

# Feature 5: Erfassen von Policy Violations

A. Heckl, A. Kohles, A. Lehene, A. Mütter

Februar 2020

## 1 Deskriptive Systembeschreibung

In einem Dienstleistungsunternehmen gibt es immer wieder Mitarbeiter, die sich nicht an Vorgaben vom Arbeitgeber halten. Zu möglichen Regelverletzungen zählen die Überschreitung der maximalen Arbeitszeit, die Nicht-Einhaltung von Pausenzeiten oder die Nicht-Einhaltung der Bearbeitungsprioritäten von Aufgaben. Diese können zu unnötigen Verzögerungen von Projekten führen, aber auch zu arbeitsrechtlichen Konsequenzen. Dem Arbeitgeber ist es daher wichtig, dass solche Verletzungen systematisch erkannt und erfasst werden, um zum einen Projektverzögerungen gegenüber den Kunden des Unternehmens zu vermeiden, und zum anderen aber auch um die Mitarbeiter vor Überlastungen oder rechtlichen Konsequenzen zu schützen. Neben Arbeitgeber und Arbeitnehmer ist dies auch für den Betriebsrat und die Rechtsabteilung des Unternehmens von Relevanz. Zur Umsetzung dieser Erfassung möchte der Arbeitgeber Jira Issues überwachen und anhand derer prüfen, ob Verstöße vorliegen. Die verübten Verletzungen sollen grafisch aufbereitet werden und den Mitarbeitern zur Verfügung gestellt werden, sodass diese erkennen, in welchen Bereichen die meisten Verstöße stattfinden und wo schlussendlich am meisten Handlungsbedarf besteht.

## 2 Wertekonflikte

### 2.1 Fragestellungen

Vor der Umsetzung einer solchen Maßnahme gibt es jedoch aufkommende Fragestellungen:

- Darf man Mitarbeiter überwachen, um herauszufinden, ob sie alles so machen wie es sich der Arbeitgeber wünscht?
- Dürfen die so gewonnenen Ergebnisse im Zweifel gegen den Mitarbeiter verwendet werden?
- Sieht nur der Mitarbeiter selbst eine Warnung / Abmahnung oder wird das gesamte Team oder weitere Personen benachrichtigt?

**Hinweis:** Wir waren uns bei der Formulierung des Tagebuchs und des Abschlussberichts über die Richtlinien der gendergerechten Sprache und der gendergerechten Formulierung bewusst. Aus Gründen der Übersichtlichkeit und der einfacheren Lesbarkeit haben wir uns entschieden darauf hinzuweisen, dass in diesem Dokument mit allen männlichen Formulierungen von Personen (Angestellter, Mitarbeiter, Vorgesetzter) selbstverständlich auch die weiblichen Pendanten gemeint sind, und diese in keiner Weise vernachlässigbar oder weniger bedeutungsvoll sind.

- Dürfen diese Daten überhaupt erhoben werden?
- Sollen Mitarbeiter durch derartige Benachrichtigungen beeinflusst werden?

## 2.2 Biases

Zunächst wollen wir uns mit den sogenannten Preexisting Biases auseinandersetzen. Es ist anzuführen, dass Arbeitnehmer im Unternehmen einen Anspruch auf selbstständiges und unabhängiges Arbeiten haben. Allerdings dient dieses System auch zum Schutz der Mitarbeiter (auch zum Schutz vor sich selbst). Außerdem führt eine Verbesserung der Produktivität zu mehr Wohlfahrt für das gesamte Team und folglich auch für das gesamte Unternehmen. Zudem werden begangene Verstöße durch das Tool sofort einsehbar, viele davon wären im Firmenumfeld früher oder später sowieso erkannt worden, sodass nicht nur mehr Verletzungen erkannt werden, sondern auch früher. Darüber hinaus sind firmeninterne Regeln oder gesetzliche Vorgaben nur dann sinnvoll, wenn diese auch eingehalten werden, weshalb eine Kontrolle hierfür nötig ist und ein Jira Plugin eine weniger eingreifende Überwachung ist als beispielsweise ein Kamerasystem. Allerdings kann es auch sein, dass die Priorität der Aufgaben nicht immer ausschlaggebend für die Bearbeitung der Tasks sein muss. Kurze Tätigkeiten mit niedriger Priorität können beispielsweise auch schnell zwischendurch erledigt werden, ohne erhebliche Einbußen in der Produktivität.

Zusätzlich lassen sich auch Technical biases feststellen: Aktivitäten in Jira spiegeln nicht zwangsläufig immer die genauen Arbeitszeiten ab und bieten nur eine eingeschränkte Sichtweise auf das Arbeitsverhalten von Personen. Es können Informationen fehlen und Ungenauigkeiten bei der Auswertung entstehen. So können fehlerhafte Informationen (false positives) zu unnötigem Stress für die Mitarbeiter führen.

Weiterhin gibt es auch Emergent biases, die für dieses Gadget aufkommen: Durch ein derartiges Tool können sich Mitarbeiter unter größeren zeitlichen Druck gesetzt fühlen, weil jede ihre Aktionen im System erfasst wird. Zudem kann durch die Art der Auswertung eine ungerechtfertigt negative Sichtweise auf den Mitarbeiter entstehen, beispielsweise wenn zwar die Prioritäten missachtet wurden, aber dennoch alle Aufgaben vor ihren jeweiligen Deadlines erfüllt wurden. Außerdem können äußere Einflüsse (öffentliche Verkehrsmittel, Unwetter, ...), insbesondere bei der Erfassung der Arbeitszeit, zu Verzerrungen führen und ungerechtfertigte Konsequenzen nach sich ziehen.

## 2.3 Vortheoretische Deliberation

In Anbetracht der Fragestellungen und verschiedenen Biases, stellt sich die Frage ob und wie dieses Feature umgesetzt werden soll: Es kann einen Mehrwert für alle Beteiligten bringen, kann aber auch zum Missbrauch durch Vorgesetzte eingesetzt werden. Das System sollte nicht als alleiniges Merkmal für berufliche Entscheidungen (Beförderungen, Abmahnungen, Gehalt, ...) gegenüber Mitarbeitern dienen.

## 3 Ethische Systemüberprüfung

Im Folgendem wird die Umsetzung des Features aus verschiedenen Sichtweisen analysiert: zunächst deontologisch nach Kants kategorischem Imperativ und anschließend konsequen-

tialistisch nach utilitaristischem Konzept.

### **3.1 Deontologische Betrachtung**

Aus deontologischer Sicht ist es als Arbeitnehmer sehr wünschenswert, wenn man zunächst vom System gewarnt wird, bevor man wiederkehrend Verstöße oder Fehler macht. Dies führt nicht nur zu besserem Schutz vor arbeitsrechtlichen Folgen, sondern auch zu vermehrter persönlicher Verbesserung. Ein solches System greift jedoch stark in die Privatsphäre der Mitarbeitenden ein, indem selbst kleinste Arbeitsschritte überwacht werden. Hierdurch wird auch auf die Autonomie und die Selbstständigkeit der Arbeitnehmer Einfluss genommen. Er wird behandelt, als ob er nicht selbst in der Lage wäre derartige Regeln einzuhalten oder die Folgen deren Nichtbeachtung abzuschätzen und eventuell bewusst in Kauf zu nehmen. Zudem kann das Tool nicht außerhalb des Arbeitslebens universalisiert werden. Im Familien- oder Privatleben ist ein Einsatz beispielsweise nicht denkbar.

### **3.2 Konsequentialistische Betrachtung**

Aus konsequentialistischer Sicht gibt es ganz andere Gegenargumente: Durch ständige Überwachung der Arbeitsschritte erhält die Einhaltung der Regeln einen höheren Stellenwert als das produktive Arbeiten. Arbeitnehmer machen sich mehr Gedanken zu Policy Violations und fokussieren sich z.B. nicht mehr voll auf ihre Arbeit. Es können so mehr Fehler entstehen, wodurch in der Überwachung wieder ein verstärkter Fokus auf diesem Mitarbeiter liegt. Langfristig entstehen so Einbußen in der Produktivität. Des Weiteren führen Benachrichtigungen über begangene Fehler dazu, dass die Motivation zur Arbeit sinkt und somit auch die Produktivität sinkt. Zudem führen derartige Methoden zum internen Vergleich unter Kollegen, was eine negative Stimmung im Team hervorrufen kann. Beides wirkt sich negativ auf die Wohlfahrt des Teams und somit des Unternehmens aus. Außerdem bedeutet eine starke Kontrolle der Angestellten für ein Unternehmen auch immer einen Imageverlust, was sich langfristig auf Verkaufszahlen oder Kundenaufträge auswirken kann und somit die Wohlfahrt negativ beeinflusst. Zuletzt besteht nach Einführung des Tools keine Möglichkeit mehr, „Marathontage“ für das Unternehmen zu leisten: Insbesondere vor wichtigen Abgaben oder Deadlines arbeiten Mitarbeiter häufig mehr als die erlaubte Stundenzahl, um Projekte noch zu Ende zu bringen. Das Wegfallen dieser Möglichkeit führt nicht nur zum Verlust von Produktivität, sondern auch zu einem möglichen Imageverlust vor den Kunden, wenn Deadlines verlängert oder Projekte noch nicht ganz fertig übergeben werden müssen.

Allerdings gibt es auch zahlreiche Argumente, die für die Einführung dieses Systems sprechen: Eine Überwachung der Regeln führt zu deren Verstärkter Einhaltung und somit zu weniger Ermahnungen, Personalgespräche oder ähnlichen damit verbundenen Konsequenzen. Die Produktivität steigt dadurch also und bedeutet mehr Wohlfahrt für das Unternehmen, welches dieses durch mehr Gehalt oder weniger Arbeitszeit auf das Team übertragen kann. Die Einhaltung der Regeln führt bei den Mitarbeitern auch zu einer Verbesserung des Moralgefühls, insbesondere wenn im Plugin keine Warnungen wegen Verstößen auftauchen. In der ganzen Gruppe führt dies zu einer verbesserten Arbeitsmoral. Das System ist zudem ein wichtiger Schutz für den Arbeitnehmer vor arbeitsrechtlichen Konsequenzen. Verstöße gegen das Arbeitsschutzgesetz haben nicht nur negative Auswirkungen auf das Unternehmen, sondern können für den Angestellten auch zu be-

rufflichen Konsequenzen, wie z.B. Abmahnungen oder Kündigungen führen. Zuletzt kann der Vergleich der Warnungen mit den Kollegen motivieren, weniger solcher Warnungen zu erzeugen und sorgfältiger zu arbeiten. Zum einen ist ein Gamification Aspekt vorhanden, welcher die Motivation erhöht, zum anderen ein schlechtes Gewissen, wenn man im Vergleich sehr weit zurückliegt.

### 3.3 Theoretische Deliberation

Aus kategorischer Sicht ist eine Umsetzung des Tools möglich. Es ist wichtig zu regeln, wer Zugang zu den gesammelten Daten erhält, um eine totale Überwachung und einen zu starken Eingriff in die Autonomie der Personen auszuschließen. Aus konsequentialistischer Sicht kann das System einen erheblichen Mehrwert bringen, wenn es korrekt eingesetzt wird.

## 4 Urteilsphase

Beide ethischen Ansichten liefern ähnliche Ergebnisse: Die Umsetzung des Plugins ist unter bestimmten Umständen sinnvoll. In beiden Fällen ist eines der Hauptgegenargumente die starke Kontrolle seitens des Arbeitgebers. Das Tool muss somit so eingesetzt werden, dass der Arbeitnehmer davon profitiert, ohne direkte Konsequenzen von seiner Führungskraft erwarten zu müssen und somit nicht allzu stark in seiner Selbstständigkeit eingeschränkt wird. Dann nämlich kann er selbst Warnungen zu Verstößen ernst nehmen und seine eigene Arbeitshaltung verbessern und schlussendlich auch seine Produktivität und Arbeitsleistung erhöhen. Insbesondere der Aspekt zum Schutz des Arbeitnehmers vor rechtlichen Konsequenzen ist ein positives Feature des Tools und verbessert die Wohlfahrt jedes betroffenen Individuums und rechtfertigt die Einflussnahme des Tools auf die Mitarbeiter.

Das Plugin soll somit umgesetzt werden unter der Prämisse, dass Führungskräfte oder Arbeitgeber erst nach starken Häufungen von Policy Violations benachrichtigt werden, also wenn der Mitarbeiter die Richtlinien trotz Hinweisen nicht einhalten kann oder will. Um die Eingangs aufgeworfene Frage, ob Ergebnisse gegen den Mitarbeiter verwendet werden dürfen, zu klären: Ja die erhobenen Daten können im Zweifel gegen einen Arbeitnehmer verwendet werden, jedoch nur, wenn dieser über längere Zeit hinweg Regelbrüche anhäuft.

## 5 Technische Umsetzung

### 5.1 Verwendete Daten

Im Abgabe-Ordner für dieses Feature befindet sich die Datei „Red Hat Issue Tracker 2020-01-15T13.30\_43-0500.csv“. Diese beinhaltet Daten zu Jira Issues eines Open Source Softwareentwicklungsprojektes. Die Datei beinhaltet etwa 360 Issues im Zeitraum von November bis Januar. Diese Daten müssen zur Verwendung unseres Features zunächst in das eigene Jira System importiert werden (siehe Anleitung im README). Zu Testzwecken sollten nur genau diese Daten in Jira importiert sein (anderweitig vorhandene Projekte im eigenen Jira System löschen).

## 5.2 High Level Funktionalitäten

Die Issue Daten aus 5.1 werden mittels der Jira API gefetcht und in 2 Diagrammen visualisiert. Diese Diagramme stellen eine Auswertung auf Team Ebene dar, d.h. es werden nur die Violations des gesamten Teams dargestellt, Rückschlüsse auf einzelne Mitarbeiter sind somit nicht möglich. Es werden 3 Arten von Policy Violations getrackt:

1. Night Violation: wenn ein Mitarbeiter nach 20 Uhr und vor 6 Uhr in Jira aktiv war.
2. Weekend Violation: wenn ein Mitarbeiter am Wochenende in Jira aktiv war.
3. Priority Violation: Wenn ein Mitarbeiter eine Aufgabe mit niedriger Priorität vor einer Aufgabe mit hoher Priorität erledigt, obwohl diese noch offen ist.

Zu guter letzt werden in einem persönlichen Auswertungsbereich die Anzahl der Violations jeder Violation-Kategorie des aktuell eingeloggten Mitarbeiters im letzten Monat angezeigt. Je nach Anzahl der Violations wird entweder ein grüner, gelber oder roter Alert angezeigt.

## 5.3 Grafische Darstellung

Mit Hilfe des Plugins Charts.js<sup>1</sup> wurden in Javascript zwei Diagramme erzeugt. Nach dem Import des Plugins, sowie dem Anlegen des Charts in einem Canvas im HTML Part, kann dieses dynamisch aus dem Javascript File befüllt werden. Im ersten Diagramm werden per Liniendiagramm die Anzahl der 3 Arten von Violations in der Firma in einem auswählbaren Zeitraum angezeigt. Die Auswahl hierfür erfolgt mittels zwei Drop Down Menüs: Eines für den Zeitraum (letzte 4, 6 oder 12 Wochen) und eines für die Arten der Violations (Priority, Night, Weekend, Gesamt). Nach jeder Auswahl wird das Diagramm dynamisch aktualisiert und der entsprechende Verlauf angezeigt. Die vorherangezeigten Daten müssen aus dem dataset-array des Charts entfernt werden (mittels pop()). Die neu anzuzeigenden Daten müssen in das dataset-array des Charts gepusht werden. Das aktuell ausgewählte Intervall wird per globaler Variable gesichert. Zur Initialisierung des ersten Diagramms wird zu Programmstart die Anzahl der gesamten Policy Violations in den letzten 4 Wochen angezeigt.

Das zweite Diagramm ist statisch und bezieht sich auf die Arten der verschiedenen Verstöße. Hierbei wird mittels eines „RadarCharts“ visualisiert, wie hoch der Anteil an den unterschiedlichen Kategorien der Verstöße ist. Das Diagramm wird zu Programmstart mit den entsprechenden Daten aus den bereitgestellten Funktionen bestückt und während der Laufzeit nicht weiter aktualisiert. Zum Erkennen von Entwicklungen werden die Daten aus dem Vormonat ebenfalls im selben Diagramm mit angezeigt. Durch die Verwendung transparenter Farben bleibt die Grafik dennoch übersichtlich.

Die weiteren Verwendeten Optionen der Diagramme (Padding, Anzeigen von Labels und Legenden, Schrittweiten der Diagramme, ...) sind dem Programmcode zu entnehmen. Wichtig ist anzumerken, dass der Parameter „responsive“ auf „false“ gesetzt wurde, sodass die Diagramme beim Zoomen im Browser nicht mit skalieren, um eine sinnvolle Anzeige auf allen Displaygrößen zu gewährleisten (da bei sehr kleinen Bildschirmen Teile des Diagramms ohne Zoom nicht zu lesen waren).

Die notwendigen Änderungen im Style wurden direkt im gadget.xml file vorgenommen, um den Transfer der Dateien übersichtlich und kompakt zu halten. Eine Einbindung in den

---

<sup>1</sup><https://www.chartjs.org/>

CSS Ordner hätte eine zusätzliche Änderung im `atlassian-plugin.xml` file bedingt. Daher wurde es der Einfachheit halber in den HTML Teil integriert.

## 5.4 Manipulation der Open Source Daten

An zwei Stellen beschlossen wir uns die Daten aus dem Open Source Projekt, welche wir in unserem Plugin fetchen, zu manipulieren.

Erstens haben wir die Datumsangaben der Jira Issues im Feld `Updated` insofern abgeändert, dass eine gleichmäßigere Verteilung vorliegt als dies ursprünglich der Fall war. Im Original sind die Daten nämlich an wenigen Tagen im November sehr stark gehäuft und im Dezember und Januar nur sehr rar. Damit die Diagramme nicht überwiegend extreme Schwankungen darstellen oder gar einige Nullwerte, haben wir uns für diesen Manipulationschritt entschieden. Dieser hat keine Auswirkungen auf die Funktionalität des Plugins. Die entsprechende funktion im Javascript Code nennt sich *fakeDate*.

Der zweite Eingriff bezieht sich auf den dritten Teil des Features, also die persönliche Analyse. In der Praxis würde hier der aktuell eingeloggte User ermittelt werden und seine entsprechende Auswertung angezeigt werden. Da uns im Rahmen des Praktikums nur den Account mit den Credentials *admin/admin* zur Verfügung steht, haben wir alle vorkommenden Usernamen aus dem OpenSource Projekt ermittelt und geben einen dieser Namen per Zufallsfunktion mit jedem neuen Laden des Features bzw. Browserrefresh aus. Die entsprechende Funktion im Javascript Code nennt sich *getRandomEmployee*.

## 6 Testfälle

Bei der Ersterstellung des Plugins wurde zunächst ausgiebig mit Testdaten aus OpenSource Projekten gearbeitet. Hierbei bemerkten wir, dass in verschiedenen Projekten das Befüllen von Feldern eines Jira Issues zu sehr unterschiedlichen Graden vollständig ist. Oftmals ist beispielsweise kein *Assignee* zugeordnet. Wir haben also in unseren Javascript Code die Funktionalitäten so umgesetzt, dass die von uns verwendeten Felder eines Issues auf null Werte geprüft werden. Sollte dies der Fall sein, wird der entsprechende Issue übersprungen und fließt nicht in die Analyse ein. Aufgrund der von uns beobachteten stark unterschiedlichen Verwendung von Jira in den OpenSource Projekten leiten wir ab, dass in der Praxis für jedes Unternehmen einzeln festgestellt werden muss, welche Felder sich genau für die Auswertung eignen und welche nicht.

Somit ist festzuhalten, dass unsere Implementierungen für die zuvor genannten Import-Daten funktionieren, es ist uns jedoch bewusst, dass wegen der im vorigen Absatz genannten Gründe im Feature an gewissen Stellen wenig sinnvolle Ergebnisse auftauchen können, wenn man es mit anderen OpenSource Projekten bestückt. Unserer Einschätzung nach legt das primäre Ziel des Praktikums wenig Augenmerk darauf, ein Feature zu implementieren, welches für eine Vielzahl an Projekten, die allesamt Jira Issues stark unterschiedlich befüllen, stabile und sinnvolle Ergebnisse liefert. Daher haben wir uns auf eine sinnvolle Umsetzung mit den von uns beschriebenen Import-Daten aus dem wildFly Projekt konzentriert.