

Software-Qualität

Hausarbeit

im Rahmen der Prüfung zum

Bachelor of Science (B.Sc.)

des Studienganges

Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Philipp Reichert

Matrikelnummer, Kurs: 1758822, TINF20B2

Gutachter der Dualen Hochschule: Dennis Kube & Jonathan Schwarzenböck

Inhaltsverzeichnis

Abbildungsverzeichnis	II
1 Einleitung	1
2 Hauptteil	3
2.1 Priorität des Projekts	3
2.1.1 Ressourcen	3
2.2 Aktuelles Vorgehen, dessen Hintergründe und Auswirkungen	4
2.2.1 Systemarchitektur	4
2.2.2 Auslieferung	5
2.2.3 Software Tests	5
2.2.4 Projektmanagement	6
2.2.5 Dokumentation und Versionskontrolle	6
2.2.6 Zusammenfassung der Probleme	7
2.3 Grundlagentheorie für Optimierung	7
2.3.1 PDCA	7
2.3.2 Wasserfall Modell und Kanban Board	8
2.4 Optimierungen und Marktspezifische Bewertung	9
2.4.1 Awareness für Sinnhaftigkeit von Planung schaffen	9
2.4.2 Einführung moderner Versionierungs- und Projektmanagement Tools	9
2.4.3 Testgetriebene Entwicklung mit automatisierten Tests	10
3 Zusammenfassung	12
Literaturverzeichnis	III

Abbildungsverzeichnis

1.1	Die Abbildung zeigt die Stellen, die Swisdec standardisierte Lohndaten entgegennehmen [1]	1
-----	---	---

1 Einleitung

Das Hauptziel dieser Hausarbeit ist es, das Softwarequalitätsmanagement (SQM) anhand eines konkreten Fallbeispiels zu untersuchen und Verbesserungen dafür zu entwickeln. Das Fallbeispiel bezieht sich auf die Umsetzung eines neuen Lohnstandards und den Erhalt der damit verbundenen Swissdec-Zertifizierung. Durch diese Ausarbeitung soll der Erhalt theoretischen und praktischen Wissens nachgewiesen werden, insbesondere das Verständnis für die praktische Anwendung des erlernten Wissens.

Elektronische Lohnabrechnungen sind ein integraler Bestandteil moderner Enterprise Resource Planning (ERP)-Software für den Bereich Human Resources (HR). Auch meine Firma bietet ERP-Systeme an, die die Möglichkeit zur elektronischen Lohnabrechnung bieten ¹. In Lohnabrechnungen vorkommende Lohndaten sind wichtige personenbezogene Informationen, die je nach Land unterschiedlichen Standards und Vorschriften entsprechen müssen. Darüber hinaus müssen sie an verschiedene Stellen und Behörden gemeldet werden, wie beispielsweise Ämtern und Versicherungen.

Um eine einheitliche und rechtlich konforme Abwicklung zu gewährleisten, hat das gemeinschaftliche Projekt Swissdec einen Standard für die Schweiz entwickelt und vergibt Zertifizierungen zur Einhaltung dieses Standards [1]. Abbildung 1.1 zeigt, dass viele bedeutende schweizerische Institutionen Swissdec Standards akzeptieren. Daher ist es wichtig, dass ERP-Systeme die aktuellen Swissdec Standards unterstützen.

¹In der Schweiz wird üblicherweise von Lohn statt Gehalt gesprochen

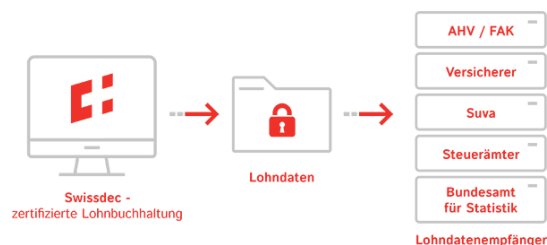


Abbildung 1.1: Die Abbildung zeigt die Stellen, die Swissdec standardisierte Lohndaten entgegennehmen [1]

Bisher werden die Standards bis ELM 4.0 unterstützt. Das Ziel dieses Projekts besteht darin, die während des Projektzeitraums aktuellste Zertifizierung für die Verarbeitung von Lohndaten, ELM 5.0, in das bestehende ERP-System zu integrieren. Wesentliche Schritte dafür sind die Verbesserung der Qualität des existierenden Codes, die Anpassung der generierten XML-Struktur sowie die Implementierung der für die Zertifizierung benötigten Verschlüsselungs- und Datenschutzverfahren.

2 Hauptteil

2.1 Priorität des Projekts

Die standardisierte Datenkommunikation spielt eine wichtige Rolle bei der Zusammenarbeit mehrerer getrennt verwalteter Institutionen und gewährleistet einen reibungslosen Ablauf von Prozessen. Da sich Prozesse im Laufe der Zeit weiterentwickeln, müssen auch die Kommunikationsstandards gelegentlich angepasst werden.

Die Unterstützung der neuesten Standards ist eine grundlegende funktionale Anforderung für ERP-Software, da sie die aktuelle und zukünftige reibungslose Kommunikation mit anderen Institutionen gewährleistet. Die Umsetzung dieser Anforderung ist von großer Bedeutung für das Gesamtprodukt.

Um zu zeigen, dass die Richtlinien unterstützt werden, soll das entsprechende Zertifikat von Swissdec erworben werden. Dadurch wird das Qualitätsmanagement vereinfacht, da einheitliche Vorgaben eingehalten werden, es sorgt für Transparenz bei Kunden und bietet eine objektive Überprüfung des Erfolges des Projekts

Es ist jedoch wichtig zu beachten, dass dieses Qualitätsmerkmal erst bei Abschluss des Projekts erworben werden kann und nichts über den tatsächlichen Verlauf des Projekts aussagt. Daher sind weitere Maßnahmen im Softwarequalitätsmanagement erforderlich.

2.1.1 Ressourcen

Das Projektteam besteht aus zwei erfahrenen Entwicklern, die bereits seit über 10 Jahren an der Software arbeiten, einem Junior-Entwickler und einem dualen Studenten. Ihre Aufgaben umfassen sowohl die Implementierung des neuen Standards als auch die Wartung der bestehenden Lohndatenabrechnung.

Das Entwicklungsteam wird von zwei Personen unterstützt, die sich mit Fragen zur Lohndatenabrechnung, den rechtlichen Bestimmungen in der Schweiz und Rückfragen an

Swissdec befassen. Ihr Verantwortungsbereich umfasst auch manuelle Ende-zu-Ende-Tests für durch Rückfragen abgeklärte Randfälle.

Der Zeitliche Rahmen für die Umsetzung des Projektes liegt bei zwei Jahren.

2.2 Aktuelles Vorgehen, dessen Hintergründe und Auswirkungen

2.2.1 Systemarchitektur

Die Codebasis erstreckt sich über eine Vielzahl von Systemen, die unabhängige Entwicklungsstände und Testdaten aufweisen. Es gibt vier Entwicklungssysteme, auf denen der neue Lohnmeldestandard implementiert werden soll. Diese unterscheiden sich in den Implementierungsdetails, da sie jeweils verschiedene ABAP-Versionen verwenden.

Der Transport von Versionsänderungen zwischen den Systemen ist teilweise automatisch möglich, kann jedoch bis zu 40 Minuten dauern und erfordert möglicherweise manuelle Codeanpassungen.

Für jedes Entwicklungssystem gibt es mindestens ein Testsystem. Es können auch mehrere Testsysteme mit unterschiedlichen Systemeinstellungen und Testdaten vorhanden sein. Die Anzahl der Testsysteme hängt davon ab, wie viele Kunden die entsprechende ABAP-Version mit verschiedenen Systemeinstellungen nutzen.

Diese Vielfalt an Systemen ergibt sich aus der SAP-Politik, alte Versionen weit über ihr technologisches Haltbarkeitsdatum hinaus zu unterstützen.

Diese verteilte Entwicklung und Testung beeinträchtigt die Effizienz der Entwickler, da sie ähnliche Algorithmen mitunter mehrfach entwickeln müssen und die Selben Tests mehrfach an verschiedenen Stellen ausgeführt werden müssen. Zudem nimmt der Transport in die verschiedenen Systeme Zeit in Anspruch. Darüber hinaus kann es lange dauern, Feedback zu Fehlern zu erhalten, wenn die zugehörigen Tests in anderen Systemen liegen.

2.2.2 Auslieferung

Es wird alle drei Monate ein Software-Update ausgeliefert. Dazwischen können auch schnellere Fehlerbehebungen ausgeliefert werden. Die Entscheidung, ob Kunden diese Updates installieren möchten oder lieber auf das nächste große Update warten möchten, liegt bei ihnen selbst.

Dies entsteht aus der Politik von SAP, die Auslieferung von Software nicht zentral zu verwalten, sondern dies den einzelnen Kunden auf ihren On-Premise-Systemen zu überlassen.

Dies führt dazu, dass Kunden auf verschiedenen Versionen operieren, was im Fall von kundenseitig entdeckten Fehlern aufwendig nachgestellt werden muss. Dies kostet viel Zeit und führt oft zu zusätzlichen Problemen und Schwierigkeiten bei der Replikation der Zustände.

2.2.3 Software Tests

Es existieren spezifische Testfälle, die von Swissdec definiert wurden und für die Zertifizierung korrekt durchgeführt werden müssen. Sobald die entsprechenden Features implementiert wurden, werden diese Testfälle in die jeweiligen Testsysteme integriert und umgesetzt. Ein Test gilt als bestanden, wenn die Lohnabrechnung der Testfälle korrekt über den vorgesehenen manuellen Prozess erstellt werden kann.

Dieses Vorgehen ergibt sich aus dem allgemeinen Charakter der Tests für die Zertifizierung des ELM-Standards. Es wird lediglich die korrekte Formatierung der Daten geprüft, nicht jedoch deren Erstellung.

Die manuelle führt zu einer fehlenden Übersicht darüber, welche Tests gerade auf welchem Stand durchgeführt werden. Das Fehlen detaillierter Tests bedeutet, dass es nicht möglich ist, im Detail zu überprüfen, ob die Software wie geplant funktioniert. Darüber hinaus werden Nebeneffekte oft erst spät bemerkt, da keine automatisierten Tests vorhanden sind. Dies ist problematisch, da die Kosten von Fehlern umso höher sind, je später sie entdeckt werden.

2.2.4 Projektmanagement

Die Zuweisung der Aufgaben erfolgt über ein dynamisches Dokument, indem ausstehende Aufgaben aufgelistet werden. Die Zuteilung der Aufgaben sowie Rückmeldungen zu Fortschritten findet in täglichen Meetings statt. Es gibt jedoch kein klar definiertes Verfahren dafür.

Dies liegt an der hohen Erfahrungsdifferenz und der damit automatisch verbundenen Hierarchie in der Abteilung.

Dies stellt jedoch ein Problem dar, da aufgrund der langen Veröffentlichungsintervalle schwierig Rückschlüsse zum aktuellen Entwicklungsfortschritt gezogen werden können. Darüber hinaus ist die Einschätzung der verbleibenden Zeit im Projekt nicht sehr zuverlässig, da mit ähnlicher Geschwindigkeit neue Aufgaben gefunden werden, wie alte Aufgaben abgearbeitet werden.

2.2.5 Dokumentation und Versionskontrolle

Wenn eine neue Aufgabe begonnen wird, wird dafür eine Feature-Note erstellt, auf der alle vorgenommenen Codeänderungen gespeichert werden. Die geplante Funktionalität wird ebenfalls auf der Note beschrieben. Zusätzlich wird am Anfang der geänderten Dateien eine Referenz auf die Feature-Note und den Entwickler hinterlegt.

Die Einführung von Feature-Notes erfolgte, um Release-Dokumentationen aus den Feature-Notes zu generieren. Dieses Konzept besteht seit den frühen Anfängen des ERP-Systems und wurde bereits verwendet, bevor moderne Projektmanagement-Tools in anderen Teilen des Unternehmens eingesetzt wurden.

Die Notizen am Anfang der Dateien haben ebenfalls ihren Ursprung in den Anfängen des Produkts. Sie wurden eingeführt, damit Autor und Zweck des Codes auch ohne moderne Versionskontrolle erkannt wurden. Diese Anmerkungen existieren seit langem und wurden teilweise sogar auf Deutsch oder Italienisch verfasst, obwohl die Implementierungssprache seit Jahrzehnten einheitlich englisch ist.

Dies führt zu Problemen, da bei häufig bearbeiteten Dateien ein unübersichtlicher Overhead am Anfang der Datei entsteht und die Verantwortlichen, die durch Mitarbeiternum-

mern repräsentiert werden, teilweise seit Jahrzehnten nicht mehr im Unternehmen tätig sind.

2.2.6 Zusammenfassung der Probleme

Der Kern der Probleme liegt in der Änderung der Zusammenstellung des Teams. Die erfahrenen Entwicklern, haben lange Zeit ohne besondere Projektmanagementtätigkeiten als Zweierteam gearbeitet und sehen keinen Anlass bei dem erweiterten Team und größeren Projekt etwas grundlegend an ihren Arbeitsabläufen zu ändern.

Zudem besteht keine klare Verantwortlichkeit, wessen Aufgabe es wäre Änderungen in die Hand zu nehmen.

2.3 Grundlagentheorie für Optimierung

Um Optimierungen für das Softwarequalitätsmanagement des Projekts vorzuschlagen, ist es wichtig, sich vorher mit den in der Vorlesung erlernten Grundlagen auseinander zu setzen.

2.3.1 PDCA

PDCA steht für den Plan-Do-Check-Act-Zyklus, der als ein strukturierter Ansatz zur kontinuierlichen Verbesserung von Prozessen, Produkten oder Dienstleistungen verwendet wird. Der PDCA Kreis besteht dabei aus vier Phasen

In Plan (P) Phase werden Ziele festgelegt und ein Aktionsplan erstellt. Es werden die zu erreichenden Ergebnisse, Ressourcenanforderungen und Maßnahmen definiert. In der Umsetzungs (Do D) Phase wird der im Planungsprozess entwickelte Aktionsplan umgesetzt. Es werden Maßnahmen ergriffen, um die gewünschten Verbesserungen in der Praxis umzusetzen. In der Überprüfungs (Check)Phase werden die durchgeführten Maßnahmen überprüft und die erzielten Ergebnisse analysiert. Es werden Daten gesammelt und Leistungskennzahlen verwendet, um den Fortschritt zu bewerten und festzustellen, ob die angestrebten Ziele erreicht wurden. Basierend auf den Ergebnissen der Überprüfung

werden in Handel (Act) Phase Anpassungen vorgenommen und Verbesserungsmaßnahmen identifiziert. Wenn die gewünschten Ziele erreicht wurden, werden die erfolgreich umgesetzten Maßnahmen standardisiert und in den laufenden Betrieb überführt. Falls notwendig, werden neue Pläne entwickelt und der PDCA-Zyklus beginnt erneut.

Der PDCA-Zyklus stellt sicher, dass kontinuierliche Verbesserungen in einem strukturierten und wiederholbaren Prozess stattfinden. Durch die wiederholte Anwendung des Zyklus können Projekte kontinuierlich lernen, ihre Prozesse optimieren und eine Verpflichtung zur Verbesserung etablieren.

2.3.2 Wasserfall Modell und Kanban Board

Das Wasserfallmodell ist ein sequenzieller und linearer Ansatz zur Softwareentwicklung. Es basiert auf der Idee, dass jeder Schritt im Entwicklungsprozess vollständig abgeschlossen sein muss, bevor der nächste Schritt beginnt. Der Name "Wasserfall" bezieht sich auf die Vorstellung, dass der Fortschritt von oben nach unten fließt, ähnlich einem Wasserfall.

Im Wasserfallmodell durchläuft ein Projekt typischerweise die folgenden Phasen in fester Reihenfolge: Anforderungsdefinition, Systementwurf, Implementierung, Überprüfung und Wartung. Jede Phase wird abgeschlossen, bevor die nächste beginnt, und es gibt wenig bis gar keine Möglichkeit für Rückwärtsbewegungen oder iterative Schleifen. Das bedeutet, dass jede Änderung oder Neuanforderung, die nach Abschluss einer Phase auftritt, erst in späteren Phasen berücksichtigt werden kann.

Das Wasserfallmodell eignet sich gut für Projekte mit klaren und stabilen Anforderungen, bei denen die Ergebnisse im Voraus genau definiert werden können. Es bietet eine klare Struktur und ermöglicht eine detaillierte Planung. Allerdings kann es bei komplexen Projekten oder solchen mit sich ändernden Anforderungen unflexibel und risikoreich sein. Fehler oder Mängel, die in einer frühen Phase übersehen werden, können sich im späteren Verlauf des Projekts schwerwiegend auswirken.

Ein Kanban Board ist ein visuelles Tool zur Verwaltung von Arbeitsabläufen und Aufgaben ist. Ein Kanban Board besteht aus Spalten, die den verschiedenen Status einer Aufgabe oder eines Arbeitsvorgangs repräsentieren, und Karten, die die einzelnen Aufgaben darstellen.

Das Kanban Board ermöglicht es einem Team, den Überblick über den Fortschritt der Arbeit zu behalten, Engpässe zu identifizieren und die Prioritäten zu setzen. Jede Aufgabe wird auf einer Karte dargestellt, die je nach Fortschritt durch die Spalten des Boards verschoben wird, von der Aufgabenplanung über die Bearbeitung bis zur Fertigstellung. Teammitglieder können den Status einer Aufgabe schnell erkennen und wissen, welche Aufgaben als nächstes erledigt werden müssen.

Das Kanban Board fördert die Transparenz und Zusammenarbeit im Team. Es ermöglicht eine kontinuierliche Verbesserung, da Engpässe und Flaschenhälse sichtbar werden und Maßnahmen ergriffen werden können, um diese zu beseitigen. Das Board kann auch dazu dienen, den Arbeitsfluss zu optimieren und die Effizienz zu steigern.

Im Vergleich zum Wasserfallmodell bietet das Kanban Board eine flexible und iterative Herangehensweise an die Arbeit. Es passt sich leicht an sich ändernde Anforderungen oder Prioritäten an und ermöglicht eine kontinuierliche Anpassung und Verbesserung des Arbeitsprozesses. Es wird oft in agilen Entwicklungsumgebungen eingesetzt, in denen eine hohe Flexibilität und schnelle Reaktion auf Veränderungen erforderlich sind.

2.4 Optimierungen und Marktspezifische Bewertung

2.4.1 Awareness für Sinnhaftigkeit von Planung schaffen

Ein wichtiger grundlegender Schritt für den Erfolg aller weiteren Maßnahmen ist die Schaffung von Verständnis, warum ein gutes Projektmanagement, regelmäßige Rückmeldung und gute Abstimmung relevant für die Qualität der Software ist. Insbesondere muss dabei die Wichtigkeit des PDCA-Protokolls (Plan-Do-Check-Act) vermittelt werden, das grundlegend für eine Verbesserung der Prozesse ist.

2.4.2 Einführung moderner Versionierungs- und Projektmanagement Tools

Um PDCA besser umsetzen zu können wäre es empfehlenswert ein Vorgehen für den Projektverlauf zu definieren. Aufgrund der Projektbeschaffenheit würde es sich dafür

anbieten, sich am Wasserfallmodell zu orientieren. Bei der Softwareentwicklung ist es zwar üblich agile Softwareentwicklung anzuwenden. Aufgrund der konkreten Anforderungen die erreicht werden müssen sowie der dafür notwendigen Dokumentation, die ohnehin erstellt werden muss, bietet sich für die Umsetzung der Anforderungen für ein Zertifikat jedoch allgemein die Struktur des Wasserfallmodells mehr.

Ein Vorteil bei dieser Umsetzung sind die bereits von Swissdec gegebenen klaren Anforderungen, die starr sind und sich nicht grundlegend ändern werden. Die entstandene Dokumentation muss für die Abnahme des Zertifikats zu Projektende ohnehin erstellt werden und der Zeitrahmen, bis zu dem das Projekt abgeschlossen werden soll kann gut geplant werden. Dadurch können Abweichungen der vorhandenen Ressourcen bzw. vom Zeitplan entsprechend PDCA frühzeitig erkannt und gegengesteuert werden. Ein weiterer Vorteil ist dass die Phasen der Anforderungen und Überprüfung durch die Rahmenbedingungen von Swissdec bereits gegeben sind. Da das Projekt auf eine existierende Codebasis aufbaut ist es sinnvoll, in der Entwurfsphase sich mit der bestehenden Architektur und nötigen Anpassungen dafür auseinanderzusetzen

Eine weitere Empfehlung ist der Umstieg auf neue Technologien, nicht zuletzt um die Planung im Wasserfallmodell sinnvoll umsetzen zu können und die Dokumentation zu unterstützen.

2.4.3 Testgetriebene Entwicklung mit automatisierten Tests

Ein weiteres Gebiet für Optimierungen ist die Überarbeitung des Testbetriebs. Ein wichtiger Schritt zur Erhöhung der Codequalität wäre die Einführung von Unit Tests, die auf kleinster Ebene lokal und schnell die Funktionalität des Codes garantieren. Zusätzlich wäre eine Adaption der Testgetriebenen Entwicklung für die bereits von Swissdec gegebenen Acceptance Tests sinnvoll. Dieses Vorgehen würde die guten Qualitätsmetriken ausnutzen. Da diese Tests fest gegeben und notwendig für eine erfolgreiche Zertifizierung sind, würden dieses Vorgehen nicht für einen höheren Aufwand oder mehr Testwartung führen. Da das Projekt im Kontext eines umfassenden Lohndatensystems stattfindet ist die Umsetzung regelmäßiger Smoke- und Tests sehr sinnvoll, um zu prüfen ob andere Systemkomponente von der Implementierung beeinflusst werden.

Eine weitere Sinnvolle Maßnahme zur Sicherung der Qualität ist die automatisierte der Tests neuen Änderungen. Dies garantiert eine regelmäßige Ausführung aller Tests und damit eine Frühzeitige Erkennung von Problemen.

Dies wäre Sinnvoll umsetzbar im Rahmen einer Continuous Integration. Continuous Delivery in die anderen Entwicklungs- und Testsysteme wäre auch eine Sinnvolle Maßnahme, da dies Zeit spart, die Entwickler für den halbautomatisierten Transport der Änderungen benötigen würden. Es wäre Sinnvoll die Pipeline um automatische Tests in den anderen Systemen zu erweitern, damit Entwickler direkt informiert werden wenn Codeänderungen für die anderen Systeme nötig sind. Continuous Deployment des Produktivcodes an produktive Systeme von Kunden ist in diesem Zusammenhang nicht Sinnvoll, da hier die Verantwortun auf Seiten der Kunden liegt und eine zu filigrane Aufteilung der Versionierung den laufenden Betrieb stören könnte.

Da die Entwicklung des neuen Lohnmeldestandards auf den Selben Systemen wie die aktuelle Lohnmeldesoftware stattfindet, muss überlegt werden, wie die Auslieferung stattfindet. Der Angesetzte Zeitrahmen für den neuen Lohnmeldestandard ist etwa 3 Jahre. Dies würde eine Verteilung der Auslieferung über 12 Releases bedeuten. Da ein unfertiger Lohnmeldestandard Kunden nichts nützt und lediglich zu Verwirrung bei den Patchnodes führen würde, wird ein gebündelter Launch vorgeschlagen. Dies führt zwar beim eventuellen Launch zu sehr großen Änderungen. Da diese jedoch ständig intern auf den neusten Stand integriert werden, kann eine Integration Hell vermieden werden. Da die bestehenden Lohnmeldestandards im System weiterhin bestehen bleiben, handelt es sich um eine Parallele Launchstrategie, sofern man die Lohnmeldungen mithilfe von verschiedenen Standards als getrennte austauschbare Programmteile betrachten kann.

3 Zusammenfassung

Zusammenfassend lässt sich sagen, dass die Probleme bei der Implementierung des neuen Lohnmeldestandards hauptsächlich auf der Langen Historie und den damit verbundenen altbackenen Prozessen bei der Entwicklung beruhen. Einige der herausgearbeiteten Probleme lassen sich auch nicht lösen, da sie auf vertraglich garantierten Zusagen an Kunden beruhen. Durch die Einführung von festen Phasen nach dem Wasserfallmodell und einer verbesserten Teststrategie in Kombination mit der Einführung neuer, automatisierbarer Tools lässt sich jedoch eine Besserung der Effizienz und Bewertbarkeit des Fortschritts schaffen. Mit den vorgestellten Maßnahmen haben die Entwickler und Projektleitung die Möglichkeit, eine Verbesserung des Prozesses eigenständig weiter zu verbessern.

Literaturverzeichnis

- [1] Swissdec. *Über Swissdec*. 06.05.2023. URL: <https://www.swissdec.ch/de/ueber-swissdec/>.