

Qualitätsmanagement in der IuK der Bayerischen Vermessungsverwaltung

Christian Albrecht



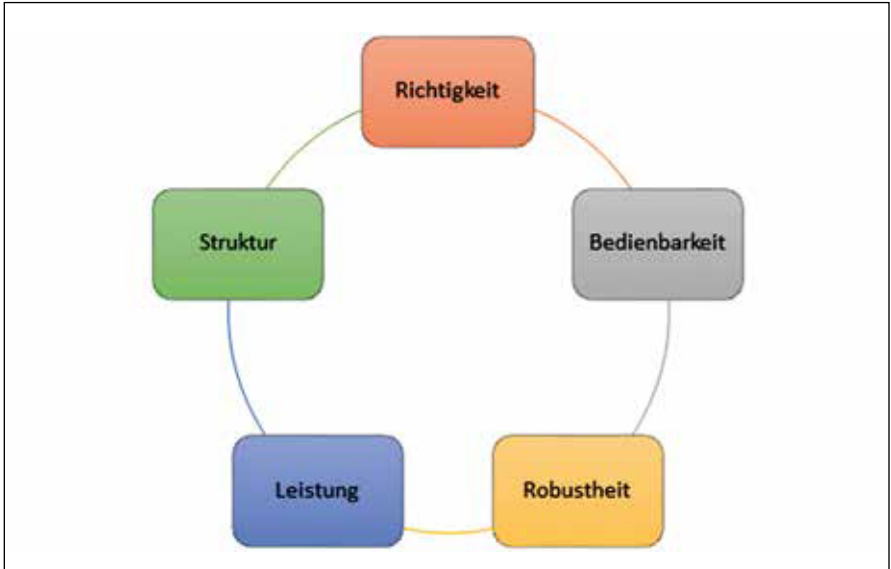
1. Software-Entwicklung in der Bayerischen Vermes- sungsverwaltung

Die IuK-Abteilung des Landesamts für Digitalisierung, Breitband und Vermessung (LDBV) entwickelt, betreut und pflegt unterschiedliche IuK-Anwendungen und Systeme für die Bayerische Vermessungsverwaltung (BVV). Diese helfen die eigenen Arbeitsprozesse effizient und umfänglich zu erledigen und die dabei erzeugten Produkte online und offline für Kunden zur Verfügung zu stellen.

2. Qualitätsmanagement in der IuK

Neben der reinen Entwicklung und Systembereitstellung ist das Qualitätsmanagement ein wesentlicher Bestandteil. Ziel des Qualitätsmanagements in der Informations- und Kommunikationstechnologie (QM-IuK) ist die permanente Qualitätssicherung und Qualitätsverbesserung von Anwendungen, Produkten und Arbeitsabläufen um somit einen reibungslosen und effizienten Betrieb zu gewährleisten. Entscheidend dafür ist die möglichst frühzeitige Information und Einbindung bei Planungen, Optimierungsvorschlägen und Projekten.

Es werden dabei verschiedene Qualitätsziele verfolgt:



- **Richtigkeit**
Die Anwendungen und Produkte müssen fachlich richtig und technisch fehlerfrei sein.
- **Gute Bedienbarkeit**
Eine einfache und intuitive Bedienbarkeit der Anwendungen und Produkte muss gewährleistet sein.
- **Robustheit**
Die Anwendungen und Produkte dürfen keine Fehler aufweisen. Ebenso müssen diese Fehlbedienungen der Nutzer abfangen.
- **Leistung**
Die Performanz muss ausreichend sein, damit alle Aufgabenstellungen effizient erledigt werden können. Dabei spielen die Nutzerzahlen, die Hardware und die Sauberkeit der Programmierung eine entscheidende Rolle.
- **Gute Struktur**
Fehlerbehebungen, Programmänderungen und Programmweiterungen sollen bei der Software gut durchführbar sein.

3. Von der Idee bis zum Rollout der Software

Ausgelöst durch Gesetzesänderungen, Vorgaben des Bayerischen Staatsministeriums der Finanzen und für Heimat aber auch durch Anforderungen der eigenen Mitarbeiter gibt es vielfältige Anforderungen an die Entwicklungsabteilung des LDBV. Auch sind bundesweite Vorgaben der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV) umzusetzen. Hinzu kommt oft, dass die jeweilige Realisierung am besten „schon gestern“ hätte erfolgen sollen.

Übersehen wird dabei gerne, dass die IT-Landschaft zunehmend komplexer geworden ist. Bei Neuentwicklungen, aber auch bei Anpassungen, sind im Normalfall viele Entwicklungsstellen betroffen. Es gibt nicht mehr „den einen“ Entwickler, der ein komplettes Programm samt Systemeinbindung erstellt. Hier übernimmt QM-luK eine koordinierende und moderierende Funktion. Dadurch wird gewährleistet, dass alle betroffenen Stellen involviert und Abhängigkeiten frühzeitig erkannt werden. Erste Testvorbereitungen und Testdurchführungen können frühzeitig in die Wege geleitet werden und mögliche Herausforderungen schnell identifiziert werden.

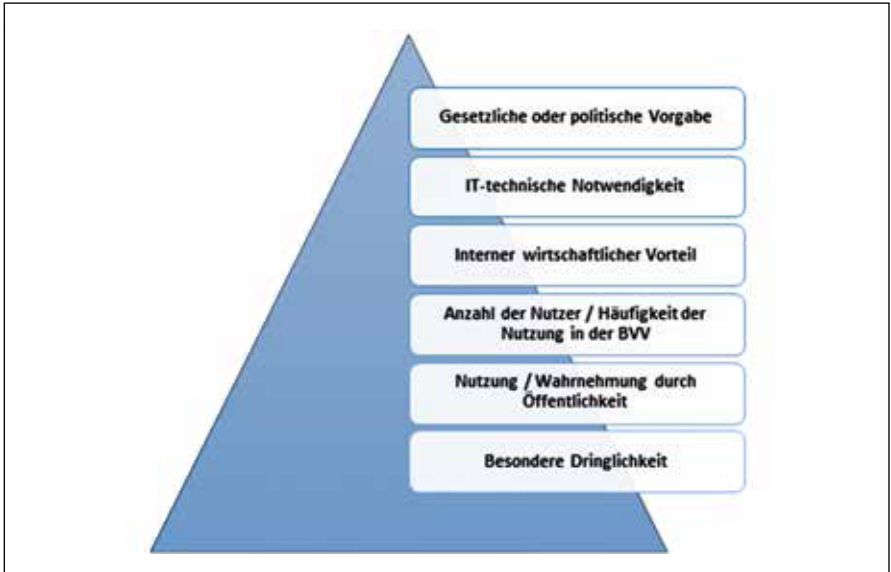
3.1. luK-Leitfaden

Der luK-Leitfaden der BVV gibt hierzu den erforderlichen Rahmen vor. Er beschreibt den Lebensweg eines luK-Verfahrens von der Idee über die Konzeption, Entwicklung, Freigabe und Einführung bis hin zur Pflege. Ziel ist es, einheitliche Vorgehensweisen festzulegen und die Kommunikation der am Entwicklungsprozess Beteiligten zu regeln. Je nach Umfang des luK-Vorhabens wird nach den Typen „luK-Projekt“, „luK-Leistung“ und „luK-Pflegemaßnahme“ unterschieden.

3.1.1. Prioritätenpyramide

Die Forderungen nach Weiter- bzw. Neuentwicklung der luK-Verfahren reichen von reiner Komfortsteigerung bis hin zur Umsetzung strategischer Zielvorgaben des Bayerischen Staatsministeriums der Finanzen und für Heimat (StMFH). Die gewünschten neuen oder geänderten luK-Verfahren übersteigen die vorhandenen Mitarbeiterkapazitäten in der luK-Abteilung. Die Entscheidung, welches luK-Verfahren zuerst umgesetzt wird, soll dabei nicht dem zuständigen Entwickler überlassen werden.

Um auf Grundlage objektiver Kriterien eine Aussage über die Wichtigkeit von luK-Vorhaben treffen zu können, wurde im Jahr 2013 eine so genannte Prioritätenpyramide eingeführt. Gemeinsam mit dem Auftraggeber werden dabei Punkte zu den folgenden Bewertungskriterien vergeben und hieraus eine Gesamtpunktzahl berechnet. Die Gesamtpunktzahl ist ausschlaggebend für den Zeitpunkt der Umsetzung des luK-Vorhabens. Bei zu geringer Punktzahl wird von einer Umsetzung abgesehen und der luK-Auftrag storniert.



- *Gesetzliche oder politische Vorgabe*
Punktwert ist abhängig davon, ob es für die LuK-Maßnahme einen konkreten gesetzlichen Auftrag, einen Auftrag des StMFH oder eine politische Vorgabe gibt.
- *IT-technische Notwendigkeit*
Punktwert ist abhängig davon, ob die LuK-Maßnahme für die Fortsetzung des Betriebs unabdingbar ist, der Umsetzung der LuK-Strategie dient oder keine IT-technische Notwendigkeit gegeben ist.
- *Interner wirtschaftlicher Vorteil*
Punktwert ist abhängig vom wirtschaftlichen Nutzen (Beschleunigung / Verbesserung der internen Prozesse).
- *Anzahl der Nutzer / Häufigkeit der Nutzung in der BVV*
Punktwert ist abhängig von der Anzahl der Nutzer und der Häufigkeit der Nutzung.
- *Nutzung / Wahrnehmung durch Öffentlichkeit*
Punktwert ist abhängig von der Häufigkeit der Nutzung durch die Öffentlichkeit oder der Bedeutung als Imageträger für die Außenwirkung der Bayerischen Vermessungsverwaltung.
- *Besondere Dringlichkeit*
Punktwert ist abhängig von der Dringlichkeit der Umsetzung. Hierbei spielt auch eine Rolle, ob Fachfragen schon geklärt sind und damit die Umsetzung schneller erfolgen kann.

3.1.2. luK-Projekt

Komplexe luK-Vorhaben werden als luK-Projekt behandelt und sind durch einen definierten Anfangs- und Endtermin begrenzt. Diese werden vom Bayerischen Staatsministerium der Finanzen und für Heimat oder / und der Amtsleitung des LDBV initiiert. Im Projektauftrag werden unter anderem der Lenkungsausschuss, die Projektleitung und die Projektgruppenmitglieder festgelegt. Wichtig ist, dass die Projektbeteiligten zu einem fest definierten Anteil von ihrer Arbeit freigestellt werden.

Typische Beispiele für luK-Projekte waren die Einführung von ALKIS sowie die erst kürzlich erfolgte Einführung von ETRS89 / UTM in Bayern.

3.1.3. luK-Leistung

Sollen neue Verfahren entwickelt werden, die nicht den Umfang eines luK-Projekts aufweisen, oder größere Änderungen bei bestehenden Verfahren vorgenommen werden, ist durch den Anwender eine luK-Leistung zu beauftragen. Hierzu zählen auch die Installation und Verbreitung von lizenzpflichtiger und lizenzfreier Software.

luK-Aufträge können – die Zustimmung des Referatsleiters vorausgesetzt – über ein automatisiertes Formular im Intranet gestellt werden. Damit ist gewährleistet, dass es sich nicht um den „Einzelwunsch“ eines Mitarbeiters handelt. Im Auftrag werden die wesentlichen Punkte dargelegt, wie zum Beispiel die Ausgangslage, Rahmenbedingungen, fachliche Zielsetzung, Wirtschaftlichkeit und Terminvorstellung. Bevor die weitere Bearbeitung übernommen wird, ist eine Zustimmung des jeweiligen Abteilungsleiters erforderlich.

Wichtig ist es, die Anforderungen zwischen dem zuständigen luK-Referat, dem fachlichen Ansprechpartner und dem Auftraggeber – sofern diese nicht identisch sind – genau abzustimmen. Daraus ergeben sich die Priorität und der erforderliche Entwicklungsaufwand. Nachdem alle Fakten zusammengestellt worden sind, erfolgt eine Abstimmung in der Leitungsrunde der Entwicklungsabteilung. Hier wird festgelegt, ob und wann mit der Umsetzung begonnen wird und wer dies übernimmt.

3.1.4. luK-Pflegemaßnahme

Als luK-Pflegemaßnahme wird die Behandlung von Fehlern oder kleineren Optimierungsvorschlägen zu bereits eingeführten luK-Verfahren verstanden. Diese werden vom zuständigen Entwickler eigenverantwortlich abgearbeitet. Bei Bedarf, insbesondere bei Optimierungsvorschlägen, wird der fachliche Ansprechpartner hinzugezogen, um eine Entscheidung über die Notwendigkeit der Umsetzung zu treffen. Auch wird vermieden, dass es zu unterschiedlichen Entscheidungen zu demselben Sachverhalt, der aber von verschiedenen Mitarbeitern herangetragen wurde, kommt.

3.2. Test des luK-Verfahrens

Bereits während des Entwicklungsprozesses beginnt der Test der Software. Hierzu werden so genannte Testfälle implementiert, die laufend den Entwicklungsstand auf Fehler prüfen. Dabei wird nicht nur die technische Funktionsfähigkeit, sondern auch die fachliche Richtigkeit geprüft. Die Entwickler sehen laufend, ob ihr Programmcode die fachlichen Festlegungen einhält oder ob Programmierfehler vorhanden sind.

Mit dem üblicherweise im Anschluss folgenden Test- und Freigabeverfahren wird die fertiggestellte Software – ebenfalls unter Verwendung von Testfällen – ausführlich getestet und – sofern die Qualitätsansprüche erfüllt werden – im Anschluss technisch freigegeben. Bei luK-Vorhaben mit Außenwirkung, zum Beispiel eine Weiterentwicklung des BayernAtlas, folgt auf die technische Freigabe in einem zweiten Schritt die vertriebliche Freigabe. Erst mit dieser erfolgt – mit einem gewissen Zeitversatz – die Produktivsetzung.

3.2.1. Testaspekte und Testfälle

Beim explorativen Testen überlässt man den Testern die Entscheidung, was sie wie testen. Die Qualität des Testergebnisses ist dabei stark von der Erfahrung des Testers abhängig und in einem gewissen Maße auch Glückssache, ob an der „richtigen“ Stelle nach Fehlern gesucht wird.

Mit Hilfe von Testaspekten und Testfällen wird den Testern konkret vorgegeben, was sie wie testen sollen. Testaspekte beschreiben dabei, was getestet werden soll. Testfälle dagegen konkretisieren die Testaspekte und geben an, wie getestet werden soll.

Um dies etwas anschaulicher darzustellen, soll dies anhand von Beispielen konkretisiert werden. Ein mögliches Szenarium wäre das Testen der Berechnung der Länge einer Strecke. Der Testaspekt würde wie folgt lauten:

Das Modul zur Berechnung der Strecke zwischen zwei Punkten soll getestet werden.

Ein möglicher zugehöriger Testfall wäre:

Berechne die Strecke zwischen den beiden Punkten A1 und A2. Der Test ist erfolgreich, wenn die Strecke 8,54 m beträgt.

Daran wird schnell ersichtlich, dass es zu einem Testaspekt viele verschiedene Testfälle geben kann. Bei der Erstellung von Testfällen müssen dabei auch die „Grenzwerte“ betrachtet werden: Was passiert, wenn die beiden Punkte identisch sind und die Strecke somit Null ergibt? Gibt es dann einen Programmfehler?

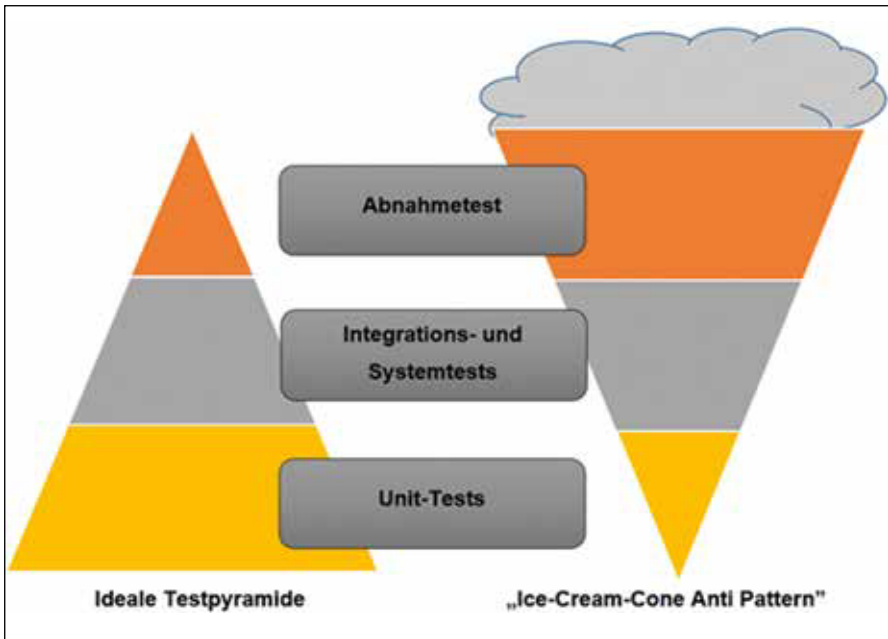
Neben grundlegender geodätischer Erfahrung ist hier vor allem Kreativität zur Erstellung der Testaspekte und -fälle gefragt. Idealerweise können die Testaspekte vom Fachkonzept abgeleitet werden, in welchem die Funktionalitäten des zu entwickelnden Programms beschrieben sind.

3.2.2. Teststufen

In der Praxis ist es nicht möglich, alle möglichen Programmfunktionalitäten und Programmeingabemöglichkeiten abschließend zu testen. Um möglichst effizient zu testen, müssen die ersten Tests schon frühzeitig starten. Denn je später ein Fehler gefunden wird, desto aufwendiger ist dessen Behebung.

Erste so genannte Unit-Tests finden bereits durch die Entwickler im Rahmen der Programmierung statt. Im Programmcode werden hierzu bereits Testfälle implementiert, die bei jeder automatisierten Paketerstellung eine Aussage treffen, ob die Testfälle alle korrekt abgearbeitet werden. Abhängig von der Größe des IuK-Vorhabens können noch Integrations- und Systemtests und letztendlich der Abnahmetest im Rahmen des Test- und Freigabeverfahrens folgen.

In der Testpyramide nach Mike Cohn wird der richtige Mix zwischen den einzelnen Teststufen anschaulich dargestellt.



Viele schnell durchzuführende und zu pflegende Unit-Tests legen das Fundament für eine solide Teststrategie. Die folgenden Integrations- und Systemtests haben meist längere Ausführungszeiten und sind aufwendiger zu warten. Die zum Abschluss folgenden Abnahmetests benötigen zum Schluss vergleichbar lange und geben einen

Einblick, ob das System im Ganzen funktioniert. In dieser abschließenden Testphase ist es nicht mehr möglich, alle möglichen Programmzweige zu testen, weshalb so viele Tests wie möglich in den beiden unteren Stufen erfolgen sollen.

Finden hingegen am Anfang nur wenige Unit-Tests statt, müssen die Testaktivitäten in den beiden folgenden Teststufen deutlich gesteigert werden. Diese umgekehrte Form der Pyramide erinnert eher an die Form einer Eiswaffel, weshalb dies auch als „Ice-Cream-Cone Anti Pattern“ bezeichnet wird. Diese Teststrategie ist ineffizient und nicht empfehlenswert.

3.2.3. Test- und Freigabeverfahren

Nach Fertigstellung der Entwicklung plant und organisiert QM-luK den Abnahmetest im Rahmen eines Test- und Freigabeverfahrens. Hierfür werden zusammen mit dem Auftraggeber und den Entwicklern Testvorgaben erarbeitet. Der Abnahmetest wird in der Regel von den zukünftigen Anwendern durchgeführt. Dies können die Mitarbeiter des jeweiligen Fachreferats am LDBV sein oder Mitarbeiter an den Ämtern für Digitalisierung, Breitband und Vermessung (ÄDBV). Philosophie ist hierbei, dass immer der Personenkreis testet, der die Anwendung später verwendet. Der Test erfolgt somit von den Personen, die die Software täglich nutzen und das entsprechende fachliche Hintergrundwissen sowie die Erfahrung aus der Praxis haben.

3.2.4. Freigabe der Software

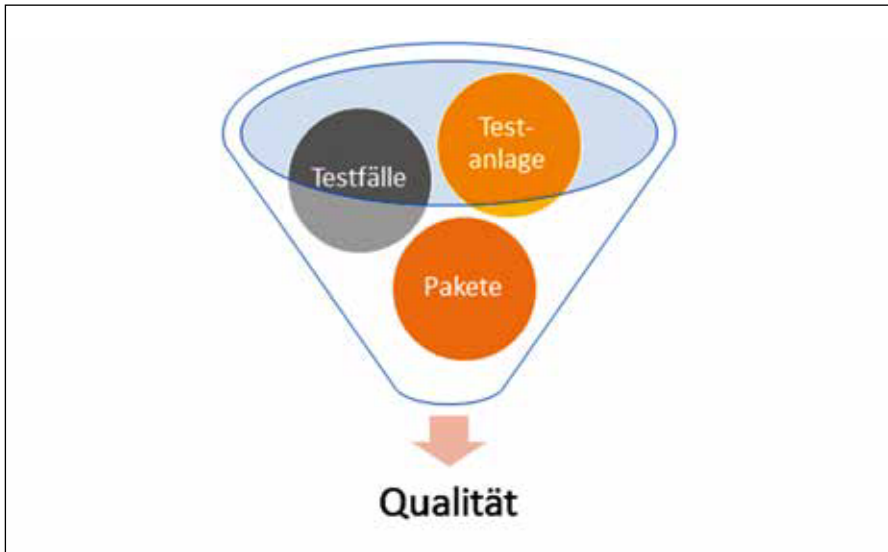
Nach Abschluss des Testverfahrens wird von den Testpersonen jeweils ein Testbericht erstellt, in dem auch auf Fehler und Optimierungsvorschläge eingegangen und letztendlich eine Aussage zur Freigabefähigkeit getroffen wird. Anhand dieser Testberichte trifft QM-luK die Entscheidung, ob die Software technisch freigegeben werden kann. Damit steht die Software zum produktiven Einsatz bereit. Bei Software mit Außenwirkung erfolgt die eigentliche Freischaltung erst mit der vertrieblichen Freigabe durch die Vertriebsabteilung.

3.3. Automatisierung der Tests

Die weitere Steigerung der Testabdeckung und die damit verbundene weitere Verbesserung der Software-Qualität kann durch eine Automatisierung der Tests erreicht werden. Insbesondere im Bereich der Unit-Tests können während der Entwicklung frühzeitig Fehler erkannt und behoben werden. Der große Vorteil der Testautomatisierung ist das permanente, meist tägliche, Prüfen der Software. Diese Testmethodik wird in der BVV derzeit schrittweise eingeführt.

Aus dem Programmcode wird üblicherweise jede Nacht automatisiert ein Programmpaket erstellt und automatisch in eine Testanlage eingespielt. Mit Hilfe der Testfälle wird die Software auf der Testanlage automatisiert dem Test unterzogen. Die beim

manuellen Testen erforderlichen Schritte zur Erstellung des Programmpakets, der Installation und dem anschließenden Test laufen hierbei automatisiert ab. Dies spart Ressourcen, die an anderer Stelle besser eingesetzt werden können. Die Entwickler und QM-luK können durch die automatisierten Tests täglich eine Aussage über die Software-Qualität treffen.



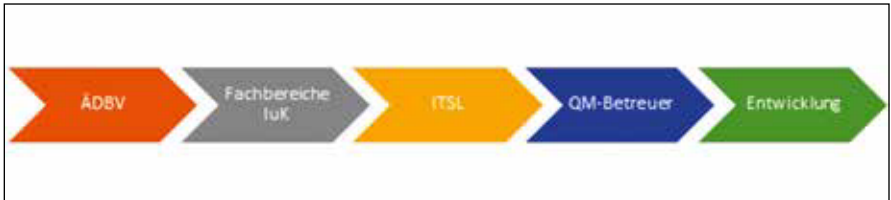
4. Kommunikationsstrukturen – das IT-Servicesystem (ITSS) und die IT-Serviceleitstelle (ITSL)

Mit dem IT-Servicesystem (ITSS), einem webbasierten Helpdesk-System, können die Anwender Fehlermeldungen, aber auch Optimierungsvorschläge, zu den luK-Verfahren melden. Hierzu erfassen die Anwender so genannte „Tickets“.

Sowohl die Anwender als auch die Entwickler haben gewisse Anforderungen an die Tickets. Die Anwender möchten nur eine Stelle haben, an die sie ihre IT-Anfrage weitermelden können („Single Point of Contact“). Neben einer schnellen Bearbeitung wünschen sie eine aussagekräftige Rückmeldung zu ihren Anfragen, beziehungsweise das Beheben des Problems. Von Entwicklerseite hingegen wird erwartet, dass nur qualifizierte, ausreichend beschriebene und nachgeprüfte Meldungen weitergestellt werden. Ein zusätzlich angehängter Screenshot sagt hier oft mehr als tausend Worte.

Auch sollen die Entwickler nur die Tickets erreichen, für die sie zuständig sind – auf Grund der komplexen Systemlandschaft gestaltet sich dies nicht immer leicht.

Um diese Anforderungen möglichst gut umsetzen zu können, gelangen Tickets über einen fest definierten Prozess zur Entwicklungsabteilung und wieder zurück zum Anwender:



Tickets, die an den ADBV erfasst werden, gelangen über den jeweiligen Systemverwalter zu den Fachbereichen IuK, die es einmal pro Regierungsbezirk gibt. Sofern dort das Problem nicht gelöst werden kann, wird das Ticket an die IT-Serviceleitstelle (ITSL) weitergestellt. Die ITSL stellt eine zentrale Schnittstelle für Anfragen an die IT-Entwicklung der BVV dar, die auch alle Störungen und Wartungen im IT-Bereich dokumentiert und publiziert. Durch feste Arbeitsschichten sind eine schnelle Reaktionsfähigkeit und eine fest definierte Erreichbarkeit gewährleistet. Von der ITSL werden die Tickets anschließend über den zuständigen QM-Betreuer an die zuständigen Entwickler weitergegeben. Sobald das Ticket bearbeitet wurde, wird dieses auf umgekehrten Weg wieder an den Kunden am ADBV zurückgestellt. Am LDBV erfolgt die Ticketbearbeitung in ähnlicher Weise, die Rolle des Fachbereichs IuK übernimmt hier jedoch der zuständige IT-Partner.

Durch diese definierten Prozesse wird gewährleistet, dass Entwickler nicht auf Zuruf einzelner Anwender nach dem „Hey Joe“-Prinzip agieren. Anstatt ein Ticket zu erfassen, gehen die Anwender direkt auf die ihnen bekannten Entwickler zu und umgehen somit den vorgesehenen Workflow. Der Anwender mag sich eine schnelle Lösung seines Problems erhoffen. Allerdings werden die Entwickler aus ihrem normalen Entwicklungsprozess gerissen und gefährden dabei die Terminplanung ihrer eigentlichen Aufgabe. Zudem sind „Hey Joe“-Anforderungen nicht dokumentiert und meist weder fachlich noch technisch abgestimmt. Dies kann zu unvorhergesehenen Seiteneffekten führen.

5. Fazit

Das Qualitätsmanagement in der IuK ist eine wichtige Säule in der Software-Entwicklung der BVV. Teststrategie und -umfang müssen an die Kritikalität und die Qualitätsziele angepasst sein. Das strukturierte Vorgehen bei neuen IuK-Vorhaben gemäß dem IuK-Leitfaden hat sich bewährt. ■■■