

6 Evaluation

Inhaltsangabe

6.1	Ziele und Vorgehensweise	73
6.2	Evaluation mit Kooperationspartner A	74
6.3	Evaluation mit Kooperationspartner B	77
6.4	Zusammenfassung	79

Dieses Kapitel beschreibt die Ziele, die Vorgehensweise und die Ergebnisse der Evaluation zu der in dieser Diplomarbeit entstandenen Werkzeugunterstützung *River*.

6.1 Ziele und Vorgehensweise

Ziel der Evaluation ist die Beantwortung der Frage, ob und inwiefern das Werkzeug *River* die Stakeholder bei der Analyse und Optimierung eines Softwareentwicklungs-Prozesses unterstützt. Kann ein Stakeholder über die Visualisierung der automatisch erhobenen Ticket-Daten und Ticket-Historien eines Change-Request-Systems Rückschlüsse auf Vorgänge im Softwareentwicklungs-Prozess, der laut der zentralen Hypothese dieser Arbeit durch das Change-Request-System abgebildet wird, ziehen? Falls ja, soll die Evaluation insbesondere ermitteln, welche Rückschlüsse möglich sind?

Die Evaluation wurde in Form von qualitativen, semi-strukturierten Einzelinterviews im Rahmen von zwei Sitzungen mit jeweils einem Mitarbeiter der beiden Kooperationspartner dieser Diplomarbeit durchgeführt. Die befragten Mitarbeiter decken die identifizierten Rollen der Stakeholder, Projektleiter und Softwareprozess-Manager, ab (vgl. Abschnitt 4.2). Der Aufgabenbereich von Stakeholder A beinhaltet beide Rollen. Als Geschäftsführer eines mittelständischen IT-Unternehmens führt er Projekte durch und besitzt gleichzeitig ein hohes Interesse an der Optimierung der im Unternehmen eingesetzten Prozesse. Stakeholder B erfüllt die Rolle eines Softwareprozess-Managers in einem großen IT-Unternehmen und ist mit vielen dort durchgeführten Softwareentwicklungs-Projekten vertraut. Beide Stakeholder sind Experten für die im Rahmen der Evaluation mit *River* zu untersuchenden Softwareentwicklungs-Prozesse aus ihrem jeweiligen Unternehmen.

Das qualitative Interview wurde aus zwei Gründen einer quantitativen Evaluation über einen standardisierten Fragebogen vorgezogen:

1. Der Fokus der Evaluation liegt darauf herauszufinden, welche Rückschlüsse auf einen Softwareentwicklungs-Prozess mittels *River* möglich sind. Das qualitative Interview mit einem Experten eignet sich insbesondere dazu, explorativ Erkenntnisse zu gewinnen, die dem Interviewer unbekannt sind (siehe [May08]). Hier ermöglicht es mit Hilfe des befragten Experten, vom Diplomanden vermutete Zusammenhänge zwischen der Visualisierung und den dem Experten vertrauten

Softwareentwicklungs-Prozess zu bestätigen oder zu widerlegen, sowie unerwartete Zusammenhänge zu erkennen, die der Experte erkennt.

2. Die Stärke des standardisierten Fragebogens, die erhobenen Daten statistisch aufzubereiten, kommt durch die geringe Teilnehmer-Anzahl (zwei) an der Evaluation nicht zum Tragen.

Ablauf des Interviews

Vor dem Interview wurden die durch den jeweiligen Kooperationspartner zur Verfügung gestellten Daten des entsprechenden Change-Request-Systems in das Werkzeug importiert und Vermutungen über Eigenschaften des Softwareentwicklungs-Prozesses auf Basis der Visualisierungen angestellt.

Zu Beginn des Interviews wurde der Experte mit der Bedienung des Werkzeugs und der Semantik der Visualisierung vertraut gemacht. Im zweiten Schritt wurden dem Experten mit Hilfe des Werkzeugs die Visualisierungen präsentiert, zu denen zuvor Vermutungen über ihre Aussagekraft bezüglich des Softwareentwicklungs-Prozesses angestellt wurden. Der Experte sollte diese Vermutungen - auf Basis seiner genauen Kenntnisse des dargestellten Prozesses - kommentieren und entweder bestätigen oder widerlegen. Schließlich sollte der Experte das Werkzeug selbständig und frei verwenden, und den Prozess weiter analysieren. Hierbei sollten Erkenntnisse über weitere Zusammenhänge zwischen Visualisierung und Prozess gewonnen werden.

Während des Interviews wurden zusätzlich die Kommentare des Experten zum Werkzeug selbst festgehalten. Darunter fallen Bemerkungen über die Benutzbarkeit, Verständlichkeit und Eignung des Werkzeugs, der Visualisierungen oder einzelner Anzeigeelemente, sowie Hinweise auf mögliche Verbesserungen.

Die nächsten beiden Abschnitte fassen die Ergebnisse der Interviews mit den beiden Stakeholdern zusammen.

6.2 Evaluation mit Kooperationspartner A

Die Analyse mit Kooperationspartner A wurde anhand der Ticket-Daten eines Redmine-Change-Request-Systems zu einem Software-Projekt durchgeführt. Die etwa 300 Tickets wurden über den *Redmine-Adapter* in das Werkzeug importiert. Die folgenden Szenarien zeigen Probleme im Softwareentwicklungs-Prozess auf, die mit Hilfe von *River* identifiziert wurden.

Szenario: Status-Fluss

Die Visualisierung des Ticket-Status-Flusses (Abbildung 6.1) zeigte keinen hauptsächlich verfolgten Workflow-Pfad. Mit Hilfe der auf die Eigenschaft "assigned to" eingestellten Histogramm-Detailansicht war erkennbar, dass die Status-Pfade stark mit dem Ticket-Bearbeiter korrelieren. Dies weist auf eine nicht standardisierte Nutzung des Change-Request-Systems hin, was der Interviewte bestätigte.

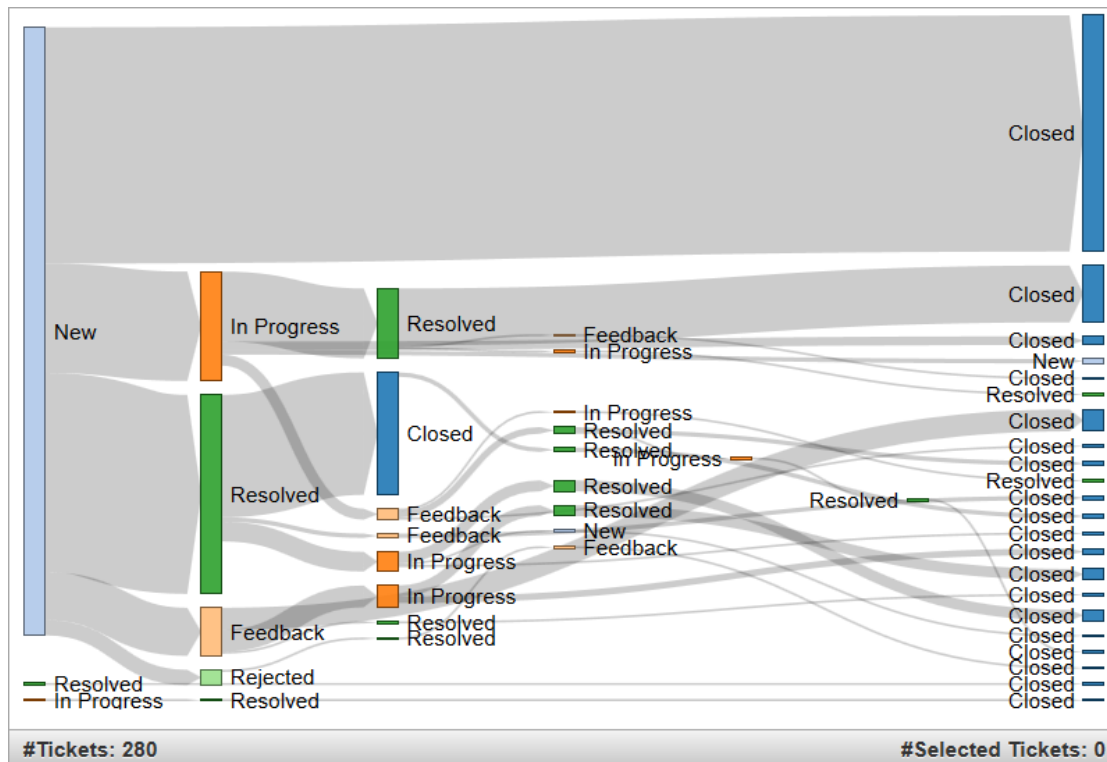


Abbildung 6.1: Evaluations-Szenario: Status-Fluss

Szenario: Ticket-Bearbeitungs-Fortschritt

In dieser Ansicht (Abbildung 6.2) ist die Nutzung der prozentual angegebenen Ticketeigenschaft “Ticket-Bearbeitungs-Fortschritt” (“done ratio”) visualisiert. Der Wertebereich zwischen 0% und 50% wird fast nicht genutzt, während der Wert 90% gehäuft auftritt. Die Eigenschaft scheint dazu genutzt zu werden, “fast erledigte” Tickets zu markieren, wird aber nicht zur Kennzeichnung von Tickets genutzt, deren Bearbeitung sich in einem frühen Stadium befindet. Möglicherweise ist der Bearbeitungs-Fortschritt zu einem frühen Zeitpunkt schwerer zu beurteilen als zu einem späten. Dieses Muster bei der Nutzung der Ticketeigenschaft “Ticket-Bearbeitungs-Fortschritt” war dem Interviewten zuvor nicht bewusst.

Szenario: Geschätzter Aufwand

In nur wenigen der etwa 300 Tickets wurde die Ticketeigenschaft “geschätzter Aufwand” verwendet (Abbildung 6.3). Dies weist auf eine nicht auf den Prozess zugeschnittene Ticketdefinition (z.B. eine nicht oder nur teilweise modifizierte Standard-Ticketdefinition) im Change-Request-System hin. In der Regel ist es jedoch sinnvoll, die Konfiguration des Change-Request-Systems an den Prozess anzupassen und nicht umgekehrt. Der Interviewte erläuterte, dass sich Qualitäts-Kontroll-Maßnahmen zum Projekt-Controlling, die auf geschätzten Aufwänden basieren, im Unternehmen derzeit im Aufbau befinden und schrittweise in den Projekten eingeführt werden.

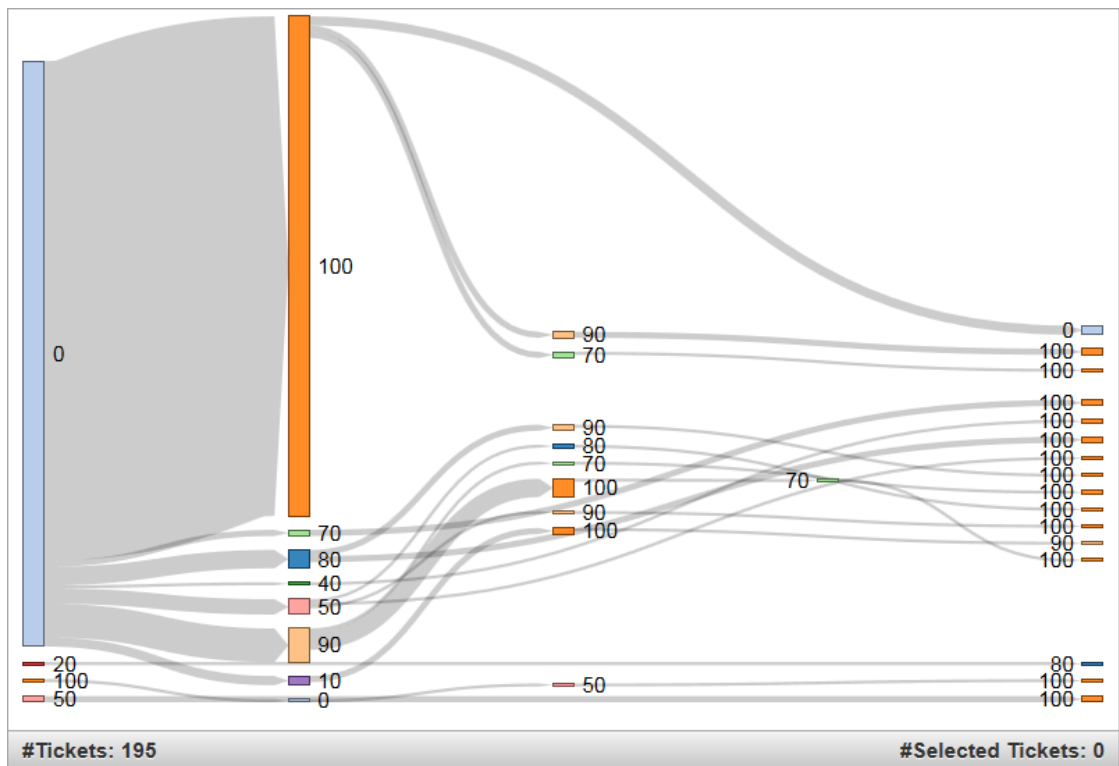


Abbildung 6.2: Evaluations-Szenario: Ticket-Bearbeitungs-Fortschritt

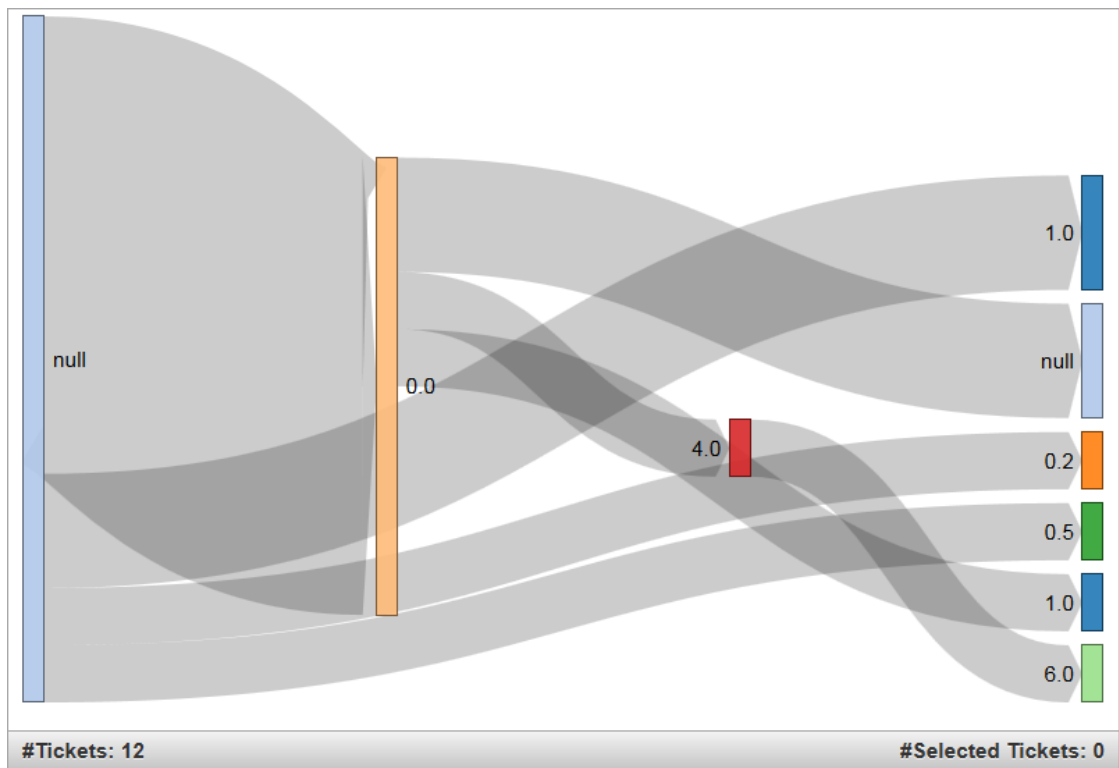


Abbildung 6.3: Evaluations-Szenario: Geschätzter Aufwand

Bewertung des Werkzeugs

- Nach einer kurzen Erläuterung der Semantik der Sankey-Diagramme zur Darstellung von Ticketeigenschafts-Flüssen war der Interviewte schnell in der Lage die Visualisierung zu verstehen und zu interpretieren.
- Aufgrund seiner Rolle im Unternehmen war der Interviewte mit den Tickets vertraut. Mehrfach fielen dem Interviewten nach der Auswahl der zu untersuchenden Ticketeigenschaft und der Definition eines Filters zunächst unerwartete Pfade im Sankey-Diagramm auf. Die zusätzliche Information der in der Detailansicht angezeigten Ticket-Themen half ihm bei der Bewertung dieses Pfades.
- Der Interviewte vermisste die konsolidierte Darstellung der aktuellen Ticketzustände im Sankey-Diagramm, wie sie im zweiten Prototypen verwendet wurde. Die Unterteilung der Tickets trotz identischem finalen Wert in der betrachteten Ticketeigenschaft sei nicht intuitiv verständlich.
- Es sollte die Möglichkeit geben, über Filterregeln den Betrachtungs-Zeitraum der zu analysierenden Tickets einzuschränken.
- Das Change-Request-System des Kooperationspartners A wird von (externen) Kunden verwendet um Fehler zu berichten, die dann von Mitarbeitern behoben werden. Daneben erstellen aber auch die Mitarbeiter selbst im Rahmen interner Prozesse Tickets. Der Interviewte schlug einen Filter vor, der auf der Gruppenzugehörigkeit (extern/intern) der Ticket-Ersteller basiert, um etwa ausschließlich die von Kunden erstellten Tickets zu analysieren.
- Die Oberfläche des Change-Request-System des Interviewten ist lokalisiert. Eine Ticketeigenschaft heißt z.B. “Geschätzter Aufwand”. Diese Lokalisierung ist in den importierten Daten nicht widergespiegelt. Daher findet der Benutzer die gleiche Ticketeigenschaft in *River* unter dem Begriff “estimated_hours” wieder. *River* sollte ebenfalls lokalisierte Ticketeigenschaftsnamen verwenden.
- Neu erstellte Tickets, die bezüglich der betrachteten Ticketeigenschaft nicht geändert werden, sollten ebenfalls dargestellt werden.

6.3 Evaluation mit Kooperationspartner B

Die folgenden Szenarien beschreiben die Erkenntnisse über die Softwareentwicklungsprozesse bei Kooperationspartner B, die durch die Analyse der Ticket-Daten und Ticket-Historien des dort genutzten Change-Request-Systems ClearQuest gewonnen wurden. Die Analysen basieren auf monatlichen Abzügen des Ticket-Status. Diese Abzüge wurden über die Datei-Import-Funktion in *River* importiert. Das Change-Request-System verwaltete etwa 30000 Tickets.

Szenario: Einschätzung der Zuständigkeit

Bei der Erfassung von Produktionsfehlern wird das zugehörige Ticket einem Teilprojekt zugeordnet, das für die Bearbeitung zuständig ist. Ist diese Zuordnung zunächst fehler-

haft, verzögert sich die Bearbeitung des Tickets ggf. erheblich und kann etwa die Einhaltung von Bearbeitungsfristen, die mit einem Kunden in einem Service Level Agreement getroffen wurden, gefährden. In der Visualisierung ist erkennbar, dass diese Zuordnung in aller Regel korrekt vorgenommen wird. Bei lediglich 26 von insgesamt ca. 30000 Tickets musste sie korrigiert werden (Abbildung 6.4). Es sind keine Hinweise auf eine Abweichung vom gewünschten Prozess erkennbar.

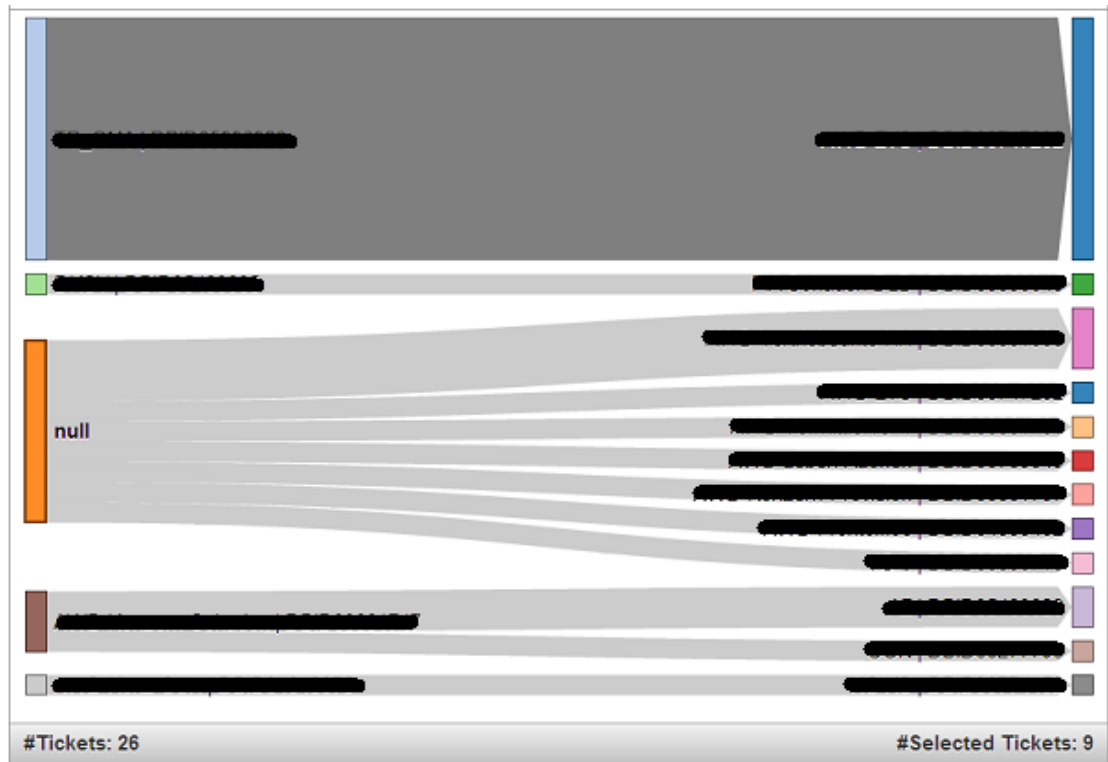


Abbildung 6.4: Evaluations-Szenario: Einschätzung der Zuständigkeit

Szenario: Häufigste Fehlerursache

In diesem Szenario zeigt die Visualisierung (Abbildung 6.5), dass viele Fehler ihren Ursprung in den Bereichen Code und Daten haben, den meisten Fehlern aber keine Ursache zugeordnet wird oder werden kann. Möglicherweise sind die verfügbaren Kategorien, die in diesem Feld eingetragen werden können, ungeeignet. Der Stakeholder sollte die Ursache für die ungewöhnliche Verwendung dieser Ticketeigenschaft weiter untersuchen.

Bewertung des Werkzeugs

- Analog zum Interview mit dem Experten des Kooperationspartners A war der Interviewte nach einer kurzen Einführung in die Bedienung des Werkzeugs und die Semantik der Sankey-Diagramm-Darstellung in der Lage, die Visualisierung des Prozesses zu verstehen und zu interpretieren.
- Er bewertete die Visualisierung nach der Einführung als intuitiv verständlich.

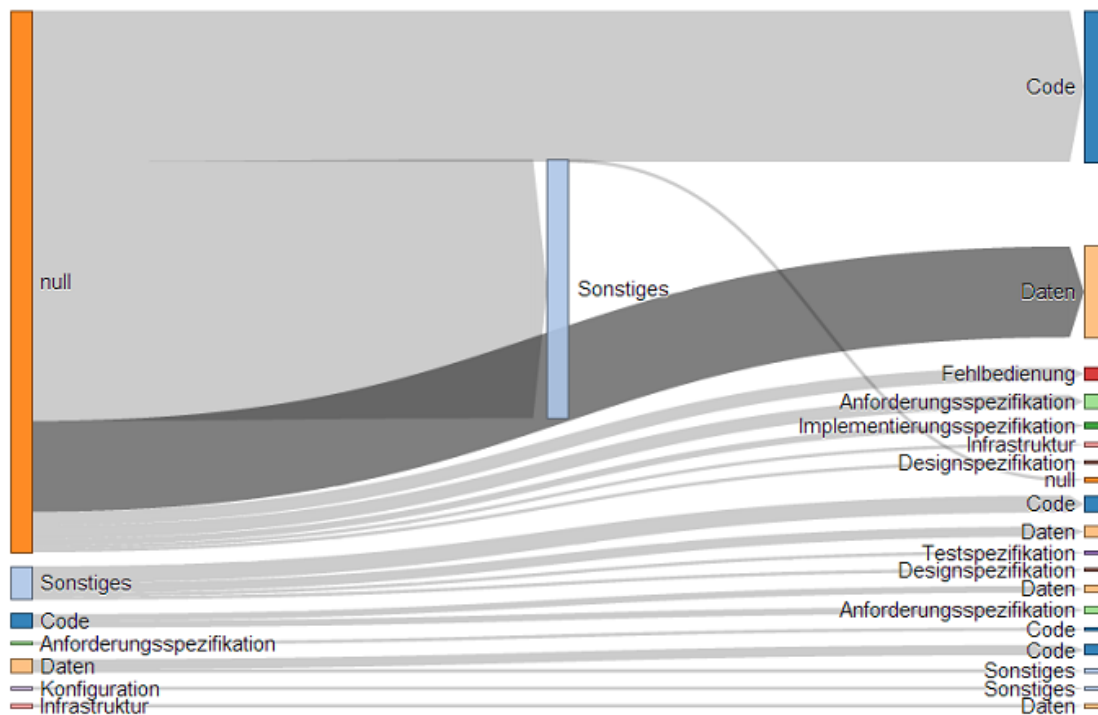


Abbildung 6.5: Evaluations-Szenario: Häufigste Fehlerursache

- Ebenso wie der Experte des Kooperationspartners A hatte der hier Interviewte den Wunsch nach einer konsolidierten Darstellung der Verteilung der Tickets am Ende des Betrachtungs-Zeitraums.
- Oft sei eine Abstraktion von den konkreten Ticketeigenschaftswerten erwünscht. Die möglichen Werte der Ticketeigenschaft "Status" lassen sich etwa in die beiden Kategorien "Offenes Ticket" und "Geschlossenes Ticket" einteilen. Das Werkzeug sollte Filter auf Basis dieser Einteilung erlauben und in der Visualisierung Tickets derselben Kategorie gemeinsam darstellen können.

6.4 Zusammenfassung

Anhand der gezeigten Szenarien waren die Evaluationsteilnehmer mit Hilfe des Werkzeugs in der Lage, bereits bekannte Prozessabweichungen (vgl. Szenario: kein Standardisierter Prozess) nachzuvollziehen. Ebenso gelang es, die erfolgreiche Umsetzung eines Prozessaspekts nachzuweisen (vgl. Szenario: Einschätzung der Zuständigkeit). Schließlich wurden auch neue Erkenntnisse über die Nutzung eines Change-Request-Systems gewonnen (vgl. Szenario: Ticket-Bearbeitungs-Fortschritt).

Die Werkzeug-Bewertung der beiden Evaluationsteilnehmer weist starke Übereinstimmung auf. Beide weisen darauf hin, dass die symmetrische Darstellung der Verteilung der Tickets gemäß der analysierten Ticketeigenschaft zu Beginn und zum Ende des Betrachtungs-Zeitraums wie sie im Prototyp verwendet wurde der im Werkzeug geänderten Sankey-Diagramm-Darstellung überlegen ist, insbesondere bezüglich der intuitiven

Verständlichkeit. Die Darstellung sollte daher in einer späteren Version von *River* entsprechend angepasst werden.

Die Forderung nach einem Filter für Ticket-Status-Kategorien bzw. Benutzer-Rollen entspricht verallgemeinert der Forderung, Metainformationen über die Semantik von Ticketeigenschaften in den Filtern des Werkzeugs zu verwenden. Diese Metainformationen stammen nicht mehr aus den Tickets selbst, sondern aus sonstigen im Change-Request-System vorhandenen Daten, wie etwa einer Workflow-Definition, die die Information enthält ob ein Ticket-Status in die Kategorie “offen” bzw. “geschlossen” fällt.

Liegen diese Metainformationen erst einmal dem Werkzeug vor, können sie, neben ihrer Verwendung in Filtern, auch zur Abstraktion der Visualisierung genutzt werden. Eine Darstellung des Ticket-Status, die lediglich die Kategorien “offen” und “geschlossen” differenziert, würde etwa den Anteil der Tickets, die in ihrem Lebenszyklus nach einer Schließung wieder eröffnet werden, sehr deutlich von den übrigen Tickets abheben.

Insgesamt gaben beide Evaluationsteilnehmer positive Bewertungen zu dem Werkzeug und seiner Visualisierung der Softwareentwicklungs-Prozesse durch Ticket-Eigenschafts-Änderungs-Flüsse ab. Die Visualisierung sei nach kurzer Erläuterung intuitiv verständlich und die Informationen in den Detailansichten (Ticket-Liste und Histogramm zu einer weiteren Ticketeigenschaft) seien hilfreich bei der Untersuchung einzelner Ticketeigenschafts-Pfade.