Trigger_Data = false;
75.         GetMyData();
76.     }
77. }
78. private async void GetMyData()
79. {
80.
81.     if (Active_Data)
82.         return;
83.
84.     try
85.     {
86.         Active_Data = true;
87.         await GetMyDataFromAsync();
88.     }
89.
90.     catch (Exception ex)
91.     {
92.         XAMLIB.TraceViewer.Trace(this.InstanceName, ex.ToString()),
93.         XAMCommon.Trace.TracePrio.ERROR);
94.     }
95.
96.     finally
97.     {
98.         Active_Data = false;
99.     }
100. }
101.
102. private System.Threading.Tasks.Task GetMyDataFromAsync()
103. {
104.     System.Threading.Tasks.Task task = System.Threading.Tasks.Task.Run(() =>
105.     {
106.
107.         try
108.         {
109.             if (AppKey != "" && Longitude != "" && Latitude != "")

```



```

110.     {
111.         string url =
112.             string.Format("https://api.openweathermap.org/data/2.5/onecall?lat={0}&lon={1}&
113.             exclude=minutely,hourly,alerts&appid={2}&units=metric&lang=de",
114.                 this.Latitude, this.Longitude, this.AppKey);
115.
116.         System.Net.HttpWebRequest request = (System.Net.HttpWebRequest)
117.             System.Net.WebRequest.Create(url);
118.
119.         request.Method = "GET";
120.         request.PreAuthenticate = true;
121.         request.Timeout = 10000;
122.
123.         using(var response = (System.Net.HttpWebResponse) request.GetResponse()) {
124.
125.             if (response.StatusCode != System.Net.HttpStatusCode.OK) {
126.                 var Message = String.Format("Request failed. Received HTTP {0}",
127.                     response.StatusCode);
128.             }
129.
130.             using(System.IO.Stream dataStream = response.GetResponseStream())
131.             {
132.                 System.IO.StreamReader reader = new System.IO.StreamReader(dataStream);
133.                 string responseFromServer = reader.ReadToEnd();
134.                 JsonResponse_Data = (responseFromServer);
135.                 Message = String.Format("Wetterabfrage OK");
136.                 LastAccess = DateTime.Now;
137.                 this.ErrData = false;
138.             }
139.
140.             Root values = Newtonsoft.Json.JsonConvert.DeserializeObject < Root >
141.                 (JsonResponse_Data);
142.
143.
144.             this.TimeZone = values.timezone;
145.             this.TimeZone_Offset = (values.timezone_offset) / 3600;
146.
147.             //Aktuelle Werte
148.             this.ActTemp = Math.Round(values.current.temp, 1);
149.             this.ActHumidity = values.current.humidity;
150.             this.ActPressure = values.current.pressure;
151.             this.ActDewPoint = Math.Round(values.current.dew_point, 1);
152.             this.ActCloud = values.current.clouds;
153.             this.ActWindSpeed = values.current.wind_speed;
154.             this.ActWindDegree = values.current.wind_deg;
155.
156.             this.ActWeather = values.current.weather[0].main;
157.             this.ActWeatherDescription = values.current.weather[0].description;
158.             this.ActWeatherID = values.current.weather[0].id;
159.
160.             System.DateTime sunrise = new DateTime(1970, 1, 1, 0, 0, 0, 0,
161.                 System.DateTimeKind.Utc);
162.             this.Sunrise = sunrise.AddSeconds(values.current.sunrise).ToUniversalTime();
163.             this.Sunrise = Sunrise.AddHours(TimeZone_Offset);
164.
165.             System.DateTime sunset = new DateTime(1970, 1, 1, 0, 0, 0, 0,
166.                 System.DateTimeKind.Utc);
167.             this.Sunset = sunset.AddSeconds(values.current.sunset).ToUniversalTime();
168.             this.Sunset = Sunset.AddHours(TimeZone_Offset);
169.
170.             System.DateTime acttime = new DateTime(1970, 1, 1, 0, 0, 0, 0,
171.                 System.DateTimeKind.Utc);
172.             this.ActTime = sunset.AddSeconds(values.current.dt).ToUniversalTime();
173.             this.ActTime = ActTime.AddHours(TimeZone_Offset);
174.
175.             this.CurrentWday = (int) ActTime.Date.DayOfWeek;
176.
177.             //Vorhersagewerte
178.             this.ActForeCastHigh = Math.Round(values.daily[0].temp.max, 1);
179.             this.ActForeCastLow = Math.Round(values.daily[0].temp.min, 1);
180.             this.ActStateCode = values.daily[0].weather[0].main;

```

```

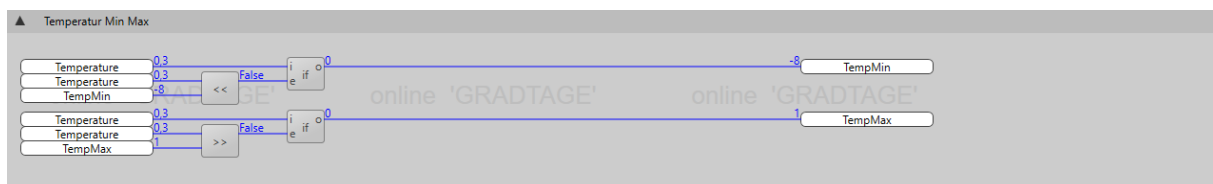
181.         this.ActStateText = values.daily[0].weather[0].description;
182.
183.         this.ForecastHigh[0] = Math.Round(values.daily[1].temp.max, 1);
184.         this.ForecastLow[0] = Math.Round(values.daily[1].temp.min, 1);
185.         this.ForecastCode[0] = values.daily[1].weather[0].main;
186.         this.ForecastText[0] = values.daily[1].weather[0].description;
187.
188.         this.ForecastHigh[1] = Math.Round(values.daily[2].temp.max, 1);
189.         this.ForecastLow[1] = Math.Round(values.daily[2].temp.min, 1);
190.         this.ForecastCode[1] = values.daily[2].weather[0].main;
191.         this.ForecastText[1] = values.daily[2].weather[0].description;
192.
193.         this.ForecastHigh[2] = Math.Round(values.daily[3].temp.max, 1);
194.         this.ForecastLow[2] = Math.Round(values.daily[3].temp.min, 1);
195.         this.ForecastCode[2] = values.daily[3].weather[0].main;
196.         this.ForecastText[2] = values.daily[3].weather[0].description;
197.
198.         this.ForecastHigh[3] = Math.Round(values.daily[4].temp.max, 1);
199.         this.ForecastLow[3] = Math.Round(values.daily[4].temp.min, 1);
200.         this.ForecastCode[3] = values.daily[4].weather[0].main;
201.         this.ForecastText[3] = values.daily[4].weather[0].description;
202.
203.     }
204. }
205. }
206.
207. catch (Exception ex)
208. {
209.     this.Message = "Wetterdaten nicht aktuell!";
210.     this.ErrData = true;
211.     XAMLIB.TraceViewer.Trace("Online-Wetter: " + this.InstanceName, ex.ToString(),
212.     XAMCommon.Trace.TracePrio.ERROR);
213. }
214.
215.     return;
216.
217. });
218.
219.     return task;
220. }

```

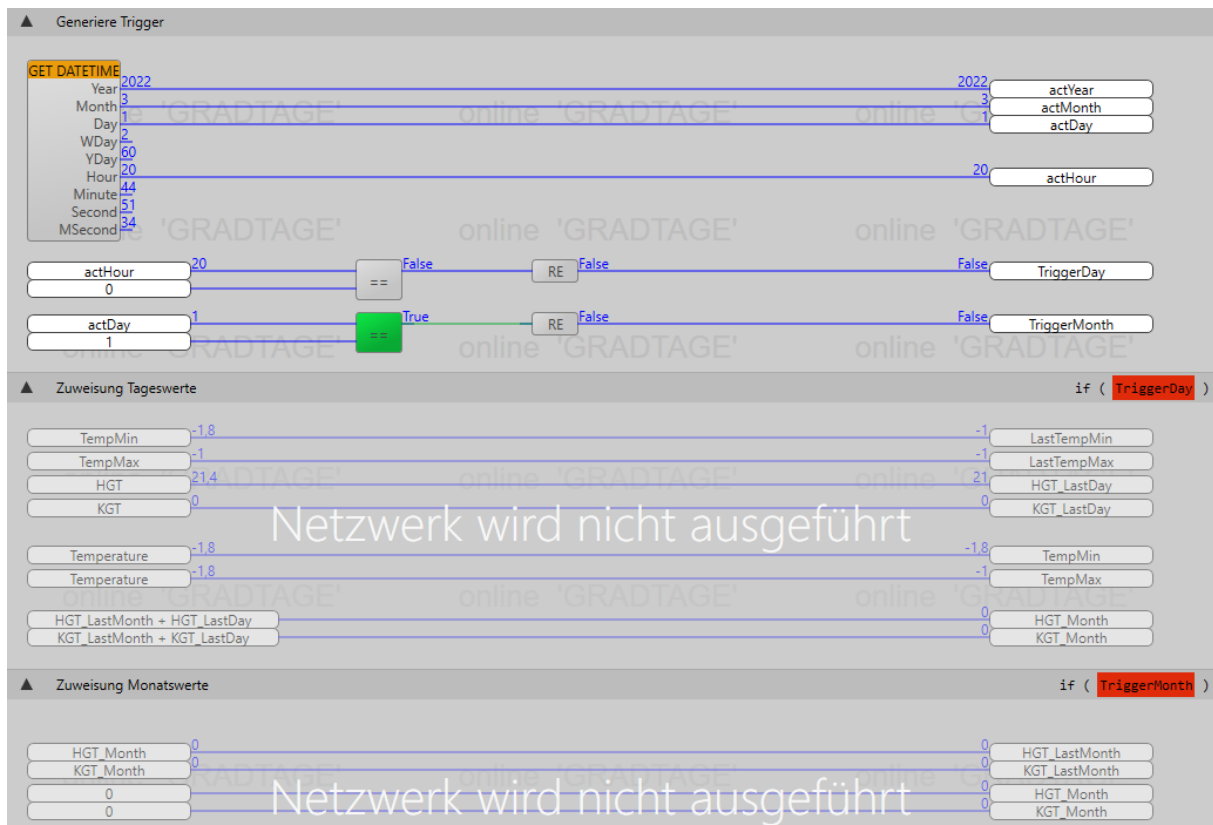
Quellcodeausschnitt 2: Abfrage Wetterdaten (eigene Darstellung)

Witterungsbereinigung:

Mithilfe von FBD werden sowohl die minimalen und maximalen Tagestemperaturen berechnet, der Trigger gebildet als auch die Werte zurückgesetzt.



Quellcodeausschnitt 3: Berechnung der Min-Max-Werte für Gradtageberechnung (eigene Darstellung)



Quellcodeausschnitt 4: Trigger bilden und Zuweisung der Gradtagewerte (eigene Darstellung)

In dieser Methode werden Heiz- und Kühlgradtage auf Basis der eingegebenen Grenzwerte und der mittleren Tagesaußentemperaturen berechnet.

```

1. void Berechnung()
2. {
3.     try
4.     {
5.         //Berechnung Temperaturmittel
6.         this.temp = (this.TempMin + this.TempMax)/2;
7.
8.         //Berechnung Heizgradtage aktueller Tag
9.         if(this.temp < this.Temp_heat)
10.        {
11.            this.HGT = this.Temp_r- this.temp;
12.        }
13.
14.        else
15.        {
16.            this.HGT = 0;
17.        }
18.
19.        //Berechnung Kühlgradtage aktueller Tag
20.        if(this.temp > this.Temp_cool)
21.        {
22.            this.KGT = this.temp - this.Temp_cool;
23.        }
24.
25.        else
26.        {
27.            this.KGT = 0;
28.        }
29.
30.        this.Error = false;
31.    }

```

```

32. catch (Exception ex)
33. {
34.     XAMLIB.TraceViewer.Trace("Gradtage: " + this.InstanceName,
35.     ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
36.     this.ErrMsg = "Berechnungsfehler Gradtage: " + ex.ToString();
37.     this.Error = true;
38. }
39. }

```

Quellcodeausschnitt 5: Berechnung der Gradtagewerte (eigene Darstellung)

In dieser Methode werden die berechneten Tageswerte in die SQL-Datenbank gespeichert.

```

1. void Datenbank_Tageswerte()
2. {
3.     DateTime StoreTime;
4.     DateTime actTime = DateTime.Now;
5.
6.     try
7.     {         if (this.Error == false)
8.             {
9.                 //Speicherung der Tageswerte
10.                StoreTime = new DateTime(actTime.Year, actTime.Month, actTime.Day,
11.                actTime.Hour,0,0);
12.                XAMLIB.Archive.StoreValue(this.XAMCom_HGT_LastDay,StoreTime);
13.                XAMLIB.Archive.StoreValue(this.XAMCom_KGT_LastDay,StoreTime);
14.                this.LastStore = DateTime.Now;
15.            }
16.     }
17.     catch (Exception ex)
18.     {
19.         XAMLIB.TraceViewer.Trace("Gradtage: " + this.InstanceName,
20.         ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
21.         this.ErrMsg = "Aufzeichnungsfehler Gradtage: " + ex.ToString();
22.         this.Error = true;
23.     }
24. }

```

Quellcodeausschnitt 6: Archivierung der Tageswerte (eigene Darstellung)

In dieser Methode werden die berechneten Monatswerte in die SQL-Datenbank gespeichert.

```

1. void Datenbank_Monatswerte()
2. {
3.     DateTime StoreTime;
4.     DateTime actTime = DateTime.Now;
5.
6.     try
7.     {         if (this.Error == false)
8.             {
9.                 //Speicherung der Monatswerte
10.                StoreTime = new DateTime(actTime.Year, actTime.Month, actTime.Day,
11.                actTime.Hour,0,0);
12.                XAMLIB.Archive.StoreValue(this.XAMCom_HGT_LastMonth,StoreTime);
13.                XAMLIB.Archive.StoreValue(this.XAMCom_KGT_LastMonth,StoreTime);
14.                this.LastStore = DateTime.Now;
15.            }
16.     }
17.     catch (Exception ex)
18.     {
19.         XAMLIB.TraceViewer.Trace("Gradtage: " + this.InstanceName,

```

```

20.         ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
21.         this.ErrMsg = "Aufzeichnungsfehler Gradtage: " + ex.ToString();
22.         this.Error = true;
23.     }
24. }

```

Quellcodeausschnitt 7: Archivierung der Monatswerte (eigene Darstellung)

Nachfolgend wird der SQL Query für den Report der Heiz- bzw. Kühlgradtage dargestellt.

```

1. SELECT a.*, i.InstancePath, case when p.Name = @PropertyNameCounter then a.Value end as
2. 'Verbrauchswert', case when p.Name=@PropertyNameDegrees then a.Value end as 'Gradtage',
3. r.Remark as RemarkPath,
4. FROM [XAMRuntimeX4].[dbo].[X4Archives] a
5.
6. inner join XAMControlX4.dbo.AutomationControlInstances i on i.ID=a.InstanceID
7. inner join XAMControlX4.dbo.AutomationControlProperties p on p.ID=a.PropertyID
8. left join XAMControlX4.dbo.AutomationControlInstanceRemarks r on
9. r.Instance_ID=a.InstanceID
10.
11. where ((i.InstancePath = @InstanceNameCounter and p.Name=@PropertyNameCounter) or
12. (i.InstancePath = @InstanceName and p.Name=@PropertyNameDegrees)) and a.DateTime >
13. @StartTime) and a.DateTime <= @EndTime)
14.
15. order by a.DateTime

```

Quellcodeausschnitt 8: SQL Query für Report der Gradtage (eigene Darstellung)

MSCONS Export:

In dieser Methode erfolgt demonstrativ die Zuweisung der Zählerwerte in das Double Array der Interface-Variable.

```

1. List<double> myList = new List<double>();
2.
3. void Zuweisung_Werte_MSCONS()
4. {
5.
6.     try
7.     {
8.         IQueryable<XAMCommon.XAMRuntimeDB.XAMTrendValue> MyQuery =
9.         XAMCommon.XAMRuntimeDB.X4Archive.Query(this.InstanceName + ".Value",
10.        DateTime.Now.AddDays(-1), DateTime.Now, XAMCommon.XAMCommonBase.X4Runtime);
11.
12.        foreach(XAMCommon.XAMRuntimeDB.XAMTrendValue entry in MyQuery.OrderBy(v =>
13.        v.DateTime))
14.        {
15.            myList.Add(w_lastInterval);
16.        }
17.
18.        Array.Clear(Values_Interval, 0, Values_Interval.Length);
19.        Values_Interval = myList.ToArray();
20.        this.Error = false;
21.    }
22.
23.    catch (Exception ex)
24.    {
25.        XAMLIB.TraceViewer.Trace("MSCONS: "
26.        + this.InstanceName,
27.        ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
28.        this.Error = true;

```

```
29. }
30.
```

Quellcodeausschnitt 9: Zuweisung Zählerwerte für MSCONS-Export (eigene Darstellung)

In dieser Methode erfolgt das Generieren einer Textdatei im MSCONS-Format mit Intervallverbräuchen.

```
1. void Generieren_der_MSCONS_Datei()
2. {
3.
4.     try
5.     {
6.         foreach(NEW_iMSCONSExport item in MSCONS_List)
7.         {
8.             string Message_ShortYear = DateTime.Now.ToString("yy");
9.             string Message_Year = DateTime.Now.ToString("yyyy");
10.            string Message_Month = DateTime.Now.ToString("MM");
11.            string Message_Day = DateTime.Now.ToString("dd");
12.
13.            string Message_Hour = DateTime.Now.ToString("HH");
14.            string Message_Minute = DateTime.Now.ToString("mm");
15.            string Message_Second = DateTime.Now.ToString("ss");
16.
17.            string Message_ShortDate = Message_ShortYear + Message_Month +
18.            Message_Day;
19.            string Message_Date = Message_Year + Message_Month + Message_Day;
20.            string Message_Time = Message_Hour + Message_Minute;
21.            string Message_LinuxTime = Message_Hour + Message_Minute +
22.            Message_Second;
23.
24.            DateTime ActTime = DateTime.Now;
25.            DateTime MSCONSDate = new DateTime(ActTime.Year, ActTime.Month,
26.            ActTime.Day, 0, 0, 0);
27.
28.
29.            string text1 = "UNB+UNOC:3+9800015600006:502+9800129700007:502+" +
30.            Message_ShortDate + ":" + Message_Time + "+" + Message_LinuxTime +
31.            "+VL'"
32.            + "UNH+" + Message_LinuxTime + "+MSCONS:D:04B:UN:2.2h'"
33.            + "BGM+7+" + Message_LinuxTime + "+9'"
34.            + "DTM+137:" + Message_ShortDate + Message_Time + ":203'"
35.            + "RFF+Z13:13002'"
36.            + "NAD+MS+9800015600006::332'"
37.            + "NAD+MR+9800129700007::332'UNS+D'"
38.            + "NAD+DP'"
39.            + "LOC+172+" + item.CounterDescription + "'"
40.            + "DTM+9:" + Message_ShortDate + ":102'"
41.            + "RFF+MG:" + item.CounterNumber + "'"
42.            + "CCI+ACH++PMR'"
43.            + "CCI+16++MRV'"
44.            + "LIN+1'"
45.            + "PIA+5+8-1?:" + item.OBIS + ":SRW'";
46.
47.            string text2 = string.Empty;
48.
49.            foreach (double value in item.Values_Interval)
50.            {
51.                text2 += "QTY+" + item.Status + ":" + value + ":" +
52.                item.Unit + "'" + "DTM+163:" +
53.                MSCONSDate.ToString("yyyyMMddHHmm") + "?+02:303'" +
54.                "DTM+164:" +
55.                MSCONSDate.AddMinutes(15).ToString("yyyyMMddHHmm") +
56.                "?+02:303'";
57.                MSCONSDate = MSCONSDate.AddMinutes(15);
58.            }
59.
```

```

60.         string text3 = "UNT+96+" + Message_LinuxTime + "''"
61.         + "UNZ+1+" + Message_LinuxTime + "'";
62.
63.         System.IO.File.WriteAllText(Path + item.CounterNumber + "_" +
64.         Message_Date + "_" + Message_Time + ".txt", text1+text2+text3);
65.         this.Error = false;
66.     }
67. }
68.
69. catch(Exception ex)
70. {
71.     XAMLIB.TraceViewer.Trace("MSCONS: "
72.     + this.InstanceName,
73.     ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
74.     this.Error = true;
75.     System.IO.File.WriteAllText(Path + ".txt", ex.ToString());
76. }

```

Quellcodeausschnitt 10: MSCONS-Export (eigene Darstellung)

Zählerreset:

In den folgenden Methoden werden alle Live-Werte sowie Archivdaten des Energiezählers zurückgesetzt.

```

1.  bool Active_Data;
2.  void Globales_Ruecksetzen_bei_Inaktiv()
3.  {
4.      if((this.GlobalReset || Reset) && !this.Activate)
5.      {
6.          this.M_Wsum_Master = 0;
7.          this.M_WsumNT_Master = 0;
8.          this.M_WsumHT_Master = 0;
9.          this.W_sum = 0;
10.         this.W_sumNT = 0;
11.         this.W_sumHT = 0;
12.         this.W_lastInterval = 0;
13.         this.W_lastDay = 0;
14.         this.W_lastMonth = 0;
15.         this.W_lastHour = 0;
16.         this.W_lastInterval = 0;
17.         this.ValueIn_HTintern = 0;
18.         this.ValueIn_NTintern = 0;
19.         this.lastValueIn_HTintern = 0;
20.         this.lastValueIn_NTintern = 0;
21.         this.W_calcInterval = 0;
22.         this.W_calcMinute = 0;
23.         this.W_calcHour = 0;
24.         this.W_calcDay = 0;
25.         this.W_calcMonth = 0;
26.         this.W_calcYear = 0;
27.
28.         DeleteArchiveData();
29.         this.GlobalReset = false;
30.         this.Reset = false;
31.     }
32. }
33. }
34.
35. private async void DeleteArchiveData()
36. {
37.
38.     if(Active_Data)
39.         return;
40.
41.     try

```

```

42.  {
43.      Active_Data = true;
44.      await DeleteArchiveDataFromAsync();
45.  }
46.
47.  catch(Exception ex)
48.  {
49.      XAMLIB.TraceViewer.Trace(this.InstanceName, ex.ToString(),
50.      XAMCommon.Trace.TracePrio.ERROR);
51.  }
52.
53.  finally
54.  {
55.      Active_Data = false;
56.  }
57. }
58.
59.
60. private System.Threading.Tasks.Task DeleteArchiveDataFromAsync()
61. {
62.     System.Threading.Tasks.Task task = System.Threading.Tasks.Task.Run(()=>
63.     {
64.         try
65.         {
66.             Err = "";
67.
68.             string SqlConnectionString = "packet size=4096;" + "user id=" +
69.             XAMServer.Server.Server.sc.Config.DatabaseUser + ";data source=" +
70.             XAMServer.Server.Server.sc.Config.DatabaseConnectionString +
71.             ";persist security info=True;initial
72.             catalog=XAMRuntimeX4;password=" +
73.             XAMServer.Server.Server.sc.Config.DatabasePasswort;
74.
75.             using (System.Data.SqlClient.SqlConnection sqlcon = new
76.             SqlConnection(SqlConnectionString))
77.             {
78.                 sqlcon.Open();
79.
80.                 string comm = "DELETE " +
81.                 "FROM [XAMRuntimeX4].[dbo].[X4Archives] " +
82.                 "WHERE InstanceID = '" + this.InstanceID + "'";
83.
84.                 using (System.Data.SqlClient.SqlCommand sqlcomm = new
85.                 SqlCommand(comm, sqlcon))
86.                 {
87.                     sqlcomm.ExecuteNonQuery();
88.                 }
89.                 sqlcon.Close();
90.             }
91.             this.Error = false;
92.         }
93.
94.         catch (Exception ex)
95.         {
96.             XAMLIB.TraceViewer.Trace("EMS Counter Reset: " + this.InstanceName,
97.             ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
98.             this.Error = true;
99.         }
100.        return;
101.    });
102.    return task;
103. }

```

Quellcodeausschnitt 11: Rücksetzen der Zählerwerte (eigene Darstellung)

Kostenstellenzuordnung:

In dieser Methode erfolgt die Berechnung der Verbrauchswerte der jeweiligen Kostenstelle.

```
1. void Intervallwerte_in_Datenbank()
2. {
3.     DateTime IntervalTime;
4.     W_lastIntervalGas = 0;
5.     W_lastIntervalHeat = 0;
6.     W_lastIntervalWater = 0;
7.     W_lastIntervalFuel = 0;
8.     W_lastIntervalCurrent = 0;
9.     W_lastIntervalHeatQ = 0;
10.    W_lastIntervalCoolQ = 0;
11.    W_lastIntervalAir = 0;
12.
13.    try
14.    {
15.        foreach (iCostCenterCounter C in CostCenterList)
16.        {
17.            //Gas
18.            if(this.CostCenterNumber == C.CostCenter1 && C.MediumCostCenter == 0)
19.            {
20.                W_lastIntervalGas += C.W_lastIntervalCostCenter *
21.                C.AllocationKeyCostCenter1/100;
22.            }
23.
24.            else if (this.CostCenterNumber == C.CostCenter2 && C.MediumCostCenter == 0)
25.            {
26.                W_lastIntervalGas += C.W_lastIntervalCostCenter *
27.                C.AllocationKeyCostCenter2/100;
28.            }
29.
30.            else if (this.CostCenterNumber == C.CostCenter3 && C.MediumCostCenter == 0)
31.            {
32.                W_lastIntervalGas += C.W_lastIntervalCostCenter *
33.                C.AllocationKeyCostCenter3/100;
34.            }
35.
36.            else if (this.CostCenterNumber == C.CostCenter4 && C.MediumCostCenter == 0)
37.            {
38.                W_lastIntervalGas += C.W_lastIntervalCostCenter *
39.                C.AllocationKeyCostCenter4/100;
40.            }
41.
42.            else if (this.CostCenterNumber == C.CostCenter5 && C.MediumCostCenter == 0)
43.            {
44.                W_lastIntervalGas += C.W_lastIntervalCostCenter *
45.                C.AllocationKeyCostCenter5/100;
46.            }
47.
48.            //Luft
49.            if(this.CostCenterNumber == C.CostCenter1 && C.MediumCostCenter == 7)
50.            {
51.                W_lastIntervalAir += C.W_lastIntervalCostCenter *
52.                C.AllocationKeyCostCenter1/100;
53.            }
54.
55.            else if (this.CostCenterNumber == C.CostCenter2 && C.MediumCostCenter == 7)
56.            {
57.                W_lastIntervalAir += C.W_lastIntervalCostCenter *
58.                C.AllocationKeyCostCenter2/100;
59.            }
60.
61.            else if (this.CostCenterNumber == C.CostCenter3 && C.MediumCostCenter == 7)
62.            {
```

```

63.             W_lastIntervalAir += C.W_lastIntervalCostCenter *
64.             C.AllocationKeyCostCenter3/100;
65.         }
66.
67.         else if (this.CostCenterNumber == C.CostCenter4 && C.MediumCostCenter == 7)
68.         {
69.             W_lastIntervalAir += C.W_lastIntervalCostCenter *
70.             C.AllocationKeyCostCenter4/100;
71.         }
72.
73.         else if (this.CostCenterNumber == C.CostCenter5 && C.MediumCostCenter == 7)
74.         {
75.             W_lastIntervalAir += C.W_lastIntervalCostCenter *
76.             C.AllocationKeyCostCenter5/100;
77.         }
78.     }
79.
80.     IntervalTime = new DateTime(this.actYear, this.actMonth,
81.     this.actDay, this.actHour, this.actMinute, 0);
82.
83.     XAMLIB.Archive.StoreValue(this.XAMCom_W_lastIntervalGas, IntervalTime);
84.     XAMLIB.Archive.StoreValue(this.XAMCom_W_lastIntervalAir, IntervalTime);
85.     XAMLIB.TraceViewer.Trace("EMS_CostCenter: " + this.InstanceName, "Store
86.     Interval", XAMCommon.Trace.TracePrio.MESSAGE);
87.     this.Error = false;
88. }
89. catch (Exception ex)
90. {
91.     XAMLIB.TraceViewer.Trace("EMS_CostCenter: " + this.InstanceName,
92.     ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
93.     this.Error = true;
94. }
95. }

```

Quellcodeausschnitt 12: Berechnung der Kostenstellenverbräuche (eigene Darstellung)

Mittelwertbildung:

In dieser Methode erfolgt die Berechnung der Mittelwerte der vier Intervalle der letzten Stunde.

```

1. void Mittelwert_W_lastInterval_der_letzten_Stunde()
2. {
3.     List<double> doub = new List<double>();
4.     double[] arrayinterval = new double[0];
5.     double sum = 0;
6.
7.     try
8.     {
9.         IQueryable<XAMCommon.XAMRuntimeDB.XAMTrendValue> MyQuery =
10.         XAMCommon.XAMRuntimeDB.X4Archive.Query(this.InstanceName + ".W_lastInterval",
11.         DateTime.Now.AddDays(-1), DateTime.Now, XAMCommon.XAMCommonBase.X4Runtime);
12.
13.         foreach(XAMCommon.XAMRuntimeDB.XAMTrendValue entry in
14.         MyQuery.OrderByDescending(v => v.DateTime))
15.         {
16.             doub.Add(entry.Value);
17.         }
18.
19.         Array.Clear(arrayinterval, 0, arrayinterval.Length);
20.         arrayinterval = doub.ToArray();
21.
22.         foreach(double item in arrayinterval)
23.         {
24.             sum += item;
25.         }

```

```

26.
27.         if(sum != 0)
28.         {
29.             Average_W_lastInterval = sum/arrayinterval.Length;
30.         }
31.     this.Error = false;
32. }
33. catch (Exception ex)
34. {
35.     XAMLIB.TraceViewer.Trace("EMS Average: " + this.InstanceName,
36.     ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
37.     this.Error = true;
38. }
39. }

```

Quellcodeausschnitt 13: Berechnung des Mittelwerts der Intervalle der letzten Stunde (eigene Darstellung)

Automatische Ersatzwertbildung:

In dieser Methode erfolgt für die Intervallwerte das Erkennen von Lücken in der Datenbank, die rückwirkende Verbrauchsaufteilung und Speicherung der Werte in die Datenbank.

```

1.  bool Save_Data;
2.  void Intervallwerte_in_Datenbank()
3.  {
4.      DateTime IntervalTime;
5.
6.      IntervalTime = new DateTime(this.actYear, this.actMonth, this.actDay,
7.      this.actHour, this.actMinute, 0);
8.
9.      if(!this.InvalidInterval)
10.     {
11.         AutomaticCorrection = false;
12.         SaveData();
13.     }
14. }
15.
16. private async void SaveData()
17. {
18.
19.     if(Save_Data)
20.         return;
21.
22.     try
23.     {
24.         Save_Data = true;
25.         await SaveFromAsync();
26.     }
27.
28.     catch(Exception ex)
29.     {
30.         XAMLIB.TraceViewer.Trace(this.InstanceName,
31.         ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
32.     }
33.
34.     finally
35.     {
36.         Save_Data = false;
37.     }
38. }
39.
40. private System.Threading.Tasks.Task SaveFromAsync()

```

```

41. {
42.     System.Threading.Tasks.Task task = System.Threading.Tasks.Task.Run(()=>
43.     {
44.         try
45.         {
46.             using (var db = new XAMCommon.XAMRuntimeDB.X4RuntimeDB(
47.                 XAMServer.Server.Server.sc.Config.DatabaseConnectionString,
48.                 XAMServer.Server.Server.sc.Config.DatabaseUser,
49.                 XAMServer.Server.Server.sc.Config.DatabasePassword))
50.             {var Queryable=
51.                 from A in db.db.X4Archives
52.                 join I in db.db.V_AutomationControlInstances on
53.                 A.InstanceID equals I.ID
54.                 join P in db.db.V_AutomationControlProperties on
55.                 A.PropertyID
56.                 equals P.ID
57.                 where I.Name == this.InstanceName && P.Name ==
58.                 "W_lastInterval"
59.                 orderby A.DateTime descending
60.                 select A.DateTime;
61.
62.                 LastIntervalTime = Queryable.FirstOrDefault().DateTime;
63.             }
64.
65.             DateTime IntervalTime = new DateTime(this.actYear, this.actMonth,
66.                 this.actDay, this.actHour, this.actMinute, 0);
67.
68.             if (LastIntervalTime == new DateTime(0001,1,1,0,0,0))
69.             {
70.                 LastIntervalTime = IntervalTime;
71.             }
72.
73.             TimeSpan SpanInterval = IntervalTime-LastIntervalTime;
74.             double DateDiffInterval = SpanInterval.TotalMinutes;
75.             double MissingIntervals = Math.Round(DateDiffInterval/15);
76.
77.             if(DateDiffInterval > 15 && Activate_AutomaticCorrection)
78.             {
79.                 W_lastInterval = this.W_lastInterval/(MissingIntervals);
80.
81.                 double n = MissingIntervals;
82.                 do
83.                 {
84.                     DateTime IntervalTime_new =
85.                         IntervalTime.AddMinutes(-(15*n));
86.
87.                     XAMLIB.Archive.StoreValue(
88.                         this.XAMCom_W_lastInterval,IntervalTime_new);
89.                     n--;
90.                 } while ( n >= 0);
91.
92.                 XAMLIB.Archive.StoreValue(
93.                     this.XAMCom_W_lastInterval,IntervalTime);
94.                 DateDiffInterval = 0;
95.                 MissingIntervals = 0;
96.                 AutomaticCorrection = true;
97.                 XAMLIB.TraceViewer.Trace("EMS_Counter: " +
98.                     this.InstanceName, "Automatic Correction of " +
99.                     MissingIntervals + " Intervals
100.                    ",XAMCommon.Trace.TracePrio.MESSAGE);
101.             }
102.
103.             else
104.             {
105.                 XAMLIB.Archive.StoreValue(
106.                     this.XAMCom_W_lastInterval,IntervalTime);
107.             }
108.
109.             XAMLIB.Archive.StoreValue(this.XAMCom_W_lastInterval,IntervalTime);
110.             XAMLIB.Archive.StoreValue(

```

```

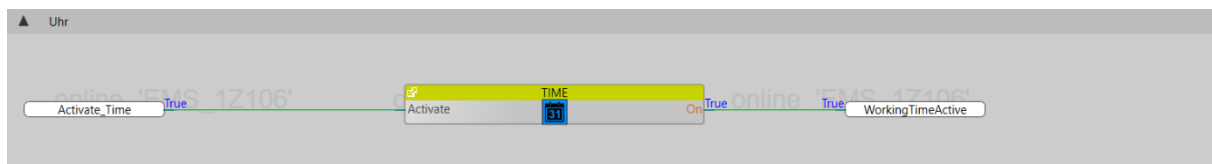
111.         this.XAMCom_ValueIn_HTintern,IntervalTime);
112.         XAMLIB.Archive.StoreValue(
113.         this.XAMCom_ValueIn_NTintern,IntervalTime);
114.         XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName,
115.         "Store Interval", XAMCommon.Trace.TracePrio.MESSAGE);
116.         this.Error = false;
117.     }
118.
119.     catch (Exception ex)
120.     {
121.         XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName,
122.         ex.ToString(), XAMCommon.Trace.TracePrio.ERROR);
123.         this.Error = true;
124.     }
125.
126.     return;
127. });
128. return task;
129. }

```

Quellcodeausschnitt 14: Ersatzwertbildung von Intervallwerten (eigene Darstellung)

Automatische Plausibilitätsprüfung:

Mithilfe von FBD wurde das Zeitschaltprogramm instanziiert.



Quellcodeausschnitt 15: Instanziierung Zeitschaltprogramm (eigene Darstellung)

In dieser Methode erfolgt die automatische Plausibilitätsprüfung des berechneten Verbrauchswertes auf Basis des Mittelwertes und des Toleranzfaktors.

```

1. Check_Plausibility = false;
2.
3. if (Activate_Plausibility && ((W_lastInterval > Average_W_lastInterval +
4. ToleranceInterval) || (W_lastInterval < Average_W_lastInterval - ToleranceInterval)))
5. {
6.     if (Activate_Time && !WorkingTimeActive)
7.     {
8.         return;
9.     }
10.
11.     else
12.     {
13.         Check_Plausibility = true;
14.         XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName,
15.         "Check Plausibility of Value " + W_lastInterval,
16.         XAMCommon.Trace.TracePrio.MESSAGE);
17.     }
18. }

```

Quellcodeausschnitt 16: Plausibilitätsprüfung von Intervallwerten (eigene Darstellung)

Kennzahlengenerierung:

In der nachfolgenden Methode erfolgt die Funktion zum Import der Messwerte über eine CSV-Datei. Diese Methode muss nicht explizit asynchron programmiert werden, da diese über ein Click-Event in der Visualisierung getriggert wird und solche Methoden von XAMControl automatisch als asynchron deklariert werden.

```
1. async void XAMButton_36_Click(object sender, RoutedEventArgs e)
2. {
3.     XAMButton Obj = sender as XAMButton;
4.     try
5.     {
6.         string SqlConnectionString = "packet size=4096;" + "user id=" +
7.         XAMServer.Server.Server.sc.Config.DatabaseUser + ";data source=" +
8.         XAMServer.Server.Server.sc.Config.DatabaseConnectionString +
9.         ";persist security info=True;initial
10.        catalog=XAMRuntimeX4;password=" +
11.        XAMServer.Server.Server.sc.Config.DatabasePasswort;
12.
13.        System.Windows.Forms.OpenFileDialog openFileDialog1 = new
14.        System.Windows.Forms.OpenFileDialog();
15.        openFileDialog1.Filter = "CSV File | *.csv";
16.
17.        if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
18.        {
19.            System.Data.DataTable _dt = SimpleCSVAdapter.Read(openFileDialog1.FileName,
20.            ';', true);
21.            System.Data.DataRow[] rows = _dt.Select();
22.
23.            using (System.Data.SqlClient.SqlConnection sqlcon = new
24.            System.Data.SqlClient.SqlConnection(SqlConnectionString))
25.            {
26.                // Connect to database.
27.                sqlcon.Open();
28.
29.                int i = 0;
30.
31.                foreach (System.Data.DataRow row in rows)
32.                {
33.                    i++;
34.
35.                    if (row[2].ToString() == "x")
36.                    {
37.                        System.Data.SqlClient.SqlCommand sqlcmd = new
38.                        System.Data.SqlClient.SqlCommand("DELETE A FROM
39.                        XAMRuntimeX4.dbo.X4Archives A " +
40.                        "INNER JOIN XAMControlX4.dbo.AutomationControlInstances as I on
41.                        I.ID = A.InstanceID " +
42.                        XAMControlX4.dbo.AutomationControlProperties as P on P.ID =
43.                        A.PropertyID " +
44.                        "Where I.Name = @instance AND P.Name = @property AND A.DateTime =
45.                        @datetime", sqlcon);
46.
47.                        sqlcmd.Parameters.AddWithValue("@datetime",
48.                        System.Convert.ToString(row[1]));
49.
50.                        sqlcmd.Parameters.AddWithValue("@instance", this.InstanceName);
51.                        sqlcmd.Parameters.AddWithValue("@property"
52.                        "ReferenceValue_Interval");
53.
54.                        sqlcmd.CommandTimeout = 180;
55.                        sqlcmd.ExecuteNonQuery();
56.                    }
57.
58.                    else if (row[0].ToString() != string.Empty)
```

```

59.         {
60.
61.             System.Data.SqlClient.SqlCommand cmd = new
62.             System.Data.SqlClient.SqlCommand("UPDATE A SET A.Value = @value
63.             From XAMRuntimeX4.dbo.X4Archives A " +
64.             "INNER JOIN XAMControlX4.dbo.AutomationControlInstances as I on
65.             I.ID = A.InstanceID " +
66.             "INNER JOIN XAMControlX4.dbo.AutomationControlProperties as P on
67.             P.ID = A.PropertyID " +
68.             "Where I.Name = @instance AND P.Name = @property AND A.DateTime =
69.             @datetime AND A.Value != @value", sqlcon);
70.
71.             cmd.Parameters.AddWithValue("@value",
72.             System.Convert.ToDouble(row[0]));
73.             cmd.Parameters.AddWithValue("@datetime",
74.             System.Convert.ToString(row[1]));
75.             cmd.Parameters.AddWithValue("@instance", this.InstanceName);
76.             cmd.Parameters.AddWithValue("@property",
77.             "ReferenceValue_Interval");
78.
79.             cmd.CommandTimeout = 180;
80.             cmd.ExecuteNonQuery();
81.         }
82.
83.     else if (row[0].ToString() == string.Empty)
84.     {
85.         System.Data.SqlClient.SqlCommand sqlCommand = new
86.         System.Data.SqlClient.SqlCommand("SELECT COUNT(*) " +
87.         "FROM XAMRuntimeX4.dbo.X4Archives as A " +
88.         "INNER JOIN XAMControlX4.dbo.AutomationControlInstances as I on
89.         I.ID = A.InstanceID " +
90.         "INNER JOIN XAMControlX4.dbo.AutomationControlProperties as P on
91.         P.ID = A.PropertyID " +
92.         "WHERE I.Name = @instance AND P.Name = @property AND A.DateTime =
93.         @datetime", sqlcon);
94.
95.         sqlCommand.Parameters.AddWithValue("@instance",
96.         this.InstanceName);
97.         sqlCommand.Parameters.AddWithValue("@property",
98.         "ReferenceValue_Interval");
99.         sqlCommand.Parameters.AddWithValue("@datetime",
100.        DateTimeOffset.Parse(System.Convert.ToString(row[1])));
101.
102.        int records = (int)sqlCommand.ExecuteScalar();
103.        if (records == 0)
104.        {
105.            System.Data.SqlClient.SqlCommand cmd = new
106.            System.Data.SqlClient.SqlCommand("INSERT INTO
107.            XAMRuntimeX4.dbo.X4Archives(Value, DateTime, InstanceID,
108.            PropertyID) "+
109.            "VALUES(@value, @datetime, (SELECT TOP 1 InstanceID FROM
110.            XAMRuntimeX4.dbo.X4Archives A INNER JOIN
111.            XAMControlX4.dbo.AutomationControlInstances I on I.ID =
112.            A.InstanceID WHERE I.Name = @instance), " +
113.            "(SELECT TOP 1 PropertyID FROM
114.            XAMRuntimeX4.dbo.X4Archives A INNER JOIN
115.            XAMControlX4.dbo.AutomationControlProperties P on P.ID =
116.            A.PropertyID WHERE P.Name = @property))", sqlcon);
117.
118.            cmd.Parameters.AddWithValue("@value",
119.            System.Convert.ToDouble(row[0]));
120.            cmd.Parameters.AddWithValue("@datetime",
121.            DateTimeOffset.Parse(System.Convert.ToString(row[1])));
122.            cmd.Parameters.AddWithValue("@instance",
123.            this.InstanceName);
124.            cmd.Parameters.AddWithValue("@property",
125.            "ReferenceValue_Interval");
126.            cmd.CommandTimeout = 180;
127.            cmd.ExecuteNonQuery();
128.        }
129.

```

```

130.         else
131.         {
132.             System.Windows.Forms.MessageBox.Show("Eintrag in Zeile "
133.             + i.ToString() + " mit dem Zeitstempel: " +
134.             DateTimeOffset.Parse(System.Convert.ToString(row[3])) + "
135.             existiert bereits!", "Error",
136.             System.Windows.Forms.MessageBoxButtons.OK,
137.             System.Windows.Forms.MessageBoxIcon.Error);
138.         }
139.     }
140. }
141.
142.     System.Windows.Forms.MessageBox.Show("Import abgeschlossen", "Info",
143.     System.Windows.Forms.MessageBoxButtons.OK,
144.     System.Windows.Forms.MessageBoxIcon.Information);
145.
146.     XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName, "Messgrößen
147.     importiert!", XAMCommon.Trace.TracePrio.MESSAGE);
148.
149. }
150. }
151.
152. }
153. catch (Exception ex)
154. {
155.     System.Windows.Forms.MessageBox.Show("Error: " + ex.ToString(), "Error",
156.     System.Windows.Forms.MessageBoxButtons.OK,
157.     System.Windows.Forms.MessageBoxIcon.Error);
158.
159.     XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName, ex.ToString(),
160.     XAMCommon.Trace.TracePrio.ERROR);
161. }
162. }
163. }
164.
165. public static class SimpleCSVAdapter
166. {
167.     public static System.Data.DataTable Read(string filename, char seperator,
168.     bool firstLineIsHeader)
169.     {
170.         System.Data.DataTable dt = new System.Data.DataTable();
171.         char[] sep = new char[] { seperator };
172.         bool columnsBuilt = false;
173.         //Datei zum lesen öffnen
174.
175.         using (System.IO.StreamReader sr = System.IO.File.OpenText(filename))
176.         {
177.             string current = null;
178.             while ((current = sr.ReadLine()) != null)
179.             {
180.                 if (columnsBuilt)
181.                     firstLineIsHeader = false;
182.
183.                 if (firstLineIsHeader && !columnsBuilt)
184.                 {
185.                     string[] colHeaders = current.Split(sep);
186.
187.                     foreach (string s in colHeaders)
188.                     {
189.                         dt.Columns.Add(s, typeof(string));
190.                     }
191.
192.                     columnsBuilt = true;
193.                 }
194.
195.                 else if (!columnsBuilt)
196.                 {
197.                     string[] colHeaders = current.Split(sep);
198.                     for (int i = 0; i < colHeaders.Length; i++)
199.                     {
200.                         dt.Columns.Add();

```



```

201.         }
202.
203.         columnsBuilt = true;
204.
205.     }
206.
207.     if (columnsBuilt && !firstLineIsHeader)
208.     {
209.
210.         string[] cells = current.Split(sep);
211.         System.Data.DataRow row = dt.NewRow();
212.
213.         for (int i = 0; i < cells.Length; i++)
214.         {
215.             row[i] = cells[i];
216.         }
217.
218.         dt.Rows.Add(row);
219.     }
220.     }
221.     }
222.     return dt;
223. }
224. }

```

Quellcodeausschnitt 17: CSV-Import von Messgrößen für EnPI-Berechnung (eigene Darstellung)

In der nachfolgenden Methode erfolgt die Funktion zum Export der Messwerte in eine CSV-Datei.

```

1. async void XAMButton_344_Click(object sender, RoutedEventArgs e)
2. {
3.     XAMButton Obj = sender as XAMButton;
4.     //Add Your Code here!
5.
6.     try
7.     {
8.         string SqlConnectionString = "packet size=4096;" + "user id=" +
9.         XAMServer.Server.Server.sc.Config.DatabaseUser + ";data source=" +
10.        XAMServer.Server.Server.sc.Config.DatabaseConnectionString +
11.        ";persist security info=True;initial
12.        catalog=XAMRuntimeX4;password=" +
13.        XAMServer.Server.Server.sc.Config.DatabasePasswort;
14.
15.        System.Windows.Forms.SaveFileDialog saveFileDialog1 = new
16.        System.Windows.Forms.SaveFileDialog();
17.        saveFileDialog1.FileName = "export.csv";
18.        saveFileDialog1.Filter = "CSV Files | *.csv";
19.
20.        if (saveFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
21.        {
22.            using (System.Data.SqlClient.SqlConnection sqlcon = new
23.            System.Data.SqlClient.SqlConnection(SqlConnectionString))
24.            {
25.                sqlcon.Open();
26.                System.Data.SqlClient.SqlCommand sqlcmd = new
27.                System.Data.SqlClient.SqlCommand("SELECT A.Value, A.DateTime " +
28.                "FROM XAMRuntimeX4.dbo.X4Archives A " +
29.                "INNER JOIN XAMControlX4.dbo.AutomationControlInstances AS I on
30.                A.InstanceID = I.ID " +
31.                "INNER JOIN XAMControlX4.dbo.AutomationControlProperties AS P on
32.                A.PropertyID = P.ID " +
33.                "WHERE I.Name = @instance AND P.Name = @property AND A.DateTime >=
34.                @datetime_start AND A.DateTime <= @datetime_end " +
35.                "ORDER BY A.DateTime desc", sqlcon);
36.
37.                sqlcmd.Parameters.AddWithValue("@property",

```

```

38.         "ReferenceValue_Interval");
39.         sqlcmd.Parameters.AddWithValue("@instance", this.InstanceName)
40.         sqlcmd.Parameters.AddWithValue("@datetime_start",
41.         DateTimeOffset.Parse(System.Convert.ToString(
42.         Export_Start_Date_Interval)));
43.         sqlcmd.Parameters.AddWithValue("@datetime_end",
44.         DateTimeOffset.Parse(System.Convert.ToString(
45.         Export_End_Date_Interval)));
46.         System.Data.SqlClient.SqlDataAdapter adp = new
47.         System.Data.SqlClient.SqlDataAdapter(sqlcmd);
48.
49.         System.Data.DataTable dt = new System.Data.DataTable();
50.         adp.Fill(dt);
51.         dt.Columns.Add("Loeschen mit x", typeof(string));
52.         System.IO.StreamWriter sw = new
53.         System.IO.StreamWriter(saveFileDialog1.FileName);
54.
55.         //headers
56.         for (int i = 0; i < dt.Columns.Count; i++)
57.         {
58.             sw.Write(dt.Columns[i]);
59.             if (i < dt.Columns.Count - 1)
60.             {
61.                 sw.Write(";");
62.             }
63.         }
64.         sw.Write(sw.NewLine);
65.         foreach (System.Data.DataRow dr in dt.Rows)
66.         {
67.             for (int i = 0; i < dt.Columns.Count; i++)
68.             {
69.                 if (!Convert.IsDBNull(dr[i]))
70.                 {
71.
72.                     if (i == 0)
73.                     {
74.                         double value = Convert.ToDouble(dr[i]);
75.                         sw.Write(value);
76.                     }
77.
78.                     else
79.                     {
80.                         sw.Write(dr[i].ToString());
81.                     }
82.
83.                 }
84.                 if (i < dt.Columns.Count - 1)
85.                 {
86.                     sw.Write(";");
87.                 }
88.             }
89.             sw.Write(sw.NewLine);
90.         }
91.         sw.Close();
92.
93.         System.Windows.Forms.MessageBox.Show("CSV gespeichert unter: " +
94.         saveFileDialog1.FileName, "Info",
95.         System.Windows.Forms.MessageBoxButtons.OK,
96.         System.Windows.Forms.MessageBoxIcon.Information);
97.
98.         XAMLIB.TraceViewer.Trace("EMS_Counter: " +
99.         this.InstanceName, "Messgrößen exportiert!",
100.         XAMCommon.Trace.TracePrio.MESSAGE);
101.
102.     }
103. }
104. }
105. catch (Exception ex)
106. {
107.     System.Windows.Forms.MessageBox.Show("Error Occurred: " + ex.Message,
108.     "Error", System.Windows.Forms.MessageBoxButtons.OK,

```

```

109.         System.Windows.Forms.MessageBoxIcon.Error);
110.
111.         XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName, ex.ToString()),
112.         XAMCommon.Trace.TracePrio.ERROR);
113.
114.     }
115. }

```

Quellcodeausschnitt 18: CSV-Export von Messgrößen für EnPI-Berechnung (eigene Darstellung)

In der nachfolgenden Methode erfolgt die Berechnung der EnPIs auf Basis der Berechnungsauswahl und den jeweiligen Archivdaten aus der SQL-Datenbank.

```

1.  async void XAMButton_724_Click(object sender, RoutedEventArgs e)
2.  {
3.      XAMButton Obj = sender as XAMButton;
4.      //Add Your Code here!
5.
6.      try
7.      {
8.          string SqlConnectionString = "packet size=4096;" + "user id=" +
9.          XAMServer.Server.Server.sc.Config.DatabaseUser + ";data source=" +
10.         XAMServer.Server.Server.sc.Config.DatabaseConnectionString +
11.         ";persist security info=True;initial
12.         catalog=XAMRuntimeX4;password=" +
13.         XAMServer.Server.Server.sc.Config.DatabasePasswort;
14.
15.         using (System.Data.SqlClient.SqlConnection sqlcon = new
16.         System.Data.SqlClient.SqlConnection(SqlConnectionString))
17.         {
18.             sqlcon.Open();
19.             System.Data.SqlClient.SqlCommand sqlcmd = new
20.             System.Data.SqlClient.SqlCommand(
21.             "DELETE A FROM XAMRuntimeX4.dbo.X4Archives A " +
22.             "INNER JOIN XAMControlX4.dbo.AutomationControlInstances as I on I.ID =
23.             A.InstanceID " +
24.             "INNER JOIN XAMControlX4.dbo.AutomationControlProperties as P on P.ID =
25.             A.PropertyID " +
26.             "Where I.Name = @instance and P.Name = @property1 and A.DateTime <=
27.             @datetime_end and A.DateTime >= @datetime_start", sqlcon);
28.
29.             sqlcmd.Parameters.AddWithValue("@datetime_start",
30.             DateTimeOffset.Parse(System.Convert.ToString(EnPI_Start_Date_Interval)));
31.             sqlcmd.Parameters.AddWithValue("@datetime_end",
32.             DateTimeOffset.Parse(System.Convert.ToString(EnPI_End_Date_Interval)));
33.             sqlcmd.Parameters.AddWithValue("@property1", "EnPI_Interval");
34.             sqlcmd.Parameters.AddWithValue("@property2", "ReferenceValue_Interval");
35.             sqlcmd.Parameters.AddWithValue("@property3", "W_lastInterval");
36.             sqlcmd.Parameters.AddWithValue("@instance", this.InstanceName);
37.
38.
39.             SqlCommand.CommandText = "Select SUM(t.Value), SUM(t.ReferenceValue), t.DateTime
40.             from " +
41.             "(SELECT a.[Value] as Value, 0 as ReferenceValue, a.[DateTime] " +
42.             "FROM [XAMRuntimeX4].[dbo].[X4Archives] A " +
43.             "INNER JOIN [XAMControlX4].[dbo].AutomationControlInstances I on I.ID =
44.             A.InstanceID " +
45.             "INNER JOIN [XAMControlX4].[dbo].AutomationControlProperties P on P.ID =
46.             A.PropertyID " +
47.             "Where I.Name = @instance and P.Name = @property2 and A.DateTime <=
48.             @datetime_end and A.DateTime >= @datetime_start " +
49.             "union all " +
50.             "SELECT 0 as Value, a.Value as ReferenceValue " +
51.             ",a.[DateTime] " +
52.             "FROM [XAMRuntimeX4].[dbo].[X4Archives] A " +
53.             "INNER JOIN [XAMControlX4].[dbo].AutomationControlInstances I on I.ID =
54.             A.InstanceID " +

```

```

55.         "INNER JOIN [XAMControlX4].[dbo].AutomationControlProperties P on P.ID =
56.         A.PropertyID " +
57.         "Where I.Name = @instance and P.Name = @property3 and A.DateTime <=
58.         @datetime_end and A.DateTime >= @datetime_start ) t GROUP BY DateTime",
59.         sqlcon);
60.
61.         sqlcmd.Parameters.AddWithValue("@datetime_start",
62.         DateTimeOffset.Parse(System.Convert.ToString(EnPI_Start_Date_Interval)));
63.         sqlcmd.Parameters.AddWithValue("@datetime_end",
64.         DateTimeOffset.Parse(System.Convert.ToString(EnPI_End_Date_Interval)));
65.         sqlcmd.Parameters.AddWithValue("@property1", "EnPI_Interval");
66.         sqlcmd.Parameters.AddWithValue("@property2", "ReferenceValue_Interval");
67.         sqlcmd.Parameters.AddWithValue("@property3", "W_lastInterval");
68.         sqlcmd.Parameters.AddWithValue("@instance", this.InstanceName);
69.
70.         System.Data.SqlClient.SqlDataAdapter adp = new
71.         System.Data.SqlClient.SqlDataAdapter(sqlcmd);
72.
73.         System.Data.DataTable dt = new System.Data.DataTable();
74.         adp.Fill(dt);
75.
76.         DateTime Interval = DateTime.Now;
77.
78.         foreach(System.Data.DataRow row in dt.Rows)
79.         {
80.             if(EnPI_Type == 0)
81.             {
82.                 if(EnPI_CalculationType == 0)
83.                 {
84.                     EnPI_Interval =
85.                     .Convert.ToDouble(row[0])/EnPI_Constant;
86.                 }
87.                 else if(EnPI_CalculationType == 1)
88.                 {
89.                     EnPI_Interval =
90.                     .Convert.ToDouble(row[0])*EnPI_Constant;
91.                 }
92.                 string interval = (row[2]).ToString();
93.                 Interval = DateTime.Parse(interval);
94.             }
95.             else if(EnPI_Type == 1)
96.             {
97.                 if(EnPI_CalculationType == 0)
98.                 {
99.
100.                     EnPI_Interval = System.Convert.ToDouble(
101.                     row[0])/System.Convert.ToDouble(row[1]);
102.                 }
103.                 else if(EnPI_CalculationType == 1)
104.                 {
105.                     EnPI_Interval = System.Convert.ToDouble(
106.                     row[0])*System.Convert.ToDouble(row[1]);
107.                 }
108.
109.                 string interval = (row[2]).ToString();
110.                 Interval = DateTime.Parse(interval);
111.             }
112.
113.             else if(EnPI_Type == 2)
114.             {
115.                 EnPI_Interval =
116.                 Regression_Coefficient*System.Convert.ToDouble(row[1]) +
117.                 Regression_Constant;
118.                 string interval = (row[2]).ToString();
119.                 Interval = DateTime.Parse(interval);
120.             }
121.
122.             System.Windows.Forms.MessageBox.Show("Kennzahlen erfolgreich
123.             generiert!", "Info", System.Windows.Forms.MessageBoxButtons.OK,
124.             System.Windows.Forms.MessageBoxIcon.Information);
125.

```

```

126.             XAMLIB.TraceViewer.Trace("EMS_Counter: " +
127.             this.InstanceName, "EnPI
128.             calculated from " + EnPI_Start_Date_Interval + " to " +
129.             EnPI_End_Date_Interval, XAMCommon.Trace.TracePrio.MESSAGE);
130.         }
131.
132.     }
133.
134. }
135.
136.
137. catch (Exception ex)
138. {
139.     XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName, ex.ToString(),
140.     XAMCommon.Trace.TracePrio.ERROR);
141.
142.     System.Windows.Forms.MessageBox.Show("Error Occurred: " + ex.Message,
143.     "Error", System.Windows.Forms.MessageBoxButtons.OK,
144.     System.Windows.Forms.MessageBoxIcon.Error);
145. }
146. }

```

Quellcodeausschnitt 19: EnPI-Berechnung (eigene Darstellung)

Die nachfolgende Methode beschreibt das Click-Event des Buttons zum Öffnen des Reports. Je nach Auswahl der Berechnungsart, wird ein anderer Report geladen.

```

1. async void XAMButton_922_Click(object sender, RoutedEventArgs e)
2. {
3.     XAMButton Obj = sender as XAMButton;
4.     //Add Your Code here!
5.
6.     if(EnPI_Type == 0 || EnPI_Type == 1)
7.     {
8.         XAMLIB.Visualization.OpenPopup("EMSControl_SQL_Monitoring_Advanced",
9.         "EnPI_PopupIntervall", this.InstanceName);
10.    }
11.
12.    else if(EnPI_Type == 2)
13.    {
14.        XAMLIB.Visualization.OpenPopup("EMSControl_SQL_Monitoring_Advanced",
15.        "EnPI_PopupIntervall_Regression", this.InstanceName);
16.    }
17. }

```

Quellcodeausschnitt 20: Reportauswahl mit Click-Event (eigene Darstellung)

Überarbeitung Mittelwertbildung:

In dieser Methode erfolgt die dynamische Anpassungen vom Suffix des Parametrierfeldes für den Zeitbereich.

```

1. void XAMEditValue_46_Timer(object sender)
2. {
3.     XAMEditValue Obj = sender as XAMEditValue;
4.     //Add Your Code here!
5.
6.     if (Average_Interval_Mode == 0)
7.     {
8.         Obj.Dimension = "Stunde/n";
9.     }
10. }

```

```

11. else if (Average_Interval_Mode == 1)
12. {
13.     Obj.Dimension = "Tage/n";
14. }
15.
16. else if (Average_Interval_Mode == 2)
17. {
18.     Obj.Dimension = "Monate/n";
19. }
20. }

```

*Quellcodeausschnitt 21: Dynamische Anpassung vom Suffix des Parametrierfeldes für den Zeitbereich
(eigene Darstellung)*

In dieser Methode erfolgt die Berechnung der Mittelwerte der Intervalle für den definierten Zeitbereich.

```

1. void Mittelwert_W_lastInterval()
2. {
3.     List<double> doub = new List<double>();
4.     double[] arrayinterval = new double[0];
5.     double sum = 0;
6.
7.     try
8.     {
9.         IQueryable<XAMCommon.XAMRuntimeDB.XAMTrendValue> MyQuery =
10.            Enumerable.Empty<XAMCommon.XAMRuntimeDB.XAMTrendValue>().AsQueryable();
11.
12.         if(Average_Interval_Mode == 0)
13.         {
14.             MyQuery = XAMCommon.XAMRuntimeDB.X4Archive.Query(this.InstanceName
15.                + ".W_lastInterval", DateTime.Now.AddHours(Average_Interval_Time),
16.                DateTime.Now, XAMCommon.XAMCommonBase.X4Runtime);
17.         }
18.
19.         else if (Average_Interval_Mode == 1)
20.         {
21.             MyQuery = XAMCommon.XAMRuntimeDB.X4Archive.Query(this.InstanceName
22.                + ".W_lastInterval", DateTime.Now.AddDays(Average_Interval_Time),
23.                DateTime.Now, XAMCommon.XAMCommonBase.X4Runtime);
24.         }
25.
26.         else if (Average_Interval_Mode == 2)
27.         {
28.             MyQuery = XAMCommon.XAMRuntimeDB.X4Archive.Query(this.InstanceName
29.                + ".W_lastInterval", DateTime.Now.AddMonths(Average_Interval_Time),
30.                DateTime.Now, XAMCommon.XAMCommonBase.X4Runtime);
31.         }
32.
33.         foreach(XAMCommon.XAMRuntimeDB.XAMTrendValue entry in
34.            MyQuery.OrderByDescending(v => v.DateTime))
35.         {
36.             doub.Add(entry.Value);
37.         }
38.
39.         Array.Clear(arrayinterval, 0, arrayinterval.Length);
40.         arrayinterval = doub.ToArray();
41.
42.         foreach(double item in arrayinterval)
43.         {
44.             sum += item;
45.         }
46.         if(sum != 0)
47.             Average_W_lastInterval = sum/arrayinterval.Length;
48.
49.         this.Error = false;

```

```
50. }
51.
52. catch (Exception ex)
53. {
54.     XAMLIB.TraceViewer.Trace("EMS_Counter: " + this.InstanceName, ex.ToString(),
55.     XAMCommon.Trace.TracePrio.ERROR);
56.     this.Error = true;
57. }
58. }
```

*Quellcodeausschnitt 22: Berechnung des Mittelwerts der Intervalle des definierten Zeitbereichs
(eigene Darstellung)*

Anhang D

In diesem Anhang befindet sich der Interviewleitfaden sowie die transkribierten ExpertInneninterviews.

Interviewleitfaden:

Begrüßung:

- Kurze Einleitung
- Zusicherung der Anonymität
- Einverständnis zur Aufzeichnung

Fragestellungen:

- Frage 1:
Wurde die Online-Wetterdatenabfrage Ihrer Meinung nach adäquat umgesetzt?
- Frage 2:
Wurde die Berechnung der Heiz- und Kühlgradtage Ihrer Meinung nach adäquat umgesetzt?
- Frage 3:
Wurde der MSCONS-Export Ihrer Meinung nach adäquat umgesetzt?
- Frage 4:
Wurde der Zählerreset nach Ihrer Meinung nach umgesetzt?
- Frage 5:
Wurde die freie Einheitendefinition Ihrer Meinung nach adäquat umgesetzt?
- Frage 6:
Wurde die Kostenstellenzuordnung Ihrer Meinung nach adäquat umgesetzt?
- Frage 7:
Wurden die neuen Reports mit Liniendiagramm Ihrer Meinung nach adäquat umgesetzt?
- Frage 8:
Wurde die Mittelwertbildung Ihrer Meinung nach adäquat umgesetzt?
- Frage 9:
Wurde die automatische Ersatzwertbildung Ihrer Meinung nach adäquat umgesetzt?
- Frage 10:
Wurde die automatische Plausibilitätsprüfung Ihrer Meinung nach adäquat umgesetzt?

- Frage 11:
Wurde die Kennzahlengenerierung und -darstellung Ihrer Meinung nach adäquat umgesetzt?
- Frage 12:
Sehen Sie im EMSControl noch Verbesserungspotenzial? Haben Sie ansonsten allgemeine Anmerkungen zum EMSControl?

Abschluss:

- Bedanken für die Unterstützung

Interview 1 am 16.02.2022 über Microsoft Teams:

Stakeholder 1: Teamleiter Technische IT eines Salzwerks

Autor (A): Guten Tag Herr XXX. Vielen Dank, dass sie sich die Zeit für das Interview genommen haben.

ExpertIn (E): Sehr gerne.

A: In meiner Masterarbeit wird ihr Name und ihr Unternehmen natürlich anonymisiert. Sind sie damit einverstanden, wenn ich das Interview aufzeichne?

E: Ja, das ist für mich kein Problem.

A: Wie in meiner Mail vom 20.01.2022 erwähnt, habe ich die Klassen überarbeitet und ihnen diese per Wettransfer übermittelt. Sie haben mir dann rückgemeldet, dass sie bereits einige ACCs getestet haben und mir gerne ein Feedback dazu geben würden. Ich hoffe, die beigelegte Kurz-Doku war für sie ausreichend, um meine Änderungen nachvollziehen zu können.

E: Ja, es war für mich alles gut verständlich.

A: Gut, dann würde ich am besten gleich starten. Ich gehe nun nach der Reihe alle Änderungen durch und würde sie bitten, dazu jeweils ein kurzes und prägnantes Feedback zu geben. Wichtig ist mir vor allem, ob die neuen Klassen praxistauglich sind. Im ersten Schritt habe ich ein Wetter ACC erstellt, bei welchem Online-Wetterdaten ausgelesen werden können. Wurde dies ihrer Meinung nach adäquat umgesetzt?

E: Ich habe das ACC getestet und finde es gut. Die Abfragen haben tadellos funktioniert. Zwar wird dieses bei uns nicht benötigt, da wir ja die Außenfühler besitzen, aber ich kann es mir gut als Alternative zum Hardwarefühler vorstellen.

A: Das freut mich. Kommen wir zu Frage 2. Wurde ihrer Meinung nach die Witterungsbereinigung adäquat umgesetzt?

E: Bisher haben wir die Heizgradtage immer vom deutschen Wetterdienst verwendet. Diese werden aber sehr zeitversetzt ins Netz gestellt. Ich finde die Berechnungen im XAMControl super. Wir können sie dann gleich exportieren und für die Analyse nutzen.

A: Perfekt. Wurde der MSCONS-Export für sie adäquat umgesetzt?

E: Mir sagt MSCONS leider nichts. Ich habe bei uns intern nachgefragt, ob meine Kollegen dazu schon mal etwas gehört haben. Allerdings kann niemand bei uns im Unternehmen damit etwas anfangen. Ich habe daher dieses Export-Tool nicht getestet. Tut mir leid.

A: Kein Problem. Fahren wir fort. Ich habe nun auch die Zähler umprogrammiert, damit man sie leicht zurücksetzen kann. Wurde das für sie gut umgesetzt?

E: Ich habe dich Funktion getestet und sie hat funktioniert. Für meine Testzähler ist das ein super Feature. Das mit dem Löschen der Instanzen hat mich schon immer sehr gestört.

A: Dann haben sich meine Umprogrammierung ja gelohnt. Falls sie es gesehen haben, gibt es nun auch die Möglichkeit, die Einheiten frei zu definieren. Was halten sie von diesem Feature?

E: Da wir bei uns nur Stromzähler verwenden, benötigen wir dieses Feature nicht. Aber ich denke, es ist sicher sinnvoll, diese Möglichkeit zu haben. Die Funktion ist aber gegeben, das kann ich rückmelden.

A: Ist ihrer Meinung nach die Kostenstellenzuordnung adäquat umgesetzt?

E: Ja. Die Summierung der Verbrauchswerte für jede Kostenstelle ist sehr vorteilhaft. Das erspart uns sehr viel Zeit in der Kostenrechnung, da wir schon die berechneten Werte fix fertig exportieren können.

A: Alles klar, das klingt erfreulich. Kommen wir zur nächsten Frage. Ich habe die Reports dahingehend angepasst, dass auch ein Liniendiagramm über ein Dropdown-Feld angezeigt werden kann. Was halten Sie von der Umsetzung?

E: Das Dropdown-Feld ist klasse, so können wir dynamisch die Ansicht wechseln. Viele meiner Kollegen haben die Linien lieber als die Balken. Es ist auch gut, dass man im Tagesreport zwischen Intervallwerte und Stundenwerte switchen kann.

A: Ja genau, das habe ich ebenfalls implementiert. Falls hier weitere Diagrammtypen gewünscht sind, bitte einfach Bescheid geben. In den Zählern werden nun auch Mittelwerte gebildet. Bisher wurden nur Summenwerte und Extremwerte angezeigt. Wurde dies ihrer Meinung nach adäquat umgesetzt?

E: Grundsätzlich ja. Ich würde allerdings vorschlagen, dass der Zeitbereich für die Mittelwertbildung frei definiert werden kann.

A: Danke für den Input. Ich werde es mir ansehen und Rückmeldung geben. Als weiteren Punkt habe ich eine automatische Ersatzwertbildung in die Zählerklassen programmiert. Haben sie die Funktion getestet und können mir zur Umsetzung ein Feedback geben?

E: Ja, ich habe die Ersatzwertbildung mit meinen Testzählern getestet und es hat einwandfrei funktioniert. Durch die Alarmmeldung in der Iris und dem Logeintrag im TraceViewer sehe ich auch sofort, wann Ersatzwerte gebildet wurden.

A: Perfekt. Ich habe auch eine automatische Plausibilitätsprüfung durchgeführt. Was halten sie von dieser Implementierung?

E: Da sich der Vergleichswert auf den Mittelwert bezieht ist es eben wichtig, dass der Zeitbereich für die Mittelwertbildung wie erwähnt eingestellt werden kann. Die Funktion hat aber auch in der jetzigen Version funktioniert.

A: Alles klar. Die Mittelwertbildung werde ich mir also nochmals gut ansehen. Danke für den Hinweis. So wir kommen schon fast zum Ende des Interviews. Als letztes Feature habe ich die Möglichkeit zur Kennzahlenbildung geschaffen. Kann dies auch in der Praxis so eingesetzt werden?

E: Ich finde es gut, dass es hier drei Auswahlmöglichkeiten zur Berechnung gibt. Wir haben bisher unsere Kennzahlen in Excel berechnet. Bei uns werden als Referenz die produzierten Mengen herangezogen. Diese kann ich nun einfach in das System importieren und die Kennzahlen berechnen. Ich bekomme dann gleich die Reports fertig raus. Das ist super.

A: Freut mich zu hören. Grundsätzlich sind wir nun mit dem Interview zu den Programmierungen durch. Haben Sie noch etwaige allgemeine Anmerkungen zum EMSControl?

E: Eine Sache wäre mir wichtig. Aktuell ist ihr System ja nicht mehr zertifiziert. Für uns ist es sehr wichtig, dass sie zeitnah das System auf die aktuelle ISO Norm zertifizieren. Die Erfüllung der Norm ist uns wegen den Umlagen sehr wichtig und muss erfüllt werden. Wenn wir überprüft werden, können wir ohne aktive Zertifizierung große Probleme bekommen. Ansonsten habe ich keine Anmerkungen. Danke für die tolle Umsetzung.

A: Dazu kann ich sagen, dass wir noch heuer die Zertifizierung des Systems anstreben. Ich bedanke mich recht herzlich für ihre Zeit und wünsche noch einen schönen Tag.

Interview 2 am 18.02.2022 über Microsoft Teams:

Stakeholder 2: Teamleiter Instandhaltung Elektro, GLT und Energiemanagement eines Energieversorgers

Autor (A): Hallo XXX. Schön, dass es nun endlich mit dem Termin geklappt hat. Ganz kurz vorab. Ich werde unser Interview anonymisieren und es aufzeichnen. Ich hoffe, das ist für dich okay?

ExpertIn (E): Kein Thema.

A: Letzte Woche habe ich dir die Klassen geschickt. Konntest du schon alle testen?

E: Soweit ich weiß, habe ich alles durchprobiert.

A: Klasse. Dann starten wir gleich mit der ersten Frage, hätte ich gesagt. Funktionierte das Online-Wetter ACC in deiner Solution?

E: Leider nicht auf Anhieb. Aber das ist nicht dem ACC verschuldet. Unser Proxy hat die Seite gesperrt und deshalb hat die Abfrage nicht funktioniert. Ich habe die Seite dann von der IT freischalten lassen und dann hat es sofort geklappt. Wir haben in unserer GLT einen Hardwarensensor. Deshalb ist dieser Anwendungsfall für die Zähler nicht wirklich relevant für uns. Aber ich habe schon andere Ideen. Wie in unserem damaligen Meeting besprochen, können wir nun damit die Regelungen für das Nachtlüften anpassen und die Vorhersagewerte verwenden. Aber dazu würde ich gerne einen gesonderten Termin mit dir vereinbaren.

A: Alles klar. Das klingt gut, können wir natürlich gerne machen. Was hältst du von der Berechnung der Heiz- und Kühlgradtage?

E: Ich finde, das hast du sehr gut umgesetzt. Wir lassen die Klasse jetzt mal laufen und vergleichen dann die Werte mit denen aus dem Internet. Bin gespannt, was da so rauskommt. Aber ich kann mir gut vorstellen, dass wir eure berechneten Werte in Zukunft verwenden.

A: Gib mir dann bitte eine Rückmeldung dazu. Kommen wir gleich zur nächsten Frage. Hat der MSCONS-Export funktioniert? Dieser war ja für euch sehr wichtig.

E: Ja genau. Wir benötigen die Zählerdaten für die Abrechnung in diesem Format. Nach der Parametrierung der Zähler habe ich den Export aktiviert und ein paar Tage laufen lassen. Die Dateien habe ich dann an unseren MSCONS-Experten weitergeleitet. Dieser bestätigte mir, dass der Export perfekt funktioniert. Ich bedanke mich hier schon mal recht herzlich. Das erspart uns jetzt sehr viel Zeit.

A: Da bin ich aber sehr froh. Danke für das Lob. Wurde deiner Meinung nach das Zurücksetzen der Zähler adäquat umgesetzt?

E: Das funktionierte problemlos.

A: Sehr schön. Bei den Zählern kannst du die Einheiten jetzt zusätzlich definieren. Wurde dies praxistauglich umgesetzt?

E: Wir haben im Unternehmen nur Zähler mit der Einheit kWh. Deshalb hatten wir nie das Problem, dass wir eine andere Einheit benötigten. Ich habe es ehrlich gesagt auch nicht getestet.

A: Okay verstehe, ist kein Problem. Hast du die Kostenstellenzuordnung getestet? Welches Feedback kannst du mir hierzu geben? Reichen euch vor allem die fünf Kostenstellen zum Definieren aus?

E: Ja, habe ich getestet. Die fünf Kostenstellen reichen aus ja. Die Verbräuche eines Zählers werden bei uns nicht auf mehr als zwei Kostenstellen aufgeteilt. Und ich kann im System ja so viele Kostenstellen anlegen, wie ich möchte. Das passt gut so.

A: Danke für das Feedback. Bei unseren Standardreports ebenfalls angepasst und eine Auswahl zwischen Linien- und Balkendiagrammen eingepflegt. Wurde dies adäquat umgesetzt?

E: Die Umschaltung funktioniert reibungslos. Allerdings muss ich sagen, dass für unsere Auswertungen der Trendviewer und die Balkendiagramme ausgereicht haben.

A: Okay, verstehe. In den Zählern werden nun auch Mittelwerte gebildet. Wie siehst du für euch die Umsetzung?

E: Ich habe in der Dokumentation gesehen, dass du die Zeiten für die Mittelwertbildung fix reinprogrammiert hast. Hier wäre es sicher von Vorteil, wenn wir diese selbst parametrieren könnten.

A: Ich werde es mir notieren, danke für den Hinweis. Die neuen Zählerklassen können auch automatisch Ersatzwerte bilden, wenn es zu Störungen gekommen ist. Wie wurde deiner Meinung nach die Funktion umgesetzt?

E: Das ist eine essenzielle Funktion, damit wir die Integrität haben. Es war sehr wichtig, dass du die Funktion eingebaut hast. Es hat bei meinen Tests hervorragend funktioniert.

A: Freut mich sehr. Was hältst du von der automatischen Plausibilitätsprüfung?

E: Ist eine tolle Funktion. So fallen gleich eigenartige Werte auf, welche wir möglicherweise in der Analyse übersehen.

A: Es besteht jetzt auch die Möglichkeit, direkt im System Kennzahlen generieren zu lassen. Ist die Funktion so für euch einsetzbar?

E: Bis dato haben wir noch nicht so viel mit Kennzahlen gemacht. Die Auswertungen haben gut funktioniert. Ich denke, dass wir durch die neue Möglichkeit uns ein Konzept überlegen können, wie wir das Feature in Zukunft sinnvoll einsetzen können.

A: Okay, perfekt. Zum Abschluss des Interviews würde ich gerne fragen, ob du bzw. ihr noch allgemeine Anmerkungen oder Verbesserungswünsche unseres EMSControl habt?

E: Die automatische Ersatzwertbildung und der MSCONS-Export waren uns sehr wichtig. Das ist klasse, dass das nun im System ist und so gut funktioniert. Da eure Zertifizierung abgelaufen ist, solltet ihr bitte demnächst eine Neuzertifizierung nach ISO 50001 anstreben.

A: Danke für das Feedback. Wir haben in diesem Jahr noch vor, das System neu zu zertifizieren. Danke für die Unterstützung bei der Umsetzung. Ich glaube, wir konnten dadurch gemeinsam das System auf den aktuellen Stand bringen. Wir hören uns bestimmt in nächster Zeit. Bis dahin alles Gute.

Interview 3 am 18.02.2022 über Microsoft Teams:

Stakeholder 3: Teamleiter eines Glücksspielunternehmens

Autor (A): Hallo XXX. Danke für den kurzfristigen Termin. Nur zur Info. Das Interview wird in der Masterarbeit natürlich anonymisiert. Ist es dir recht, wenn ich es aufzeichne? Dann tue ich mich beim Transkribieren im Anschluss leichter.

ExpertIn (E): Jaja, das stellt für mich kein Problem dar.

A: Wie ich schon aus deinen gemailten Screenshots gesehen habe, hast du dich bereits ausführlich mit den neuen EMS Klassen beschäftigt. Wie vorab gesagt, werde ich im Interview alle Änderungen, die ich in den Klassen gemacht habe, Schritt für Schritt durchgehen und dich bitten, ein kurzes Feedback dazuzugeben. Ist die Vorgehensweise für dich so in Ordnung?

E: Natürlich.

A: Gut, dann starten wir mit der ersten Frage hätte ich gesagt. Wurde deiner Meinung nach die Online-Wetter Klasse adäquat umgesetzt?

E: Ich hatte anfangs das Problem, dass unser Ort nicht gefunden wurde. Aber nach mehreren Versuchen, mit verschiedensten Schreibweisen, hat es dann geklappt. Aber dafür könnt ihr nichts, der Dienst stellt die Orte nur in einer gewissen Schreibweise zur Verfügung. Ansonsten passten die Temperaturwerte gut mit unserem Temperatursensor überein. Wie du aber weißt, haben wir sowieso einen Sensor für die Regelungen verbaut und brauchen dieses ACC eigentlich nicht. Trotzdem hat es gut funktioniert und kann sicher für andere Kunden interessant sein.

A: Ja genau, die Schreibweise vom Ort kann ich nicht beeinflussen, den gibt OpenWeather vor. Fahren wir fort. Ist deiner Meinung nach die Berechnung der Heiz- und Kühlgradtage passend umgesetzt geworden? Kommen wir zur zweiten Frage.

E: Ja, das passt gut. Ich habe schon ein paar Tage aufzeichnen lassen und werde die Werte gegenchecken. Bisher haben wir die Werte vom Meteorologieinstitut genutzt. Diese sind leider kostenpflichtig. Wenn eure Werte mit unseren zusammenpassen, werden wir in Zukunft sicher die kostenpflichtige Variante aufkündigen.

A: Gib mir bitte Bescheid, wenn du Vergleichswerte hast. Was hältst du vom MSCONS-Export funktioniert?

E: Dazu kann ich nichts sagen. Ich kenne dieses Format nicht und wir benötigen es auch nicht.

A: Das verstehe ich natürlich, dass du dir damit dann nicht beschäftigt hast. Die Werte der Zählerinstanzen können nun auch vollständig zurückgesetzt werden. Hast du dies auch getestet?

E: Die Funktion wird eigentlich nur gebraucht, wenn man ein Projekt neu startet und nach den Tests alles zurücksetzen möchte, bevor es in den Livebetrieb geht. Da wir aktuell nicht vorhaben umzubauen, ist die Funktion für uns nicht so wichtig. Trotzdem habe ich die Funktion mit einer neuen Instanz getestet. Alle Werte wurden ohne Probleme aus der Datenbank entfernt.

A: Danke für das Testen. Ich habe bei den Zählern die Möglichkeit geschaffen, die Einheiten beliebig zu definieren. Wurde dieses Feature adäquat umgesetzt?

E: Ja funktioniert.

A: Super. Jedem Zähler kann nun auch eine Kostenstelle zugeordnet werden. Kann die Klasse so in der Praxis eingesetzt werden?

E: Ja kann sie. Wurde sehr gut gelöst.

A: Prima. In den Reports können die Daten nun auch in einem Liniendiagramm dargestellt werden. Man kann sehr einfach zwischen den Darstellungen wechseln. Wurde dies passend umgesetzt?

E: Ja, funktioniert super.

A: Sehr gut. Konntest du die Mittelwertbildung auch testen?

E: Habe ich getestet, funktioniert. Mir persönlich sind die Mittelwerte aber nicht so wichtig. Aber je mehr Informationen, desto besser.

A: Alles klar. In den Zählern werden nun automatisch Ersatzwerte bei Ausfällen gebildet. Wurde dies adäquat umgesetzt?

E: Ja. Hat gänzlich funktioniert. Ich habe verschiedenste Szenarien getestet. Die Werte wurden immer richtig aufgeteilt und die Logeinträge stimmten auch.

A: Perfekt. Wie hat die automatische Plausibilitätsprüfung funktioniert?

E: Funktion ist gegeben. Ich finde, sie ist allerdings nur einsetzbar, wenn die Verbräuche, wie bei uns ziemlich konstant sind. Bei großen Schwankungen macht es keinen Sinn, da die Prüfung dann immer anschlagen würde. Ich finde es deshalb gut, dass man sie beliebig aktivieren kann.

A: Da hast du recht, ja. Du hast ja einmal angemerkt, dass ihr über das System Kennzahlen generieren wollt. Ich habe nun diese Funktion mit drei verschiedenen Berechnungsmethoden implementiert. Wurde dies deiner Meinung nach praxistauglich umgesetzt?

E: Ich finde deine Lösung mit der Regressionsgleichung gut. An das hatten wir noch gar nicht gedacht. Wir werden jetzt einmal ein paar Referenzwerte sammeln und dann richtige Analysen durchführen. Den Test habe ich mit fiktiven Daten gemacht und der hat schon mal gut funktioniert.

A: Das freut mich, zu hören. Gibt mir dann bitte eine Rückmeldung, wenn ihr mit den Tests durch seid. Vielleicht habt ihr dann noch ein paar wichtige Erkenntnisse für mich.

E: Mache ich.

A: Kommen wir zur letzten Frage. Gibt es sonst noch irgendwelche Anmerkungen zum EMSControl oder seid ihr mit der neuen Version zufrieden?

E: Ich bin neugierig, wie das mit den Kennzahlen in Zukunft funktioniert. Wie gesagt, mir gefällt die Regressionsgleichung sehr gut und hat uns einen Denkanstoß gegeben. Ansonsten habe ich eigentlich nichts anzumerken.

A: Prima. Dann bedanke ich mich für die Hilfe zur Verbesserung des Produkts und wünsche dir noch einen schönen Feierabend.

Interview 4 am 19.02.2022 via Webex:

Stakeholder 4: Electrical Engineer Maintenance eines Herstellers von Halbfabrikaten aus Kupfer und Kupferlegierungen

Autor (A): Guten Tag Herr XXX. Kurz vorweg. Sind sie mit der Aufzeichnung des Interviews einverstanden?

ExpertIn (E): Ja.

A: Dann starte ich gleich mal die Aufzeichnung. Ich werde das Interview auch anonymisieren. Kurz zur Agenda. Wie sie in der übermittelten Doku bestimmt gesehen haben, wurden einige Anpassungen im EMSControl durchgeführt. Die neuen ACCs haben sie zum Testen bereits erhalten. Im Zuge des Interviews würde ich gerne von Ihnen erfragen, ob die neuen Features praxistauglich einsetzbar sind und ich sie ohne Bedenken in den Store laden kann. Ich würde um kurze und prägnante Antworten bitten. Wenn sie bereits sind, würde ich dann gleich mit der ersten Frage starten.

E: Kann losgehen.

A: Sehr schön. So Frage 1 lautet. Wurde ihrer Meinung nach die Online-Wetterdatenabfrage adäquat umgesetzt?

E: Aus meiner Sicht ja.

A: Alles klar, danke für das Testen. Haben sie sich die Berechnung der Heiz- und Kühlgradtage angeguckt?

E: Habe ich. Berechnungen passen meines Erachtens.

A: Danke für die Info. Haben Sie sich den MSCONS-Export auch angesehen?

E: Nicht wirklich. Wir kennen dieses Nachrichtenformat nicht.

A: Das habe ich mir fast gedacht. Haben Sie auch den neuen Reset-Button getestet?

E: Ja funktionierte. Habe ich in unserer Test-PLC ausprobiert.

A: Sehr gut. Ihr Feedback aus dem vorigen Jahr habe ich nun auch implementiert. In jedem Zähler kann nun die Einheit frei definiert werden. Ich hoffe, ich konnte es passend umsetzen.

E: Habe ich gesehen, danke funktioniert. In den aktuellen Zählern können wir das jetzt zwar nicht mehr anpassen, da die Vergangenheitswerte mit der Einheit nicht mehr übereinstimmen würden. Ich denke aber, dass es sicher in vielen Situationen brauchbar ist.

A: Das ist verständlich. Fahren wir fort. Wurde die Kostenstellenzuordnung praxistauglich umgesetzt? Ist die Aufteilung der Werte auf maximal fünf Kostenstellen ausreichend?

E: Passt gut so. Wichtig ist, dass wir so viele Kostenstellen anlegen können, wie wir wollen. Das ist gegeben. Die Aufteilung der Verbräuche eines Zählers auf mehr als fünf Kostenstellen kommt eigentlich bei uns nie vor.

A: Verstehe. Danke für die Information. Ein weiteres Feedback von ihnen habe ich nun auch umgesetzt. Sie können bei den Standardreports in den Zählern nun einfach zwischen der Darstellung von Balken und Linien wählen. Ich hoffe, ich konnte damit ihre Anforderung praxistauglich umsetzen.

E: Gefällt mir sehr gut. Die dynamische Auswahlmöglichkeit ist klasse.

A: Danke. Im Popup der Zähler werden nun auch Mittelwerte abgebildet. Was halten sie von dieser Umsetzung?

E: Könnten sie noch darstellen, wie sich der Mittelwert zusammensetzt?

A: Hier wurde ich bereits aufmerksam gemacht, dass der Zeitbereich frei definierbar sein soll. Ich werde das Popup also noch ein wenig anpassen. dann wird dies verständlicher sein.

E: Kling gut.

A: Eine Anforderung war das Bilden von automatischen Ersatzwerten. Konnte dies adäquat umgesetzt werden?

E: Es ist gut, dass man die Funktion aktivieren und deaktivieren kann. In der Simulation hat es ohne irgendwelche Probleme funktioniert. Hat mir gut gefallen.

A: Prima. Konnten sie auch die automatische Plausibilitätsprüfung überprüfen?

E: Wir werten unsere Daten sowieso monatlich selbst aus. Ich denke aber, dass es doch sehr hilfreich sein kann, wenn schon das System vorab eigenartige Werte aufdeckt. Es ist gut, dass man die Funktion aktivieren und deaktivieren kann.

A: Sehe ich gleich. Danke für das Testen. Zu guter Letzt habe eine Kennzahlengenerierung erstellt. Wurde dies für Praxis adäquat umgesetzt?

E: Es funktionierte einwandfrei. Ich habe alle Varianten ausprobiert.

A: Freut mich sehr zu hören. Vielen Dank. Das war es nun auch schon mit den Fragen zu den jeweiligen Anpassungen im System. Hätten sie noch allgemeine Anmerkungen?

E: Bitte schleunigst die ISO Zertifizierung vornehmen.

A: Ist bereits auf unserer Agenda für das Jahr 2022.

E: Das hört sich ja vielversprechend an.

A: Mit den neuen Funktionen sollte einer positiven Zertifizierung auch nichts im Wege stehen. Vielen Dank für das Interview und ihre Hilfe.