DER ARBEITSZEITRECHNER: EINE ERFOLGREICHE VERBINDUNG VON RECHTSINTERPRETATION UND IT

Johannes Gärtner / Georg Gasteiger / Christian Dunst

XIMES GmbH calcuLex GmbH
Hollandstr 12/12, 1020 Wien AT
gaertner@ximes.com; http://www.ximes.at

AK-Wien
Prinz Eugenstr. 20-22, 1040 Wien AT
Georg.Gasteiger@akwien.at; Christian.Dunst@akwien.at; http://www.akwien.at

Schlagworte: ***Legal Tech, Software-Entwicklung, Explainability, automatisierte juristische Begründungserzeugung, Arbeitszeitrechner, Smart Contracts, Smart Laws***

Abstract: *Relevante Teile des Rechts, wie etwa das Arbeitszeitgesetz (AZG) beziehen sich auf Berechnungen. Die Repräsentation dieser Gesetze in der Literatur ist ungeeignet für eine direkte Abbildung in Software. Umgekehrt sind auch Programme schwer zugänglich und für JuristInnen meist unlesbar und deren Berechnungen nur schwer nachvollziehbar. In einem F&E Projekt gelang es uns die Darstellungen des Rechts mit der Informatik zu verbinden: Kompakte Falldarstellungen, automatisierte Tests und die Devise, nicht nur richtig, sondern nachvollziehbar richtig zu rechnen, waren die Schlüssel.*

1. Einleitung

Das Arbeitszeitrecht regelt einen äußerst sensiblen Bereich, nämlich das Austauschverhältnis von Lebenszeit gegen Entgelt. Umso unerfreulicher ist es, dass diese Bestimmungen höchst komplex und mitunter unübersichtlich sind[1]. Untersucht man diese Bestimmungen jedoch näher stellt man fest, dass viele Gesetze, insbesondere im Bereich des Arbeitsrechts – aber auch Vorschriften der Sozialversicherung und des Steuerwesens – Rechenvorschriften darstellen. Im AZG[2] und im ARG[3] werden eine Vielzahl derartiger Rechenregeln definiert. Dies zeigt sich besonders deutlich an folgenden Beispielen:

- Prüfung der Zulässigkeit von Arbeitszeiten bezüglich Höchstarbeitsgrenzen – z.B. §§ 7, 9 AZG.
- Lage und Ausmaß täglicher Ruhezeiten samt Unterbrechungsregeln – z.B. §§ 12, 20a AZG.
- Regeln was unter Teilzeit zu verstehen ist und wann Zuschläge bei Mehrarbeit anfallen – z.B. §§ 19d, 19e AZG.
- Erfordernisse der Wochenendruhe (Lage, Dauer) – z.B. §§ 3 ff ARG.

In Kollektivverträgen kommen typischerweise weitere Regeln hinzu:

- Nutzung der gesetzlichen Gestaltungsspielräume, wie etwa die Ermöglichung flexibler Arbeitszeitverteilung in Form von Durchrechnungs-/Bandbreitenmodellen (vgl. § 7 des Rahmenkollektivvertrages für

[1] Vgl. dazu ausführlich etwa Heilegger/Klein, Arbeitszeitgesetz, 4. Aufl., ÖGB Verlag, Wien 2016, S. 32.

[2] Bundesgesetz vom 11. Dezember 1969 über die Regelung der Arbeitszeit (Arbeitszeitgesetz – AZG) StF: BGBl. Nr. 461/1969 i.d.F. BGBl. I Nr. 53/2018.

[3] Bundesgesetz vom 3. Feber 1983 über die wöchentliche Ruhezeit und die Arbeitsruhe an Feiertagen (Arbeitsruhegesetz – ARG) StF: BGBl. Nr. 144/1983 i.d.F. BGBl. I Nr. 53/2018.

Arbeiterinnen/Arbeiter in der Denkmal-, Fassaden- und Gebäudereinigung, im sonstigen Reinigungsgewerbe und in Hausbetreuungstätigkeiten KV Reinigung[4]).
- Erhöhte Zuschläge für bestimmte Arbeitszeiten, wie etwa für Nacht- oder Wochenendarbeit (vgl. § 10 Abs. 5 KV Reinigung).
- Sonstige Sonderregelungen wie etwa eine Mindestanzahl zu verrechnender Stunden bei kurzen Einsätzen (vgl. § 10 Abs. 12 KV Reinigung).

In Unternehmen und auch in der Arbeitsrechtsberatung der AK Wien geht es darum, Arbeitszeiten von ArbeitnehmerInnen im Hinblick auf Mehr- und Überstunden, sowie sonstige Zuschläge auszuwerten, um die Zulässigkeit der Zeiten beurteilen zu können und insbesondere die Höhe des Entgelts zu berechnen.

Bei manueller Auswertung ist dies mit erheblichem Zeitaufwand verbunden (fallweise mehrere Stunden Arbeit pro abzurechnender Person). Die Berechnungen sind teils kurz und einfach (z.B. Anzahl der Stunden eines Einsatzes), meistens erfordern sie jedoch mehrere Schritte (z.B. rollierende Ausgleichszeiträume für Mehrstunden laut § 10 Abs. 11 KV Reinigung). Entsprechend gibt es einen Bedarf an Automatisierung.

In einem sehr einfachen Verständnis der Aufgabenstellung könnte eine direkte Abbildung der Normen in Rechenregeln und ihre Integration in ein Softwaresystem angestrebt werden. So einfach ist es aber nicht: Erstens müssen Rechtsnormen erst interpretiert werden und diese Interpretationen weisen oft eine Bandbreite auf[5]. Der Spruch «Zwei JuristInnen ergeben drei Meinungen» scheint nach wie vor berechtigt. Zusätzlich variiert auch die Darstellung von Sachverhalten, wobei teils unterschiedliche Aspekte berücksichtigt werden oder im Laufe der Zeit weitere dazu kommen. Dazu kommt eine Vielzahl an möglichen Sachverhaltskonstellationen, die alle bei der Erstellung des Programmes mitbedacht und gelöst werden müssen. Zusätzlich bedarf es für sämtliche Varianten sinnvoller Eingabewege, ohne jedoch den Enduser durch eine unüberschaubare Anzahl an Eingabemöglichkeiten zu überfordern. Für die Programmierung in ein Softwaresystem ist dies ein sehr schwieriger Zustand. Ganz generell ist hier das Zusammenspiel von Recht und Software ein wichtiges Thema[6].

In einem F&E-Projekt der AK Wien haben wir Methoden und Vorgehensweisen entwickelt, die einen guten Umgang mit der Vielfalt der Rechtsinterpretationen ermöglichen, sowie die Darstellung der Sachverhalte und eine Abbildung von Rechtsinterpretationen in Software unterstützen. Dabei haben wir drei Themenfelder im Detail bearbeitet:

- Wie können unterschiedliche Rechtsinterpretationen bei gleichem Sachverhalt gut miteinander verglichen werden? Die Erfahrungen und der gewählte Ansatz werden im Abschnitt «Rechtsinterpretationen explizieren & stabilisieren» diskutiert.
- Auch wenn sich Juristen perfekt verstünden ist noch nicht sichergestellt, dass InformatikerInnen sie verstehen (und umgekehrt). Daraus ergibt sich das zweite zentrale Thema: Wie kann möglichst gut sichergestellt werden, dass die Abbildung im Softwaresystem die schlussendlich gewählte Rechtsinterpretation widerspiegelt? Der Ansatz wird im Abschnitt «Verknüpfung Rechtsinterpretation und Berechnung» dargestellt.
- Sehr häufig erfolgen Berechnungen über viele Zwischenschritte. Entsprechend ist ein konkretes Ergebnis für den Rechtsanwender in der Praxis häufig nicht sofort oder nur mit hohem Aufwand nachvollziehbar («Wie kommt das Programm auf diesen Wert?»). In Diskussionen (sowohl bei der Entwicklung als auch bei der späteren Nutzung) führt das dann sehr leicht zu Unsicherheit, ob richtig gerechnet wurde und damit zu entsprechenden Zusatzschleifen. Daraus ergibt sich die nächste Hürde: Wie kann sicher-

[4] Abrufbar unter: https://www.kollektivvertrag.at/kv/denkmal-fassaden-und-gebaeudereiniger-arb/denkmal-fassaden-und-gebaeudereiniger-rahmen/265108 (abgefragt am 4. Januar 2019).

[5] Vgl. Die unterschiedlichen Interpretationen zur Frage zu Überstunden bei Gleitzeit in Heilegger/Klein, Arbeitszeitgesetz, ÖGB Verlag, Wien 2016, S. 250 und Schrank, Arbeitszeit, Linde-Verlag, Wien 2018, S. 220–222.

[6] Vgl. Gantner, «CODE IS LAW» aber «IS LAW CODE»?, JusIT, 2018.

gestellt werden, dass die durchgeführte Rechnung auch im konkreten Fall (und nicht nur in den Tests) der Rechtsinterpretation folgt? Dies wird im Abschnitt «Nachvollziehbar Rechnen» dargestellt.

Das System ist nun im operativen Einsatz durch ca. 60 BeraterInnen der AK Wien. Abschließend werden bisherige Erfahrungen und nächste geplante Schritte bezüglich Rechtsthemen und der IT-Lösung dargestellt.

2. Rechtsinterpretationen explizieren & stabilisieren

Für Juristen ist es ein zentrales Element ihrer Arbeit, konkrete Lebenssachverhalte rechtlich zu beurteilen (Subsumtion). Im Zuge dessen kommen Rechtsinterpretationen zur Anwendung. Die in der Literatur und Rechtsprechung vertretenen Rechtsmeinungen unterscheiden sich häufig und können sich im Laufe der Zeit überdies verändern (durch Judikatur, Novellen, etc.). Zusätzlich folgt die Diskussion der Rechtsaspekte in zahlreichen Büchern, zumindest zum Teil, dem Aufbau der Norm. Die praktischen Fragen der Berechnung folgen aber oft einer anderen Logik.

Rechtsinterpretationen liegen typischerweise als Texte vor. Im Unterschied zu mathematischen Formeln und klaren ziffernmäßigen Ergebnissen erschwert sich dadurch der Vergleich mehrerer Rechtsinterpretationen.

In einer Arbeitsgruppe mit JuristInnen und InformatikerInnen experimentierten wir mit verschiedenen Verfahren am Beispiel konkreter Aufgabenstellungen (z.B. der Prüfung auf Einhaltung der täglichen Ruhezeit und, damit eng verwandt, der Bestimmung des Arbeitszeittages) und kamen zu folgenden Schlüssen:

- Eine Darstellung textueller Rechenregeln wurde schnell unübersichtlich.
- Eine Darstellung in Ablaufform («Flow-Chart») wurde selbst nach mehreren Anläufen aufgrund der Vielzahl an erforderlichen Rechenschritten und möglichen Sachverhaltskonstellationen leicht unübersichtlich. In den meisten Rechtsgebieten erfordert die Beurteilung eines Sachverhalts eine Vielzahl an Prüfungen und Festlegungen («Wenn… dann…, es sei denn …, dann …»). Das Arbeitszeitrecht ist dabei keine Ausnahme, sondern gehört nach unserer Einschätzung vielmehr zu den – gemessen an der Anzahl der zu prüfenden Aspekte – komplizierteren Rechtsgebieten. Dies ergibt sich schon daraus, dass alle Prüfschritte für jedes einzelne Zeitintervall wiederholt werden müssen und schon die Frage, welche Zeitintervalle überhaupt unterschieden werden müssen, einer eigenen Prüfung bedarf. Daher konnte die Darstellung in Ablaufform zwar als eine Art Startpunkt oder Grundgerüst sinnvoll genutzt werden. Eine Detaildarstellung aller möglichen Konstellationen und/oder Rechtsinterpretationen erfordert jedoch nicht nur enormen Zeitaufwand bei der Erstellung und laufenden Aktualisierung, sondern wird rasch unübersichtlich. Sie erwies sich daher nicht als eine für alle Beteiligten taugliche Repräsentationsform.

Nach mehreren Anläufen gelang ein Schritt nach vorne: Den zentralen Ansatz bildete dabei eine kompakte Darstellung von Sachverhalten und juristischen Bewertungen zu jeweils einer konkreten Berechnungsfrage in tabellarischer Form. Oder mit anderen Worten: Zu jeder Rechtsfrage (z.B. wann ist die Ruhezeit verletzt?) wurden verschiedenste Fallkonstellationen juristisch bewertet, kompakt dargestellt um die Lesbarkeit zu erhöhen, und gleichzeitig mit den Rechenregeln abgeglichen.

Dieser Ansatz soll am folgenden Beispiel der Prüfung auf eine Verletzung der täglichen Ruhezeit erläutert werden. Konkret war zu prüfen, ob die Ruhezeit von 11 Stunden zwischen zwei Arbeitszeittagen eingehalten wurde.[7]

[7] Vgl. Heilegger/Klein, Arbeitszeitgesetz, 4. Aufl., ÖGB Verlag, Wien 2016, S. 179.

BEISPIEL 1a: Berechnung der Grenzen des Arbeitszeittages mit Erklärungskomponenten und Verweis auf Literatur

Variable	Werte	Hinweise & Erklärungen	Notiz
Beginn AZ-Tag aus bisher betrachteter AZ	21.12 08:30		
Ende AZ-Tag aus bisher betrachteter AZ	21.12 20:30		
Beginn nächstes AZ-Intervall	22.12 05:30		
Ende nächstes AZ-Intervall	22.12 20:30		
Mindestruhezeit Arbeit	11,00 h		
Beginn neuer AZ-Tag weil ausreichend Ruhezeit	☐	Da keine ausreichende Ruhezeit vorliegt, verlängert sich der Arbeitszeittag.	Heilegger & Klein. AZG. 4. Auf. S179
RECHTSVERLETZUNG mehr als 24h ohne ausreichend Ruhezeit	☑	Der Zeitraum von Beginn der ersten Arbeit bis Ende des zweiten Blockes würde 24-Stunden überschreiten. Es beginnt daher nach 24-Stunden ein neuer Arbeitszeittag.	Heilegger & Klein. AZG. 4. Auf. S179
neuer Beginn betrachteter AZ-Tag	**So 22.12 08:30**	Es ist ein neuer Arbeitszeittag.	
neues Ende betrachteter AZ-Tag	**So 22.12 20:30**		

BEISPIEL 1b: Einige Beispiele aus der Fallsammlung (gegenwärtig 13 Fälle) zu Ruhezeitverletzung und Arbeitszeittag.

Variable	Werte	Fall 1	Fall 2	Fall 3
Beginn AZ-Tag aus bisher betrachteter AZ	21.12 08:30	21.12 08:30	21.12 08:30	21.12 08:30
Ende AZ-Tag aus bisher betrachteter AZ	21.12 20:30	21.12 12:30	21.12 12:30	21.12 20:30
Beginn nächstes AZ-Intervall	22.12 05:30	21.12 13:30	22.12 07:30	22.12 05:30
Ende nächstes AZ-Intervall	22.12 20:30	21.12 14:30	22.12 12:30	22.12 20:30
Mindestruhezeit Arbeit	11,00 h	11,00 h	11,00 h	11,00 h
Beginn neuer AZ-Tag weil ausreichend Ruhezeit	☐	☐	☑	☐
RECHTSVERLETZUNG mehr als 24h ohne ausreichend Ruhezeit	☑	☐	☐	☑
neuer Beginn betrachteter AZ-Tag	**So 22.12 08:30**	21.12 08:30	22.12 07:30	22.12 08:30
neues Ende betrachteter AZ-Tag	**So 22.12 20:30**	21.12 14:30	22.12 12:30	22.12 20:30

Abbildung 1: Aufbereitung von Sachverhalten, Materialien und Fällen zum Beispiel Bestimmung Beginn und Ende des Arbeitszeittages sowie Prüfung auf Ruhezeitverletzung

Das Beispiel mag einfach wirken, die juristische Diskussion zum Thema war aber facettenreich und es dauerte einige Stunden bis eine gemeinsame Rechtsinterpretation entwickelt wurde. Der Vorteil dieser kompakten, tabellarischen Darstellung ist einerseits die Übersichtlichkeit für alle Beteiligten, welche besonders für die Entwicklung einer gemeinsamen Sichtweise unverzichtbar ist. Andererseits ist es eine Darstellung, die auch mit geringer juristischer Expertise schnell und einfach zugänglich ist. Darüber hinaus kann – automationsunterstützt – schnell geprüft werden, ob die gewählten Rechenregeln für sämtliche bisher bedachten (und neu auftretenden) Sachverhaltskonstellationen das gewünschte Ergebnis liefern. Auf diese Art können sowohl die Rechenregeln, als auch die Rechtsinterpretation weiter verfeinert werden. Schlussendlich kann die juristische Diskussion (z.B. Gesetztestexte, Literaturzitate, …) mit eingepflegt werden und muss nicht jedes Mal neu zusammengesucht werden.

3. Verknüpfung Rechtsinterpretation und Berechnung

Die Verknüpfung von Rechtsinterpretationen und Berechnung ist zusätzlich zu den oben angeführten Schwierigkeiten auch von IT-Seite her problematisch: Typischerweise unterscheiden sich nämlich Umsetzungen einer Aufgabenstellung in Computerprogramme durch verschiedene ProgrammiererInnen oft erheblich voneinan-

der. Selbst bei identischen Werkzeugen setzen verschiedene Personen Aufgaben sehr unterschiedlich um und entsprechend gibt es auch von dieser Seite viel Raum für Missverständnisse und Fehler. Zusätzlich ist auch für geschulte und geübte Personen das Nachvollziehen und Durchdenken von Programmen (der eigenen und noch mehr der von anderen) schwierig und aufwändig. Für Personen ohne entsprechenden Hintergrund, was für viele JuristInnen gelten dürfte, ist dies praktisch unmöglich. Um Missverständnisse zu vermeiden muss aber auch das Ergebnis der Umsetzung einer Rechtsinterpretation leicht prüfbar sein.

Um diese Kluft zwischen Programmierung und Rechtsinterpretation möglichst gut zu überbrücken, entwickelten wir in mehreren Anläufen folgende Elemente:

- Eine kompakte Darstellung, die es sehr einfach und interaktiv erlaubt die verwendeten Rechenregeln auf die definierten Fälle anzuwenden und automatisiert Unterschiede zwischen einer Interpretation der JuristInnen und der Realisierung im Programm zu zeigen. Damit wird es sehr leicht, die Programmierung mit der aktuellen Fallsammlung zu vergleichen.
- Für eine direkte Eingabe von Formeln wird eine Excel-Oberfläche verwendet, da dies wohl eine der bekanntesten Umgebungen darstellt. Damit ist es auch wenig geübten AnwenderInnen möglich weitere Fälle zu definieren und zu prüfen, ob die Programmierung die gewünschten Ergebnisse bringt.

Im Forschungsprojekt für den KV Reinigung konnte ein großer Teil der Berechnungsfragen auf diese Weise abgebildet werden. Trotzdem war bei manchen Fragen die Erstellung kleiner spezifischer Programme erforderlich, welche in der Programmiersprache Python realisiert wurden. Im Testablauf können diese gleich wie die EXCEL-Formeln getestet werden.

Diese Teilautomatisierung führt zu schneller Generierung neuer Fälle, die dann immer wieder zu neuen juristischen Diskussionen und dann wieder zu Verfeinerungen führen. Nach einiger Zeit wurden aber neue Fallunterscheidungen sowie sichtbar gewordene Missverständnisse seltener.

Mit diesem Vorgehen ist keine 100%-Fehlerfreiheit garantiert. Es ist aber von hoher und im Laufe der Zeit zunehmender Stabilisierung von und Übereinstimmung mit Rechtsinterpretationen und ihrer Abbildung als Rechenwerk auszugehen.

4. Nachvollziehbar Rechnen

Die Verbindung von Rechenregeln und Fällen bewährte sich für die isolierte Abbildung der einzelnen Vorschriften und der dazugehörigen Rechenregeln. Im Zuge der Anwendung wurde aber eine weitere Hürde sichtbar: Bei Anwendung mehrerer Rechenregeln war es selbst für Projektmitglieder nicht immer leicht nachzuvollziehen, wie diese zusammenspielen und wie ein Gesamtresultat zustande gekommen war.

Das führte zu Irritationen und schlussendlich zur Einsicht, dass das Bemühen um «richtiges Rechnen» nicht genügt, sondern dass es zusätzlich erforderlich ist, den Rechenweg nachvollziehbar darzustellen, also auch die Zwischenschritte beschreibend zur Verfügung zu stellen.

Diese Idee wurde in einem «Journal» realisiert, in dem die rechtliche Bewertung jeder Zeitbuchung (Eingabe) möglichst nachvollziehbar dargestellt wird. Das folgende Beispiel illustriert die Berechnung von Mehrstundenzuschlägen und ihrem allfälligen Ausgleich.

BEISPIEL: Mehrstundenzuschläge
Der KV Reinigung sieht abweichend von § 19d Abs. 3b Z 1 AZG vor, dass Mehrstunden nicht zuschlagspflichtig sind, wenn sie innerhalb von drei Monaten durch Zeitausgleich (1:1) abgebaut werden. Diese 3-monatige Frist läuft ab Leistung der jeweiligen Mehrstunde. Wurde eine Mehrstunde jedoch nicht ausgeglichen, steht ein Zuschlag von 25% zu. Dieser Zuschlag kann sodann – genauso wie die zugrundeliegende Mehrstunde – ebenfalls durch Zeitausgleich konsumiert werden. Bei der Berechnung der anfallenden Zuschläge hat man es daher einerseits mit Salden zu tun (z.B. Wochensalden), aber auch mit Fristen. Bei schwankenden Arbeitszeiten ist dies eine sehr herausfordernde Berechnung. In der Realisierung wird in der Version der AK Wien bei zuschlagspflichtig gewordenen Mehrstunden im Journal folgender Text angezeigt:

Fr 22.01.2016 22.01.2016 . 8:00-12:00 . . . . . . 4,00 Std. .

8:00	-11:00	3,00	AZ exkl ÜStd (ohne FtgsArb)	-
11:00	-12:00	1,00	AZ exkl ÜStd (ohne FtgsArb)	MStd

Zuschlag 25% für 1.00 Std. am 23.04.2016 00:00 da 3 Monate ohne Ausgleich
0.25 Std. MStd Zuschlag Ausgleich mit Diff zu wöchentl. NAZ vom 29.04.2016 12:00
0.75 Std. MStd Ausgleich mit Diff zu wöchentl. NAZ vom 29.04.2016 12:00

Die erfasste Arbeitszeit von 8:00 bis 12:00 wurde von der Software automatisch qualifiziert. Während die Zeit bis 11:00 noch Normalzeit ist, wurde die Zeit von 11:00–12:00 als Mehrstunde erkannt. Da in diesem Testbeispiel kein Ausgleich binnen 3 Monaten erfolgte, wurde der Mehrstundenzuschlag von 25% mit Ablauf des 22. April 2018 (sohin um 0:00 Uhr des 23. April 2018) wirksam. Dieser Zuschlag wurde anschließend zusammen mit der Grundstunde durch Zeitausgleich am 29. April 2018 konsumiert.

Abbildung 2: Beispiel für nachvollziehbare Darstellung der Zwischenschritte einer Berechnung

Der Zugang nachvollziehbar zu rechnen bewährte sich und machte eine Reihe von Fehlern in der Gesamtberechnung sichtbar, die bis dahin unentdeckt geblieben waren. Besonders amüsant war ein Fehler bei der Fristberechnung von 3 Monaten: Aus Informatikperspektive war 3 Monate nach dem 21. Januar 2016 um 13:00 der 21. April 2016, 13:00. Aus Juristenperspektive (§ 902 Abs. 1 ABGB) ist es aber der 22. April 2016 00:00.

In anderen Fällen führte genau diese Diskrepanz zwischen den beiden Perspektiven zu juristisch höchst komplexen und bisher – soweit ersichtlich – nicht behandelten Rechtsfragen:

So sieht der KV Reinigung in § 10 Abs. 4 vor, dass die «Verrechnung der Überstundenarbeit viertelstundenweise [erfolgt und] Bruchteile einer viertel Stunde auf eine viertel Stunde aufgerundet werden». Zusätzlich findet sich in § 11 Abs. 6 («Allgemeine Lohnbestimmungen») die wortgleiche Anordnung für die «Verrechnung der Stundenlöhne». Daraus ergibt sich eine Vielzahl an möglichen Auslegungen: Wird sowohl Normalarbeitszeit, als auch Überstundenarbeit aufgerundet? Was gilt, wenn sich unterschiedliche Intervalle nur wegen der Unterscheidung zwischen Normalarbeitszeit und Überstunden ergeben (z.B. 8:10 Std. NAZ und 1:05 Std. ÜST)? Wird einmal pro Arbeitszeittag oder einmal pro Zeitintervall (z.B. 8–10:13 Uhr und 16:00–18:02 Uhr) aufgerundet? Und erfolgt die Rundung durch fiktive Verschiebung der Beginn- und Endzeit oder durch Rundung der Zeitsumme (beides kann unterschiedliche Auswirkungen auf die Zuschlagshöhe haben)? All diese Fragen müssen nicht nur gelöst, sondern vielmehr auch im Zuge der – abstrakten – Erstellung der Rechenregeln erkannt werden.

5. Gesamtsicht und Ausblick

Insgesamt lässt sich der Lernprozess über mehrere Jahre wie folgt beschreiben (siehe Abbildung 3): Ausgangspunkt war eine naive Sicht der Informatik, die Unterschiede von Rechtsinterpretationen als vernachlässigbare Unschärfe interpretierte. Das führte zu aufwändiger mehrfacher Programmierung und konnte mit der ersten Verfeinerung überwunden werden, mit denen Interpretationen an Hand von Fällen sichtbar und vergleichbar wurden. Als zweite Verfeinerung war darauf aufbauend die Automatisierung des Vergleichs der Implementierung zu den Fällen möglich (da nun explizit vorliegend) und erforderlich, eine Art von Unit-Testing[8] in Richtung einer testgetriebenen Softwareentwicklung[9]. Im dritten Schritt ging es um Nachvollziehbarkeit, die Voraussetzung für die Arbeit mit großen Beispielen und damit eine Art von Integrationstest[10]. Gleichzeitig erlauben diese Tests auch sehr gut, einzelne Versionen der Rechtsinterpretation wie auch einzelne Softwareversionen miteinander zu vergleichen und erlauben so auch eine Art von Regressionstesting[11].

In der Übersicht lässt sich die Erweiterung des Verständnisses wie folgt in Abbildung 3 darstellen.

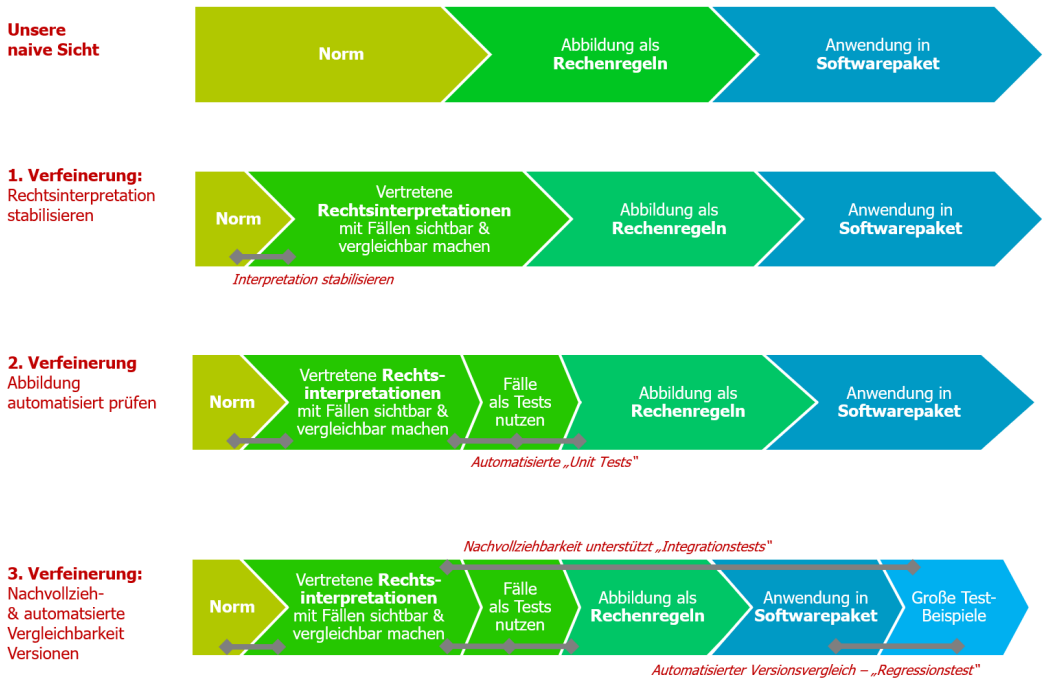

Abbildung 3: Entwicklung des Problemverständnisses und der technischen Umsetzung des Arbeitszeit-Rechners

---

8 Möglichst automatisierter Test kleiner zusammenhängender Funktionalitäten auf Korrektheit – z.B. Test auf Ruhezeitverletzung der täglichen Ruhezeit.

9 Vgl. Beck, Test Driven Development by Example, Addison-Wesley Verlag, 2002.

10 Spielen die einzelnen Komponenten wie gewünscht zusammen, so dass das Gesamtergebnis passt?

11 Beim Regressionstest wird geprüft, ob sich etwas – und wenn ja, was sich – zu früheren Durchläufen verändert hat, damit soll unabsichtliche Änderung vermieden werden.

In Summe wurden mit Stand Oktober 2018 rund 720 verschiedene Fälle (inkl. Test und Schulungsfälle) ausgewertet. Regelmäßig ergaben sich mit der Erweiterung des Nutzerkreises neue juristische Fragen, die zu kleineren Anpassungen, Erweiterungen und Korrekturen führten.

Im Hinblick auf die zukünftige Entwicklung sind inhaltliche Erweiterungen bezüglich der abgebildeten Rechtsgrundlagen sowie Vereinfachungen in der Dokumentation der rechtlichen Aspekte und ihre anschließende Einbettung in Berichte angedacht (z.B. Hinweis auf Ruhezeitverletzung führt mit Link direkt zur Darstellung der zur zugrundeliegenden Rechtsansicht). Absehbar sind auch bereits Herausforderungen des Versionsmanagements: Vor allem wenn sich nicht Rechenregeln ändern, sondern zusätzliche Sachverhaltskonstellationen berücksichtigt werden sollen, müssen nicht nur entsprechende Eingabemöglichkeiten geschaffen, sondern auch (möglichst automatisiert) geprüft werden, ob und wie neue Regeln auf alte Datenbestände anwendbar sind. Im Ergebnis stellt dies jedoch die Basis für großflächige Vergleiche dar («big data»).