

Institut für Softwaretechnologie

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 87

Benutzerschnittstelle für einen Spreadsheet-Prüfstand

Jonas Scheurich

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr. Stefan Wagner
Betreuer/in:	Daniel Kulesz, M.Sc.
Beginn am:	29.07.2013
Beendet am:	28.01.2014
CR-Nummer:	H.4.1, D.2.5, H.5.2

Kurzfassung

Der Einsatz von Spreadsheets ist in vielen Branchen etabliert. Die verwendeten Spreadsheets sind oft fehlerhaft und können nur durch Experten geprüft werden. Am Institut für Softwaretechnologie wurde ein Framework zur statischen und dynamischen Prüfung von Spreadsheets entwickelt. Die Konfiguration dieses Frameworks ist über Konfigurationsdateien realisiert. Der praktische Einsatz des Frameworks in einem Spreadsheet-Entwicklungsprozess ist durch diese Einschränkung nicht praktikabel.

In dieser Arbeit wurde die Konfiguration des Frameworks zum Prüfen von Spreadsheets in eine populäre Spreadsheet-Ablaufumgebung integriert. Die Umsetzung realisiert ein Konzept der Szenario-basierten Konfiguration einer Spreadsheet-Prüfung. Die Konfiguration der Prüfung ist in der grafischen Oberfläche der Spreadsheet-Ablaufumgebung realisiert. Unabhängig von Experten können Benutzer Spreadsheet-Prüfungen konfigurieren, durchführen und wiederverwenden.

In der durchgeführten Evaluation wurde festgestellt, dass sich die grafische Oberfläche einer Spreadsheet-Ablaufumgebung gut eignet um Szenarien zur Konfiguration einer Spreadsheet-Prüfung zu erfassen. Die Evaluation zeigt, dass Benutzer mehr Informationen zum Ablauf der Spreadsheet-Prüfung benötigen.

Abstract

The use of spreadsheets are established in many industries. The Spreadsheets often contains errors and only can be tested by experts. The institute of softwaretechnology deployed a framework for dynamic and static tests of spreadsheets. The configuration of this framework is implemented by configuration files. It's not possible to use the framework in a practical spreadsheet deployment prozess.

This work integrates the configuration of the spreadsheet test framework into a popular spreadsheet execution enviroment. The implementation realizes a concept of a scenario based configuration of a spreadsheet test. The configuration of the spreadsheed test takes place in the spreadsheet view of a spreadsheet execution enviroment. Independent of qualified experts end-users have the ability to configurate, execute and reuse spreadsheet tests.

The completed evaluation shows that the graphical user interface of a spreadsheet execution enviroment is well suited to create szenarios for a spreadsheet test configuration. The evaluation also shows that end-users need more details of a spreadsheet test cycle.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Motivation	7
1.2. Problembeschreibung	7
1.3. Beitrag	8
1.4. Abgrenzung der Arbeit	8
1.5. Gliederung	9
2. Hintergrund	11
2.1. Spreadsheets	11
2.2. Fehler in Spreadsheets	13
2.3. Planung einer Spreadsheet-Erstellung	14
2.4. Prüfung von Spreadsheets durch Werkzeuge	14
3. Verwandte Arbeiten	15
3.1. WYSIWYT	15
3.2. Soap Opera Testing	17
3.3. Spreadsheet Inspection Framework	18
3.4. Spreadsheet Inspection Framework Excel Integration	22
3.5. Bewertung der Arbeiten	24
3.6. Problemanalyse und Vorüberlegungen	25
4. Konzeption der Konfiguration einer Spreadsheet Prüfung	27
4.1. Anforderungen	27
4.2. Lösungsansatz	27

4.3. Szenarien zur Schnittstellen-Abstraktion	28
4.4. Erfassung eines Szenarios	28
5. Umsetzung	31
5.1. Projektplanung	31
5.2. Architektur von Spreadsheet Inspection Framework Excel Integration (SIFEI) und Spreadsheet Inspection Framework (SIF)	33
5.3. Implementierung	33
5.4. Herausforderungen	39
6. Evaluation	41
6.1. Ziele	41
6.2. Versuchsaufbau	41
6.3. Durchführung	46
6.4. Ergebnisse	46
6.5. Schlussfolgerungen	47
7. Zusammenfassung und Ausblick	49
7.1. Zukünftige Arbeiten	50
A. Anhang	51
A.1. Evaluation	52
A.2. Inhalt der Daten-CD	62
Abkürzungsverzeichnis	63
Literaturverzeichnis	65

1. Einleitung

1.1. Motivation

Spreadsheets werden vielseitig in betrieblichen Prozessen und in Behörden zur Entscheidungsfindung eingesetzt [Pano8a]. Spreadsheets deren Formeln Fehler enthalten, berechnen oft falsche Ergebnisse. Falsche Ergebnisse eines Spreadsheets können wiederum zu falschen Entscheidungen führen. Die Folgen fehlerhaften Entscheidungen können wirtschaftliche Schäden in Millionenhöhe sein [eus]. Zum Beispiel musste ein Beratungshaus aufgrund eines fehlerhaften Spreadsheets eine Gewinnwarnung um £ 4,3 Mio. herausgeben. Das Spreadsheet eines externen Mathematikers hatte den Verlust eines Pensions-Fonds falsch berechnet [eus].

Ein Ansatz zur Reduzierung der Fehler in Spreadsheets sind Prüfungen durch Software-Werkzeuge [Kul13].

1.2. Problembeschreibung

Das Institut für Softwaretechnologie (ISTE) der Universität Stuttgart beschäftigt sich mit der Prüfung von Spreadsheets durch Software-Werkzeuge. In der Vergangenheit wurde zur Prüfung von Spreadsheets das Spreadsheet Inspection Framework (SIF) entwickelt. Das SIF prüft die Formeln eines Spreadsheets durch statische und dynamische Prüfungen. Nach der Konfiguration einer Prüfung führt SIF die Prüfung aus und fasst die Ergebnisse in einem Bericht zusammen. SIF ist nur über die Konsole und XML-Konfigurations-Dateien

1. Einleitung

(siehe 3.3.3) zu konfigurieren. Diese umständliche Konfiguration fällt den Benutzern schwer [Lem13, S. 40]. Ziel dieser Bachelor-Arbeit ist es die Konfiguration einer Prüfung durch SIF in eine Spreadsheet-Ablaufumgebung (siehe 2.1) zu integrieren.

1.3. Beitrag

Im Rahmen dieser Arbeit wurde die Konfiguration des Spreadsheet Inspection Framework in die Spreadsheet-Ablaufumgebung Microsoft Excel integriert. Die Spreadsheet-Erstellung und -Prüfung ist nun in einem einzigen Werkzeug möglich. Benutzern wird durch diese Arbeit die Möglichkeit geboten selbst Prüfungen zu konfigurieren und Durchzuführen. Die Abhängigkeit von Experten zur Spreadsheet-Prüfung konnte erheblich gesenkt werden.

1.4. Abgrenzung der Arbeit

Diese Arbeit beschäftigt sich nicht mit der Visualisierung von Verstößen, die das SIF zur Verfügung stellt. Die Visualisierung der Verstöße wurde in der Arbeit von Ehsan Doust umgesetzt [Dou13].

Die Ausführung der Prüfung eines Spreadsheets wird in dieser Arbeit nicht bearbeitet. Die Prüfung von Spreadsheets wurde in den Arbeiten [Zit12, Lem13] entwickelt.



Abbildung 1.1.: Einordnung der Arbeit in den Ablauf einer Spreadsheet-Prüfung durch SIFEI. Der schwarz markierte Prozess-Abschnitt wurde in dieser Arbeit entwickelt.

1.5. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Hintergrund beschreibt den fachlichen Hintergrund der Arbeit.

Kapitel 3 – Verwandte Arbeiten erläutert ähnliche Arbeiten.

Kapitel 4 – Konzeption der Konfiguration einer Spreadsheet Prüfung stellt den entworfenen Lösungsansatz vor.

Kapitel 5 – Umsetzung beschreibt die Umsetzung des Lösungsansatzes.

Kapitel 6 – Evaluation fasst die Realisierung und die Ergebnisse der Evaluation zusammen.

Kapitel 7 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Möglichkeiten zur Weiterentwicklung vor.

2. Hintergrund

In diesem Kapitel werden die Hintergründe dieser Arbeit besprochen. Grundlegende Begriffe zu Spreadsheets werden definiert, Fehler in Spreadsheets und deren Folgen beschrieben. Aktuelle Prüfverfahren werden diskutiert und die Werkzeuge beschrieben, auf denen diese Arbeit aufbaut.

2.1. Spreadsheets

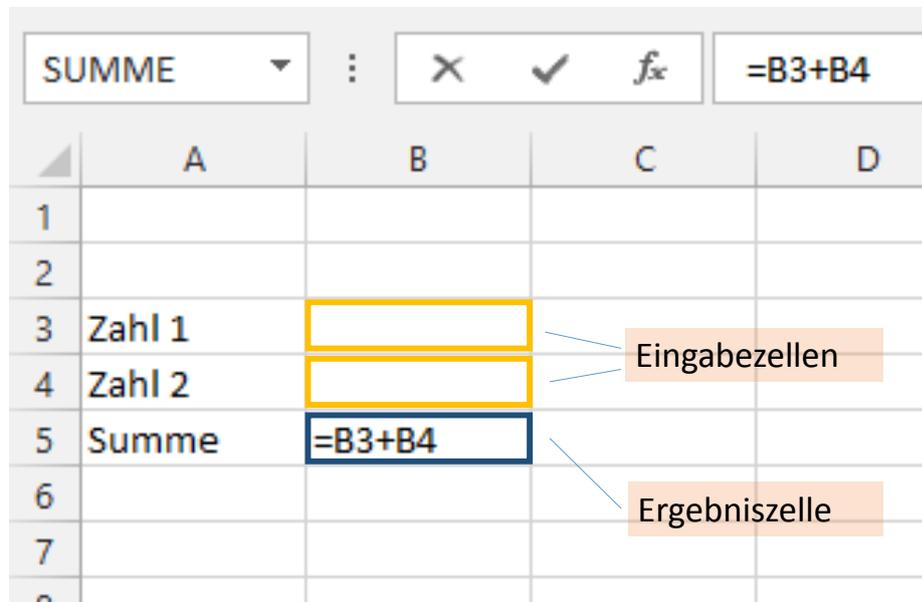
Ein Spreadsheet ist ein Programm, das Berechnungen durchführt und wird durch Spreadsheet-Ablaufumgebungen in Tabellenform dargestellt. Die auf dem Markt vornehmlich verwendeten Spreadsheet-Ablaufumgebungen sind Microsoft Office Excel, Open Office Calc Distributionen oder Apple iWorks Numbers [Tea10]. Ein Spreadsheet kann in mehrere Worksheets gegliedert sein. In ein Spreadsheet werden über Eingabezellen Werte erfasst, aus denen Formeln die Resultate der Ergebnis- oder Zwischenergebniszellen berechnen. Das Resultat einer Formel wird in der zugehörigen Ergebnis- oder Zwischenergebniszelle dargestellt. Nach der Änderung einer Zelle werden die Resultate der Ergebnis- und Zwischenergebniszellen durch die Spreadsheet-Ablaufumgebung automatisch neu berechnet.

Eingabezelle - Eine Zelle in einem Spreadsheet, die konstante Werte enthält und von Formeln in anderen Zellen zur Berechnung genutzt wird. Zellen die Text enthalten, der zur Erklärung oder Bezeichnung anderer Zellen dient, sind keine Eingabezellen.

Zwischenergebniszelle - Eine Zelle in einem Spreadsheet, die eine Formel enthält und von Formeln in anderen Zellen zur Berechnung genutzt wird.

2. Hintergrund

Ergebniszelle - Eine Zelle in einem Spreadsheet, die eine Formel enthält und von keiner Formel in anderen Zellen zur Berechnung genutzt wird.



The image shows a spreadsheet interface. At the top, there is a formula bar with a dropdown menu set to 'SUMME', a colon separator, and buttons for 'X', '✓', and 'fx'. The formula bar contains the text '=B3+B4'. Below this is a grid with columns labeled A, B, C, and D, and rows numbered 1 through 8. In row 3, column B, there is a yellow-bordered cell containing 'Zahl 1'. In row 4, column B, there is a yellow-bordered cell containing 'Zahl 2'. In row 5, column B, there is a blue-bordered cell containing '=B3+B4'. Two callout boxes with arrows point to the yellow-bordered cells, labeled 'Eingabezellen'. One callout box with an arrow points to the blue-bordered cell, labeled 'Ergebniszelle'.

	A	B	C	D
1				
2				
3	Zahl 1			
4	Zahl 2			
5	Summe	=B3+B4		
6				
7				
8				

Abbildung 2.1.: Beispiel Eingabezellen und Ergebniszellen

Formeln berechnen Resultate aus Operatoren und Operanden. Operanden bestehen aus konstanten Werten, Werten aus referenzierten Zellen und Funktionen der Spreadsheet-Ablaufumgebung. Funktionen einer Spreadsheet-Ablaufumgebung besitzen mehrere Parameter, die Konstanten, Verweise auf eine Zelle, oder Verweise auf einen Zell-Bereich enthalten können. Konstante Werte sind Zahlen, Zeitangaben oder Text. Zellen werden durch Zell-Adressen referenziert.

Zell-Adresse - Eine Zelle-Adresse identifiziert eine Zelle einem Spreadsheet. In dieser Arbeit werden Zell-Adressen in der A1-Notation verwendet. Diese besteht aus dem Worksheet-Namen, dem Spaltenbuchstaben und der Zeilennummer: [<Worksheet-Name>!]<Spaltenbuchstabe><Zeilennummer>. Eine Zelle im Worksheet "Tabelle1", in der zweiten Spalte und der dritten Zeile wird mit dem Worksheet-Namen, dem

Spaltenbuchstaben und der Zeilennummer identifiziert. Dieses Beispiel wird folgendermaßen dargestellt: `Tabelle1!B3`.

Zell-Bereich - Ein Zell-Bereich wird durch zwei, durch einen Doppelpunkt getrennte, Zell-Adressen identifiziert: `<Zell-Adresse>:<Zell-Adresse>` Zum Beispiel wird der Bereich zwischen den Zellen B1 und B3 folgendermaßen dargestellt: `B1:B3`.

2.2. Fehler in Spreadsheets

Fehler teilen Liggesmeyer und die IEEE [Lig09, IEE10] in *Fehlverhalten*, *Fehler* und *Ausfälle* ein. *Fehlverhalten* sind Handlungen von Menschen, die ein falsches Ergebnis zur Folge haben können. Ein *Fehler* ist das Auftreten eines Software-Fehlers auf Grund eines *Fehlverhaltens*. Ein *Ausfall* tritt auf Grund eines *Fehlers* auf. Ein Programm in dem ein *Ausfall* auftritt ist nicht mehr in der Lage, innerhalb seiner Spezifikation die implementierte Funktionalität bereit zu stellen.

Fehler in Spreadsheets lassen sich in quantitative und qualitative Fehler einteilen. Quantitative Fehler sind Fehler, die ein falsches Ergebnis zu Folge haben [Pano8a]. Powel et al. schlagen vor die quantitativen Fehler nochmals zu unterteilen:

Mechanisches Fehlverhalten - Hervorgerufen durch falsche Eingabewerte bei der Spreadsheet-Anwendung.

Logischer Fehler - Hervorgerufen durch fehlerhafte Formeln oder Funktionen.

Unterlassendes Fehlverhalten - Grundlage für ein Unterlassendes Fehlverhalten ist ein falsches mentales Modell der Situation.

Fehler treten bei etwa 20 % bis 40 % der in Experimenten erstellten Spreadsheets auf [Pano8a]. Diese Werte entsprechen allerdings keiner oberen Grenze. Panko berichtet von kommerziellen Spreadsheet-Audits, in denen in allen eingereichten Spreadsheets Fehler gefunden wurden. In den Studien ist erkennbar, dass die Vorbildung der Probanden die Spreadsheets erstellen, sich nicht in der Cell Error Rate (CER) niederschlägt [Pano8a, Pano8c].

2.3. Planung einer Spreadsheet-Erstellung

Die Prozessmodelle der Softwaretechnik [Lud13] legen viel Wert auf Planung, Entwurf und Dokumentation. Spreadsheets werden dagegen ohne vorherigen Entwurf oder Planung erstellt [Pano8a]. Analog zu dem in der Softwaretechnik als „Code-and-fix“ bekannten Vorgehen, ist die Fehlerwahrscheinlichkeit in ungeplanten Spreadsheets höher [SGP13]. Die Bereitschaft des Spreadsheet-Erstellers, das Spreadsheet auf Fehler zu Prüfen oder zu Testen sinkt, desto mehr die eigenen Fähigkeiten überschätzt werden [Pano8b].

2.4. Prüfung von Spreadsheets durch Werkzeuge

Auf dem Markt sind einige Werkzeuge zur Prüfung von Spreadsheets erhältlich [BNV12]. Die Werkzeuge unterscheiden sich in ihrer Zielgruppe und der Bereitstellung der Ergebnisse. Berberich et al. unterscheiden Spreadsheet-Prüfwerkzeuge in vollautomatische Werkzeuge und Review-unterstützende Werkzeuge [BNV12, S. 17]. Ein Teil der Werkzeuge ist in eine Spreadsheet-Ablaufumgebung integriert oder werden als Software as a Service (SaaS) angeboten. Ein Teil auf dem Markt erhältlichen Werkzeuge visualisieren das Spreadsheet-Programm in Berechnungsbäumen und tragen so zur besseren Verständlichkeit des Spreadsheets bei. Andere Werkzeuge ermöglichen die Markierung von fehlerfreien Zellen. Berechnungsbäume sind Graphen, deren Kanten die Assoziationen zwischen den Operanden der Formel einer Zelle darstellen.

Andere Werkzeuge suchen nach leeren Zellen, Abweichungen von wiederholenden Mustern, Referenzen zu leeren Zellen oder Abweichungen von wiederholendem Text [CFRS12, CFM⁺12]. Weitere Werkzeuge visualisieren die Formeln in Graphen [BNV12]. Die Werkzeuge können in Review- oder Audit-Prozessen eingesetzt werden, ermöglichen aber keine dynamische Prüfung der Formeln eines Spreadsheets.

3. Verwandte Arbeiten

Zur Durchführung einer Spreadsheet-Prüfung mit Software-Werkzeugen werden verschiedene Techniken eingesetzt. Dieses Kapitel erläutert Arbeiten die Techniken beschreiben, die sich zur Spreadsheet-Prüfung mit Werkzeugen eignen können.

3.1. WYSIWYT

Gregg Rothermel et al. beschreiben in ihrer Veröffentlichung *What you see is what you test: a methodology for testing form-based visual programs* [Rot98] eine Methodik zum Testen von Formular-basierter Software. Die Prüfung der Zellen wird durch den Spreadsheet-Ersteller vorgenommen, der auf Grund seiner Erfahrung das Ergebnis einer Formel überprüft und seine Einschätzung in dem Spreadsheet dokumentiert. Beispiel der Veröffentlichung ist eine Implementierung einer Software zur Berechnung verschiedener Darstellungen einer Uhrzeit. Umgesetzt wurde das Beispiel in der Visual Programming Language (VPL) *Forms/3*. Die Methodik visualisiert die Prüfvollständigkeit eines Spreadsheets nach einer Änderung. Dazu erhält jede Zelle ein Statusfeld, das nach der initialen Spreadsheet-Erstellung zunächst leer bleibt. Die Arbeit gliedert die Änderungen des Statusfeldes in fünf Aufgaben:

Task 1: Erkennen von Eingabezellen - Die Berechnungen des Spreadsheet-Programms werden in einem Berechnungsbaum dargestellt. Die Zellen mit konstanten Einträgen, die als Eingabezelle verwendet werden, werden identifiziert.

Task 2: Aufzeichnen des Auswertungspfades - Wird das Spreadsheet-Programm ausgeführt, werden die Resultate der Formeln berechnet. Während dieser Berechnung wird

in der zweiten Aufgabe der Ausführungspfad für diese Zelle aufgezeichnet und im Datenmodell hinterlegt.

Task 3: Ergebnis für gültig markieren - Nach der erstmaligen Erfassung von Eingabewerten durch die Eingabezellen, wird das Spradsheet ausgeführt. Der Spreadsheet-Ersteller prüft nun die Resultate der Zwischenergebnis- und Ergebniszellen. Nimmt der Spradsheet-Ersteller an, dass das berechnete Resultat richtig ist, wird die Zelle vom Spreadsheet-Ersteller markiert. Die Markierung erfolgt im Statusfeld der Zelle. Ändert sich der Wert einer Eingabezelle, werden von allen Zellen, zu denen ein Pfad im Berechnungsbaum besteht, die Statusfelder auf ein Fragezeichen gesetzt. Dem Spreadsheet-Ersteller wird durch das Fragezeichen symbolisiert, dass die Prüfung dieser Zellen erneut erfolgen muss. Die Rahmen der Zwischenergebnis- und Ergebniszellen zeigen den Prüffortschritt der Operanden der Zelle an. Die Rahmenfarbe wird von rot (0 % geprüft) auf blau (100 % geprüft) abgebildet. Ein blauer Rahmen einer Zelle symbolisiert, dass die Berechnung des Resultates vollständig geprüft wurde.

Task 4: Änderungen an Formeln behandeln - Die vierte Aufgabe behandelt die Änderungen von Zellen mit Formeln. Wird eine Formel zu einer Konstante geändert, wird das Statusfeld auf ein Fragezeichen gesetzt. Wird eine Formel gelöscht, wird dies wie einer Änderung der Formel zu BLANK behandelt und das Statusfeld ebenfalls auf ein Fragezeichen gesetzt. Dieser Vorgang wird rekursiv wiederholt bis alle Zellen deren Operanden mit einem Fragezeichen markiert wurden ebenfalls mit einem Fragezeichen markiert sind.

Task 5: Stapelverarbeitung - Die fünfte Aufgabe beinhaltet besondere Algorithmen zur Bearbeitung von Änderungen die mehrere Zellen betreffen. Diese Methoden werden in der Veröffentlichung nicht näher erläutert.

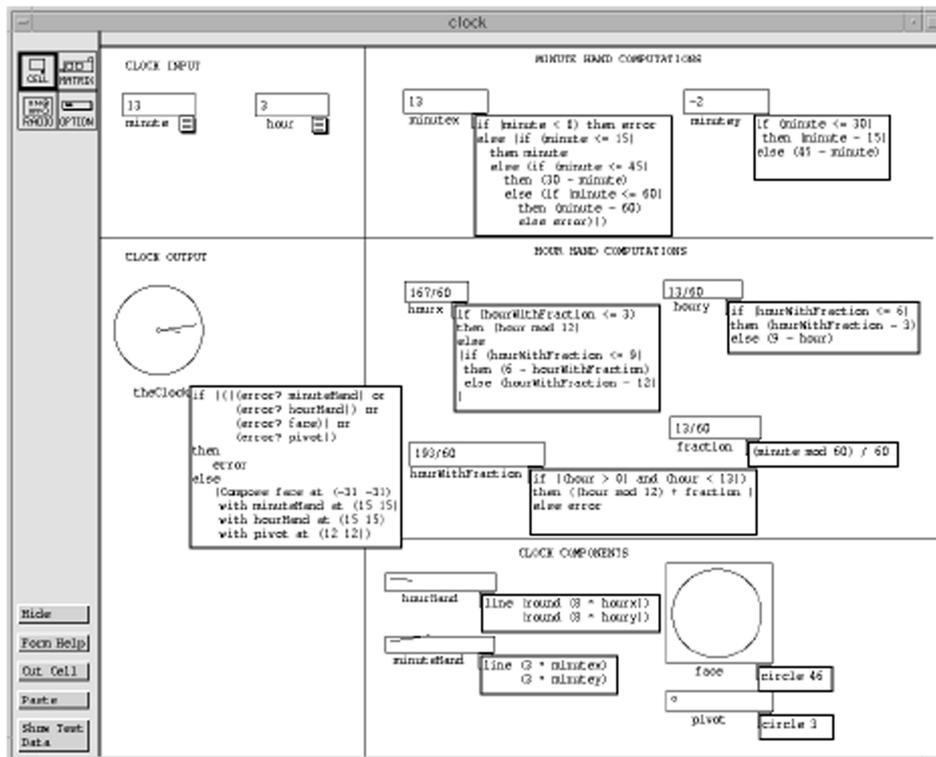


Abbildung 3.1.: Implementierungsbeispiel der Veröffentlichung WYSIWYT [Rot98, S. 200]

3.2. Soap Opera Testing

Hans Buwalda beschreibt in seiner Veröffentlichung *Soap Opera Testing* eine Methode zur kreativen Testfallerstellung [Buwo4]. Software-Ingenieure und Informatiker testen Software-Systeme oft in kleinen Komponenten-Tests, die anschließend zu großen Systemtests zusammengefasst werden. Die Benutzer der zu testenden Software haben dagegen ein größeres praktisches Wissen über den zukünftigen Einsatz der Software. Die Zusammenführung dieser beiden Know-How-Bereichen ergibt der Ansatz des *Soap Opera Testing*. Vor der Erstellung eines Szenario-basierten Tests wird eine fiktive Anwendungsumgebung erstellt. In dieser Anwendungsumgebung möchte eine Person eine Aufgabe mit dem zu testenden System lösen (z. B. eine Person die einen Versicherungsfall mit ihrer Versicherung abwickelt). Aus der Anwendungsumgebung wird ein erstes Szenario abgeleitet. Dieses Szenario ist die

Einsatzumgebung mit der Bearbeitungsschritten der zu lösenden Aufgabe. Das Szenario wird um Ausnahme-Situationen erweitert, so dass sich eine Verkettung von „aggressiven“ Belastungen für das System ergibt. Aus dem erweiterten Szenario wird abschließend der Testfall entwickelt. Vorteil dieses Vorgehens ist, dass die Beziehung zwischen den Teilen eines Tests und einer Menge von Funktionalitäten eines Programms durch *Soap Opera Tests* nun keine 1:1-Beziehung mehr ist, sondern eine m:n-Beziehung darstellt.

3.3. Spreadsheet Inspection Framework

Das SIF ist ein Werkzeug zur Prüfung von Spreadsheets anhand statischer und dynamischer Prüfungen. SIF wurde in Java implementiert und benutzt das Framework Apache POI. Über eine Socket-Schnittstelle wird SIF konfiguriert [Dou13]. Dazu dient das von [Lem13] erstellte SpRuDeL-XML-Schema.

3.3.1. Statische Prüfungen

Statische Prüfungen sind Code-Analysen aus der Softwaretechnik. Ein Programm-Code wird ohne ausgeführt zu werden, auf verschiedene Qualitätsmerkmale überprüft. Dabei wird nur der Text des Source Codes untersucht. Zum Beispiel wird nach Code-Duplikaten oder Verstöße gegen die Programmier-Richtlinien gesucht.

Das Konzept der statischen Prüfungen im SIF wurden von Sebastian Zitzelsberger im Rahmen einer Diplomarbeit [Zit12] konzipiert und umgesetzt. Statischen Prüfungen ermöglichen die Erfassung von quantitativen Merkmalen. Im SIF sind die statischen Prüfungen *Konstanten in Formeln*, *Leserichtung* und *Formelkomplexität* implementiert. Das Spreadsheet Rule Definition Language (SpRuDeL)-XML-Schema (siehe 3.3.3) unterstützt zur Zeit nicht die Konfiguration der statischen Prüfungen von SIF.

3.3.2. Dynamische Prüfungen

Eine Dynamische Prüfung ist eine Methodik aus der Softwaretechnik zum Testen von Software. Im Rahmen einer dynamischen Prüfungen wird der Programm-Code eines Programms ausgeführt. Während der Ausführung werden Testeingaben vorgenommen und die Resultate der Ausführung mit zuvor definierten Soll-Resultaten verglichen.

Die dynamischen Prüfungen im SIF wurden von Manuel Lemcke im Rahmen einer Diplomarbeit konzipiert und im SIF implementiert [Lem13]. Wie in der dynamischen Prüfung von Software wird im Rahmen einer dynamischen Prüfung eines Spreadsheets, das Spreadsheet mit Testeingaben befüllt und anschließend ausgeführt. Die Ausführung eines Spreadsheets ist die Berechnung des Spreadsheet-Programms. SIF stellt eine Architektur zur Verfügung, in der verschiedene dynamische Prüfungen integriert werden können. Eine dynamische Prüfung im SIF wird durch eine Menge von Regeln beschrieben (siehe 3.3.3). Zur Prüfung führt SIF eine temporäre Kopie des Spreadsheet-Programms durch Apache POI mit den Testeingaben der Regeln aus. Im Rahmen der Ausführung des Spreadsheets werden Invarianten und Nachbedingungen mit den Resultaten des Spreadsheets verglichen.

Invariante - Eine Invariante besteht aus einer Zell-Adresse und einem Soll-Resultat. Das Soll-Resultat wird während der Ausführung der dynamischen Prüfung durch das SIF zu mehreren Zeitpunkten mit den Ergebnis der referenzierten Zelle verglichen. (vgl. Abb. 3.2).

Nachbedingung - Eine Nachbedingung besteht aus einer einer Zell-Adresse und einem Soll-Resultat. Das Soll-Resultat wird als letzter Schritt der der dynamischen Prüfung durch das SIF mit den Ergebnis der referenzierten Zelle verglichen. (vgl. Abb. 3.2).

3. Verwandte Arbeiten

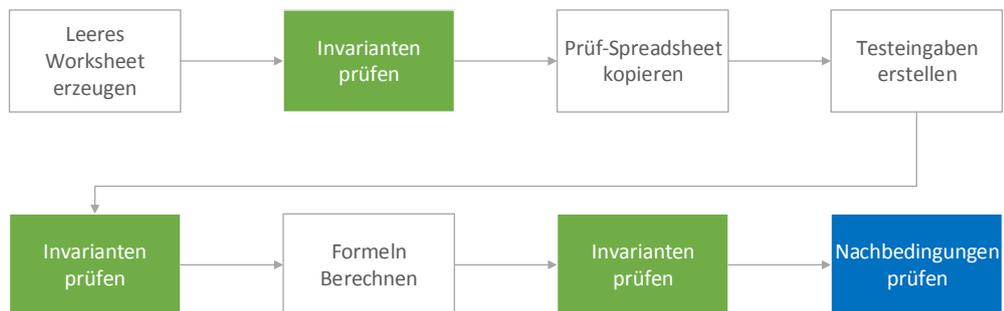


Abbildung 3.2.: Zeitlicher Ablauf der Prüfung von Invarianten und Nachbedingungen nach Lemcke [Lem13, S. 27]

3.3.3. Regeln

Zur Durchführung einer dynamische Prüfung mit SIF müssen Regeln in einer XML-Konfigurations-Datei definiert werden. Diese Datei muss dem SpRuDeL-XML-Schema entsprechen. Eine Regel besteht aus einer Menge von Testeingaben und Invarianten oder Nachbedingungen (vgl. Abb. 3.3). Invarianten und Nachbedingungen beschreiben Soll-Resultate. Invarianten und Nachbedingungen bestehen entweder aus einer *BinaryCondition*, einer *TernaryCondition*, oder einer *ElementCountCondition*:

BinaryCondition - Die Bedingung vergleicht den Wert der zu prüfenden Zelle mit einem Soll-Wert.

TernaryCondition - Die Bedingung vergleicht den Wert der zu prüfenden Zelle mit einem Soll-Intervall.

ElementCountCondition - Die Bedingung zählt die Vorkommen eines Zelltypen.

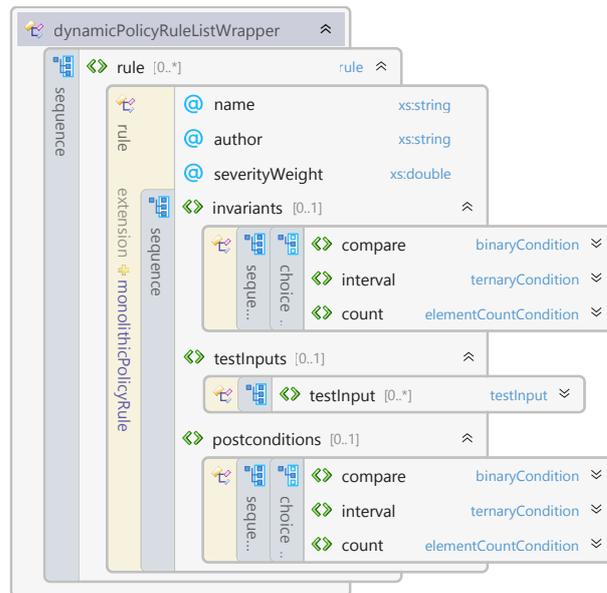


Abbildung 3.3.: Struktur einer Regel nach dem SpRuDeL-XML-Schema von Lemke [Lem13]

3.3.4. Ergebnisse der Prüfung

Die Ergebnisse einer Prüfung stellt SIF in einer HTML-Datei oder im XML-Format über eine Socket-Schnittstelle zur Verfügung. Zu jedem Verstoß werden Hintergrundinformationen, eine Beschreibung der Prüfregel und Lösungsvorschläge beschrieben.

3.3.5. Einschränkungen

Die dynamischen Prüfungen des SIF sind durch das Framework Apache POI eingeschränkt. Es ist nicht möglich Makros eines Excel-Spreadsheets in die dynamische Prüfung mit einzubeziehen. Die Konfiguration einer Prüfung ist nur durch XML-Dateien möglich, die vom Spreadsheet-Prüfer erstellt werden müssen.

3.4. Spreadsheet Inspection Framework Excel Integration

Die *Spreadsheet Inspection Framework Excel Integration* stellt Spreadsheet-Prüfung des SIF in der Spreadsheet-Ablaufumgebung Microsoft Excel zur Verfügung. Die Verstöße werden durch die *SIFEI-Visualisierung* direkt in den betroffenen Zellen dargestellt und zusammengefasst in einer Seitenleiste aufgelistet.

3.4.1. Konzept einer Prüfung durch SIFEI

Nachdem eine Konfigurationsdatei im SpRuDeL-XML-Schema erstellt wurde, wird die Prüfung durch das SIF vorgenommen. Die vom SIF ermittelten Verstöße werden anschließend im Spreadsheet visualisiert und können vom Nutzer zur Verbesserung des Spreadsheets genutzt werden.



Abbildung 3.4.: Ablauf eines SIFEI-Prüfvorgangs. Die erstellten Regeln werden geprüft und anschließend zur Analyse visualisiert.

3.4.2. Schnittstelle zwischen SIF und SIFEI

Im Rahmen der Entwicklung der ersten Ausbaustufe der SIFEI durch Ehssan Doust wurde in SIF eine Socket-Schnittstelle implementiert. Zur Vorbereitung eines Prüfvorgangs wird das zu prüfende Spreadsheet und die Konfigurationsdatei zur dynamischen Prüfung über die Socket-Verbindung an das SIF gesendet. Nach Abschluss des Prüfvorgangs wird der vom SIF erstellte Bericht im XML-Format an SIFEI zurückgesendet.

3.4.3. Visualisierung der Befunde des SIF

Die vom SIF ermittelten Verstöße werden in SIFEI durch einen Icon in der fehlerhaften Zelle dargestellt. Im Kontextmenü des Icons und in der Seitenleiste werden ausführliche Informationen und Verbesserungsvorschläge zu dem Befund angezeigt. Befunde können als False-Positive gekennzeichnet werden.

The image shows an Excel spreadsheet with a sidebar titled 'Findings'. The spreadsheet is titled 'Afög-Rechner für vorzeitige Rückzahlung (vereinfacht)'. The sidebar lists several violations:

- Bachelor + Master** (1800): 'Executes' the spreadsheet and checks for several possible conditions. **Violations (1)**
- Numeric-Cell [Formula]: 5964.03316600001** (1800): The found value was not in the interval [4886.0;4946.0]. **Violations (1)**
- Policy Rule: No Constants In Formulas** (280): Checks whether formulas contain constant values. **Violations (5)**
- Formula[=(100-(B30*0.00285))* (1/100)*B30)-(0*B9)]** (100): The following constants were found: Integer, Integer (2x), Integer, Numeric.
- Formula[=IF (B8=105,6,IF (B8=61.36,4,IF (B8=25.56,2,IF (B8=40.9,2,-1)))]** (100): The following constants were found: Numeric,

A context menu is open over a cell, showing details for the 'Numeric-Cell' violation:

- Numeric-Cell [Formula]: 5900.88110350001**
- The found value was not in the interval [6185.0;6215.0]
- Possible Solution**: Check the formulae in the causing cell or region.
- Background**: A testscenario which is built by userdefined conditions. While the process of checking the spreadsheet is executed, that is formulae of the spreadsheet are evaluated.
- [False Positive](#)

Abbildung 3.5.: SIFEI-Visualisierung durch Icons in der Zelle, sowie einer Zusammenfassung in der Seitenleiste.

3.4.4. Einschränkungen

SIFEI im Ausbauzustand von Ehssan Doust [Dou13] unterstützt ebenfalls keine Konfiguration der Prüfung durch Regeln. Die Regeln müssen vom Benutzer im SpRuDeL-XML-Schema erstellt werden [Lem13].

3.4.5. Historie von SIF und SIFEI

SIF und SIFEI wurde in mehreren Ausbaustufen entwickelt und angepasst.

Zeitraum	Bearbeiter	Änderung	Komponente
Juli 2011 - Jan. 2012	S. Zitzelsberger	Architektur	SIF
		Statische Prüfungen	SIF
Aug. 2012 - Febr. 2013	M. Lemcke	Dynamische Prüfungen	SIF
		SpRuDeL-XML-Schema	SIF
Juni - Sept. 2013	E. Doust	Socket-Schnittstelle zu SIFEI	SIF
		Socket-Schnittstelle zu SIF	SIFEI
		<i>SIFEI-Visualisierung</i>	SIFEI
Aug. - Okt. 2013	J. Scheurich	Integration	SIF
		<i>SIFEI-Szenarienerfassung</i>	SIFEI
		Integration	SIF

Tabelle 3.1.: Historie von SIF und SIFEI

3.5. Bewertung der Arbeiten

What You See Is What You Test (WYSIWYT) ist vermutlich gut geeignet die Prüfvollständigkeit darzustellen. Die Integration in eine Spreadsheet-Ablaufumgebung ermöglicht eine einfache Nutzung durch Benutzer. Herausfordernd ist es die Berechnung des

Spreadsheet-Programms korrekt in einem Berechnungsbaum darzustellen, da Spreadsheet-Ablaufumgebungen eine Vielzahl von aufwändigen Funktionen zur Verfügung stellen.

Sopa Opera Testing ist ein vielversprechender Ansatz zur Erstellung von Testfällen für Software. Szenarien können durch Benutzer ohne Kenntnisse des Software-Testens erstellt werden. Durch Soap Opera Testing ist vermutlich nicht gewährleistet, dass eine volle Testüberdeckung erreicht wird. Es ist nicht sichergestellt, dass alle Systemgrenzen getestet werden. Dieser Ansatz lässt sich evtl. auch auf die Konfiguration von Spreadsheet-Prüfungen anwenden.

SIF ermöglicht die dynamische Prüfung der Formeln eines Spreadsheets. Die Darstellung der Verstöße einer Prüfung durch SIFEI in einer Spreadsheet-Ablaufumgebung ermöglicht es Benutzern Fehler in Spreadsheets zu finden. Die Konfiguration von SIF und auch SIFEI durch eine Konfigurations-Datei setzen Kenntnisse in XML und des SpRuDeL-XML Schemas voraus. Diese Voraussetzungen schließt viele Benutzer aus.

3.6. Problemanalyse und Vorüberlegungen

Die in diesem Kapitel betrachtete Literatur beschreibt gute Vorschläge zur Prüfung von Spreadsheets. Allerdings weisen die vorgestellten Lösungen individuelle Nachteile auf. Die Konfiguration einer Spreadsheet-Prüfung durch SIF oder SIFEI ist sehr kompliziert und unübersichtlich. Die fehlende Integration in eine Spreadsheet-Ablaufumgebung erschwert einen einfachen Prozess zur Erstellung und Prüfung von Spreadsheets. Die Folge ist, dass eine Prüfung durch SIFEI unnötig aufwändig ist.

Benötigt wird eine Integration der Konfiguration einer Spreadsheet-Prüfung durch SIF in eine Spreadsheet-Ablaufumgebung. Die Konfiguration sollte an die Durchführung einer Spreadsheet-Nutzung angelehnt sein und keinerlei Vorkenntnisse zu Technologien (z.B. XML) oder weiterer Werkzeuge (z.B. SIF) voraussetzen.

4. Konzeption der Konfiguration einer Spreadsheet Prüfung

Aus der Problemanalyse und den Vorüberlegungen wurden Anforderungen und ein Lösungskonzept entwickelt, die in diesem Kapitel beschrieben werden.

4.1. Anforderungen

Aus den Vorüberlegungen wurde in Zusammenarbeit mit dem Betreuer eine Reihe von Anforderungen an eine Lösung abgeleitet.

- Integration der Lösung in SIFEI
- Konfiguration von Regeln und Testfällen für eine Prüfung durch SIF
- Entwicklung einer geeigneten Metapher zur Erfassung der Regeln.
- Aktualisierung der Regeln und Testfälle nach Spreadsheet-Änderungen
- Speichern der Regeln und Testfälle in der Datei eines Spreadsheets.

4.2. Lösungsansatz

Die Konfiguration einer Prüfung durch SIF findet in der grafischen Oberfläche der Spreadsheet-Ablaufumgebung statt. Der Benutzer erstellt mehrere Szenarien in den Zellen des Spreadsheets, die einer möglichen Nutzung des zu prüfenden Spreadsheets entsprechen.

Aus je einem Szenario wird anschließend eine Regel und Testfälle im SpRuDeL-XML-Schema generiert. Die Konfiguration einer Prüfung durch SIF soll an die Benutzung eines Spreadsheets angelehnt sein. Dieses Konzept wird im folgenden als *Szenarienerfassung* bezeichnet.

4.3. Szenarien zur Schnittstellen-Abstraktion

Das Datenmodell der Schnittstelle zu SIF ist sehr technisch und orientiert sich im Aufbau und der Terminologie sehr an den Anforderungen einer Prüfung durch SIF. Der Benutzer ist nicht mit den technischen Grundlagen einer Spreadsheet-Prüfung vertraut und soll diese Grundlagen auch nicht kennen müssen. Um die Bedienung der *SIFEI-Szenarienerfassung* nahe an den Bedienungsmodellen der Endnutzer zu realisieren wird die Metapher *Szenario* eingeführt, die folgende Terminologie nutzt:

Szenariowert - Ein Szenariowert wird zur Erstellung eines Szenarios in ein eine Eingabezelle eingegeben.

Erwartetes Ergebnis - Ein erwartetes Ergebnis wird zur Erstellung eines Szenarios in eine Zwischenergebnis- oder Ergebniszelle eingegeben.

Szenario - Ein Szenario ist die Abbildung einer fiktiven Benutzung des Spreadsheets für einen realen Anwendungsfall. Ein Szenario wird gebildet aus den je einem *Szenariowert* für jede Eingabezelle des Spreadsheets und je einem *erwartetem Ergebnis* mit einer Abweichung nach oben und unten für jede Zwischenergebnis- und Ergebniszelle. Ein Titel, eine Beschreibung, das Erstellungs-Datum und der Autor bilden die Metadaten des Szenarios.

4.4. Erfassung eines Szenarios

Zur Erfassung eines Szenarios werden durch den Nutzer zunächst die Zellen als Eingabezelle, Zwischenergebniszelle oder Ergebniszelle definiert, die für die Abbildung eines Anwendungsfalls auf ein Szenario benötigt werden. Zellen, die als Eingabe-, Zwischenergebnis- und

Ergebniszelle definiert wurden stehen später zur Szenario-Erstellung zur Verfügung. Eine oder mehrere Zellen werden markiert und per Klick auf den entsprechenden Toggle-Button als Eingabezelle, Zwischenergebniszelle, oder Ergebniszelle definiert. Ist eine Zelle definiert, wird der Button des jeweiligen Typs grün hinterlegt (vgl. Abb. 4.1). Eine Zelle kann entweder Eingabezelle, Zwischenergebniszelle oder Ergebniszelle sein.

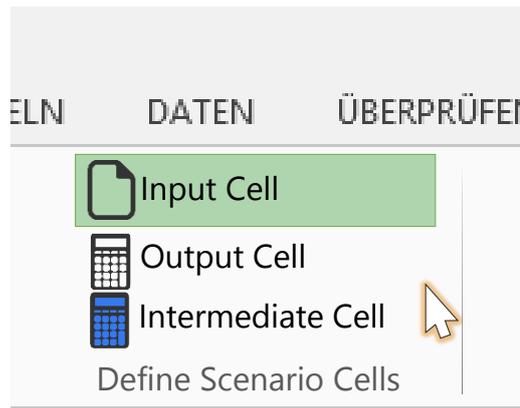


Abbildung 4.1.: Definieren von markierten Zellen als Eingabe-, Zwischenergebnis- oder Ergebniszelle. In dem dargestellten Mockup ist die markierte Zelle eine Eingabezelle.

Jede definierte Eingabezelle, Zwischenergebniszelle und Ergebniszelle wird durch ein Symbol auf der linken Seite der Zelle gekennzeichnet:



Abbildung 4.2.: Kennzeichnung von Eingabezellen



Abbildung 4.3.: Kennzeichnung von Zwischenergebniszellen

4. Konzeption der Konfiguration einer Spreadsheet Prüfung



Abbildung 4.4.: Kennzeichnung von Ergebniszellen

Wird ein neues Szenario erfasst, wird der Inhalt aller Eingabe-, Zwischenergebnis- und Ergebniszellen ausgeblendet. Alle restlichen Zellen werden deaktiviert und ausgegraut.

Der Nutzer gibt nun für jede Eingabezelle einen *Szenariowert* an. In die Zwischenergebniszellen und Ergebniszellen gibt der Nutzer ein *erwartetes Ergebnis* ein. Das erwartete Ergebnis kann vom Nutzer mit einer Abweichung versehen werden. Das Feld zur Erfassung der Abweichung wird durch einen Klick auf das Symbol der Zwischenergebniszelle oder Ergebniszelle geöffnet. Die Spreadsheet-Prüfung durch SIF prüft, ob das Ergebnis einer Formel, das aus den Szenariowerten der Eingabezellen berechnet wird, innerhalb der Abweichung des erwarteten Ergebnisses liegt.

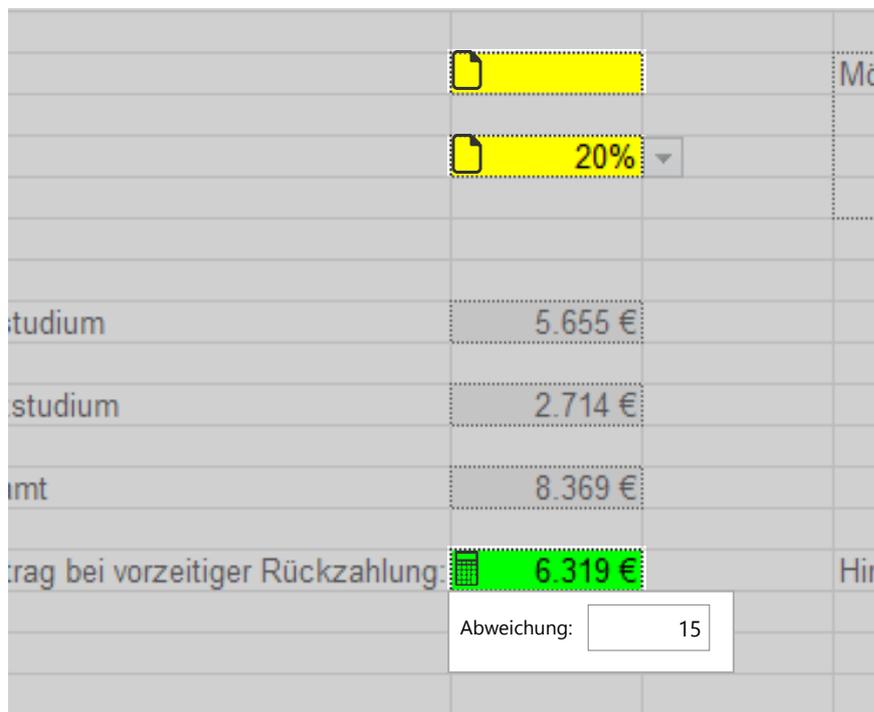


Abbildung 4.5.: Erfassen eines Szenarios

5. Umsetzung

Dieses Kapitel beschreibt Umsetzung des Konzeptes aus Kap. 4. Schwerpunkte sind Projektplanung, Architektur und Implementierung. Abschließend wird noch auf Abweichungen vom Lösungskonzept und Herausforderungen eingegangen.

5.1. Projektplanung

Zur Entwicklung der *SIFEI-Szenarienerfassung* wurde die *Evolutionäre Entwicklung* [Lud13, S. 171 f.] angewandt. Die Entwicklung fand in drei Zyklen statt. Im ersten Zyklus wurde die Architektur und die grafische Oberfläche entwickelt. Im zweiten Zyklus wurden Transformationen des Datenmodells zum Speichern, Laden und Ausführen von SIF implementiert. Im dritten Zyklus wurde die Integration und die Systemtests durchgeführt.

5.1.1. Schnittstellen zu anderen Projekten

Die *SIFEI-Szenarienerfassung* zur Konfiguration von SIF ist eine Komponente von SIFEI. Die Erweiterung von SIFEI vermeidet eine Kopplung zweier Excel-Addins. Es war nicht möglich die Entwicklung der *SIFEI-Szenarienerfassung* auf einen finalen Entwicklungsstand von SIFEI aufzubauen, da die Entwicklung der *SIFEI-Visualisierung* noch nicht abgeschlossen war. Zur gemeinsamen Konfigurationsverwaltung der parallelen Arbeiten wurde ein gemeinsames Repository genutzt.

5. Umsetzung

5.1.2. Zeitplan

Das folgende Timedrift-Diagramm verdeutlicht den Ablauf dieser Bachelor-Arbeit. Während des zweiten Zyklus wurde die Integration der Komponenten in eine eigene Stufe ausgelagert. Es wurde absehbar, dass die Integration von SIF in SIFEI einen größeren Aufwand beanspruchen wird. Der zusätzliche Aufwand wurde durch die nicht vollständig spezifizierte Schnittstelle des SIF verursacht.

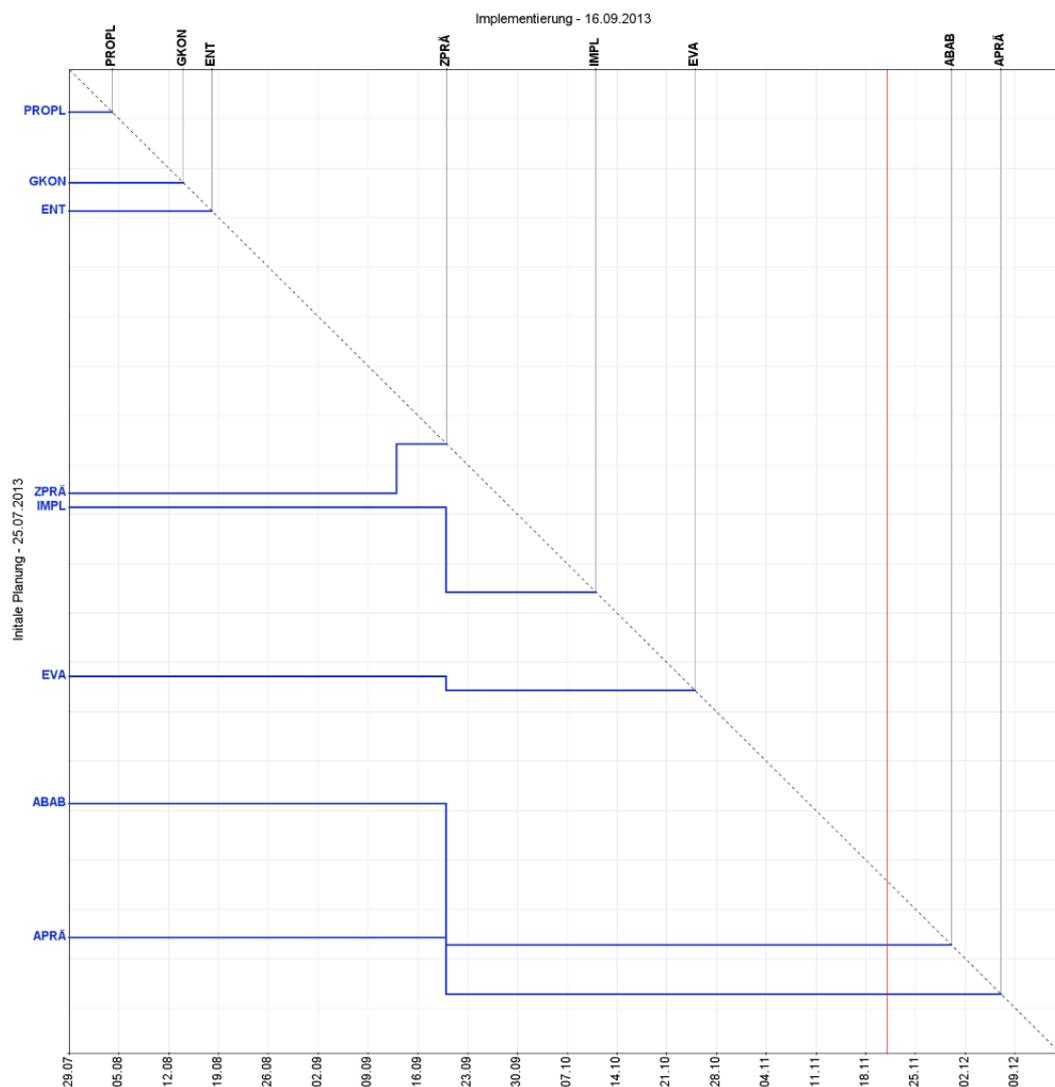


Abbildung 5.1.: zeitlicher Ablauf des Projektes

5.2. Architektur von SIFEI und SIF

Die beiden Module *SIFEI-Szenarienerfassung* und *SIFEI-Visualisierung* bilden zusammen die Komponente SIFEI. SIFEI startet eine Instanz der Komponente SIF und sendet die generierten Regeln im SpRuDeL-XML-Schema und das zu prüfende Spreadsheet über eine Socket-Schnittstelle an SIF. SIF prüft das Spreadsheet wie in Kap. 3.3 beschrieben. Abschließend werden die Ergebnisse der Prüfung im XML Format über die Socket-Schnittstelle an die *SIFEI-Visualisierung*-Komponente von SIFEI gesendet.

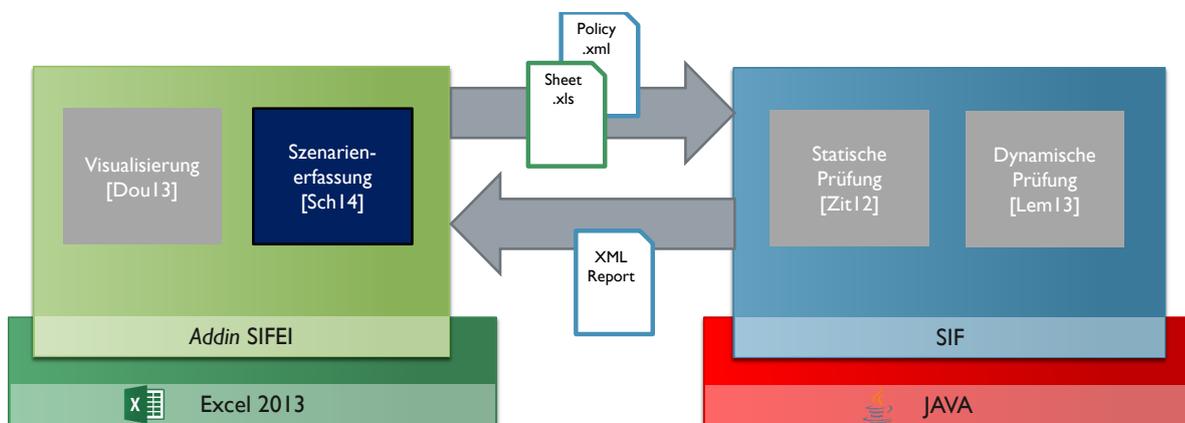


Abbildung 5.2.: Architektur der Komponenten SIFEI und SIF. Der Ablauf einer Prüfung durch SIF ist in Kap. 3.3 beschrieben.

5.3. Implementierung

Die Implementierung der Konfiguration einer Spreadsheet-Prüfung durch SIF beinhaltet die Definition von Eingabe-, Zwischenergebnis- und Ergebniszellen und eine Übersicht der definierten Zellen. Szenarien werden in den Zellen des Spreadsheets erfasst und in einer Seitenleiste angezeigt. Die erstellten Szenarien werden in der Spreadsheet-Datei gespeichert und zur Prüfung konvertiert.

5. Umsetzung

5.3.1. Eingabe-, Zwischenergebnis- und Ergebniszellen definieren

Eingabe-, Zwischenergebnis- und Ergebniszellen dienen zur Erfassung eines Szenarios. Markierte Zellen werden durch Klick auf einen der ToggI-Buttons der Ribbon-Group *Define Scenario Cells* definiert und stehen nun der Szenario-Erstellung zur Verfügung. In der Seitenleiste *Cell Definitions* wird eine Übersicht der definierten Eingabe-, Zwischenergebnis- und Ergebniszelle angezeigt. Die Selektion der Listenelemente ist mit der Selektion der Zellen im Spreadsheet synchronisiert.

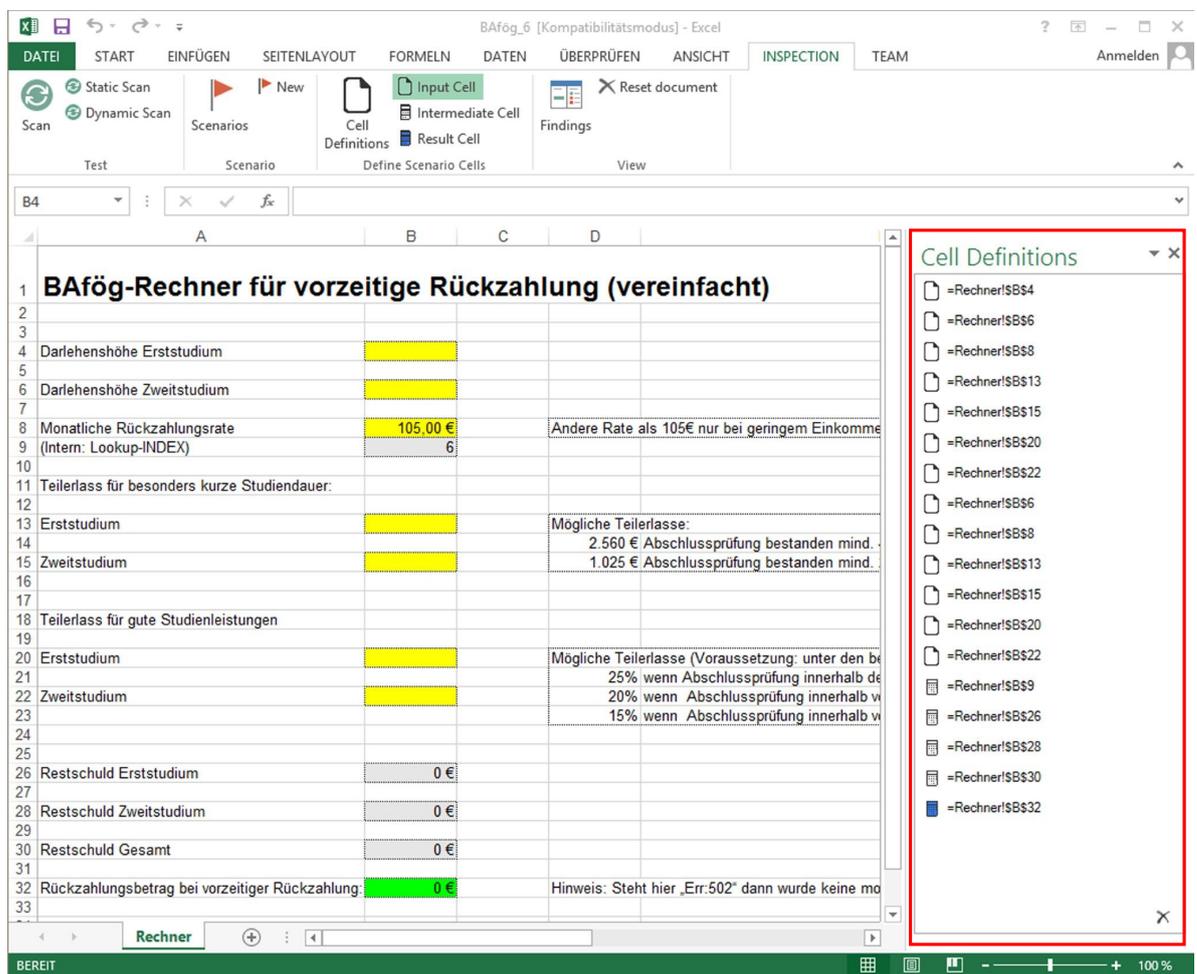


Abbildung 5.3.: Auflistung der definierten Zellen (siehe rote Markierung).

5.3.2. Erfassen eines Szenarios

In der Ribbon-Group *Scenarios* wird mit Klick auf *New* die Erstellung eines neuen Szenarios gestartet. Die Eingabe-, Zwischenergebnis- und Ergebniszellen werden mit Steuerelementen überdeckt, die ein Symbol und ein Textbox enthalten. In die Textbox werden durch den Benutzer die Szenariowerte und erwarteten Ergebnisse eingetragen. Mit Rechtsklick auf das Symbol einer Zwischenergebnis- oder Ergebniszelle wird ein spezialisiertes Kontextmenü zur Erfassung der Abweichung nach oben und unten geöffnet. Ein Klick auf den Expander erweitert das Kontextmenü um zwei freischaltbare Textboxen für die Eingabe einer expliziten Abweichung nach oben oder unten.

In der Ribbon-Group *Scenario* werden während der Erstellung eines Szenarios Buttons zum Abbruch oder zur Bestätigung der Szenario-Erstellung eingeblendet.

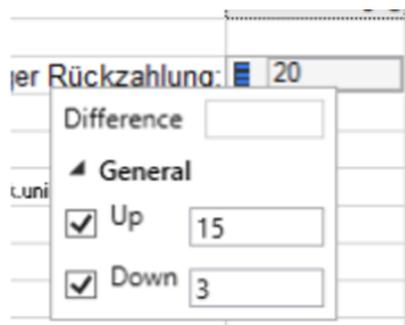


Abbildung 5.4.: Erfassen eines Szenarios

5.3.3. Übersicht der Szenarien

In der Seitenleiste *Scenarios* werden alle verfügbaren Szenarien mit Titel und einer Vorschau der Beschreibung aufgelistet (linke Seitenleiste in der Abb. 5.5). Ein Klick auf einen Listen-Eintrag öffnet eine zweite Seitenleiste, die die vollständige Ansicht des ausgewählten Szenarios enthält (vgl. Abb. 5.5). Das repräsentierte Datenmodell eines Szenarios ist zur besseren Übersicht in Expander gegliedert. Für jede definierte Eingabezelle, Zwischenergebniszelle und Ergebniszelle wird ein Tupel aus Zell-Adresse und des Szenariowerts bzw. des

5. Umsetzung

erwarteten Ergebnisses angezeigt. In der Detail-Ansicht ist es möglich einen Szenariowert oder ein erwartetes Ergebnis zu löschen.

The screenshot displays the Microsoft Excel interface with the 'BAFög_6 [Kompatibilitätsmodus] - Excel' window. The ribbon includes 'DATEI', 'START', 'EINFÜGEN', 'SEITENLAYOUT', 'FORMELN', 'DATEN', 'ÜBERPRÜFEN', 'ANSICHT', 'INSPECTION', and 'TEAM'. The 'INSPECTION' ribbon is active, showing options like 'Static Scan', 'Dynamic Scan', 'Scenarios', 'New', 'Input Cell', 'Cell Definitions', 'Intermediate Cell', 'Result Cell', 'Findings', and 'Reset document'. The spreadsheet shows a 'BAFög-Rechner für vorzeitige Rückzahlung' with columns A, B, and C. Rows 4-32 contain financial data, including 'Darlehenshöhe', 'Monatliche Rückzahlungsrate', 'Teilerlass', 'Restschuld', and 'Rückzahlungsbetrag'. The 'Scenario Manager' task pane is open, showing a list of scenarios: 'Bachelor + Master', 'Bachelor Studium', 'Bachlor + Master Abbruch', 'Student Hugo Müller', and 'Student Jan Hofer'. The 'Student Hugo Müller' scenario is selected, and its 'Scenario Data' pane is visible, showing a list of cell references and values: '=Rechner!\$B\$4' (7342), '=Rechner!\$B\$6' (2384), '=Rechner!\$B\$8' (105), '=Rechner!\$B\$13' (0), '=Rechner!\$B\$15' (1025), '=Rechner!\$B\$20' (25), '=Rechner!\$B\$22' (25), and '=Rechner!\$B\$32' (5120). The 'Up' and 'Down' buttons are also visible.

Abbildung 5.5.: Übersicht der Szenarien und Detail-Ansicht eines Szenarios

5.3.4. Starten einer Prüfung

Ist mindestens ein Szenario erstellt, kann eine dynamische Prüfung durchgeführt werden. Die dynamische Prüfung wird zusammen mit der statischen Prüfung mit Klick auf *Scan* gestartet. Die Ergebnisse sind in der Übersicht der Verstöße aufgelistet [Dou13].

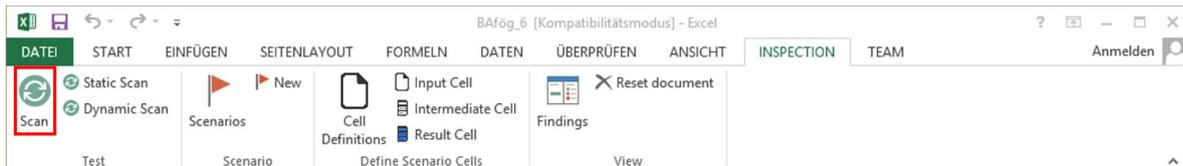


Abbildung 5.6.: Starten einer Prüfung (siehe rote Markierung).

Transformation der Szenarien zu Regeln im SpRuDeL-XML-Schema

Vor dem Start einer Prüfung durch SIF werden die Szenarien in das SpRuDeL-XML-Schema transformiert. Abb. 5.7 zeigt die Transformation der Szenarien auf das SpRuDeL-XML-Schema. Jedes Szenario eines Spreadsheets wird in eine Regel (siehe 3.3.3) einer dynamischen Prüfung durch SIF umgewandelt. Die Szenariowerte der Eingabezellen werden in Testeingaben umgewandelt. Eine Testeingabe besteht aus der jeweiligen Zell-Adresse und einem Testwert, der einem Szenariowert entspricht. Die erwarteten Ergebnisse der Zwischenergebnis- und Ergebniszellen werden in eine *TernaryCondition* oder *BinaryCondition* (siehe 3.3.3) umgewandelt. Ist ein erwartetes Ergebnis keine Zahl, wird das Szenario in eine *BinaryCondition* umgewandelt. Ist ein erwartetes Ergebnis eine Zahl wird das Szenario in eine *TernaryCondition* umgewandelt. Das Intervall der *TernaryCondition* wird aus dem erwarteten Ergebnis und der erlaubten Abweichung nach oben und unten gebildet.

5. Umsetzung

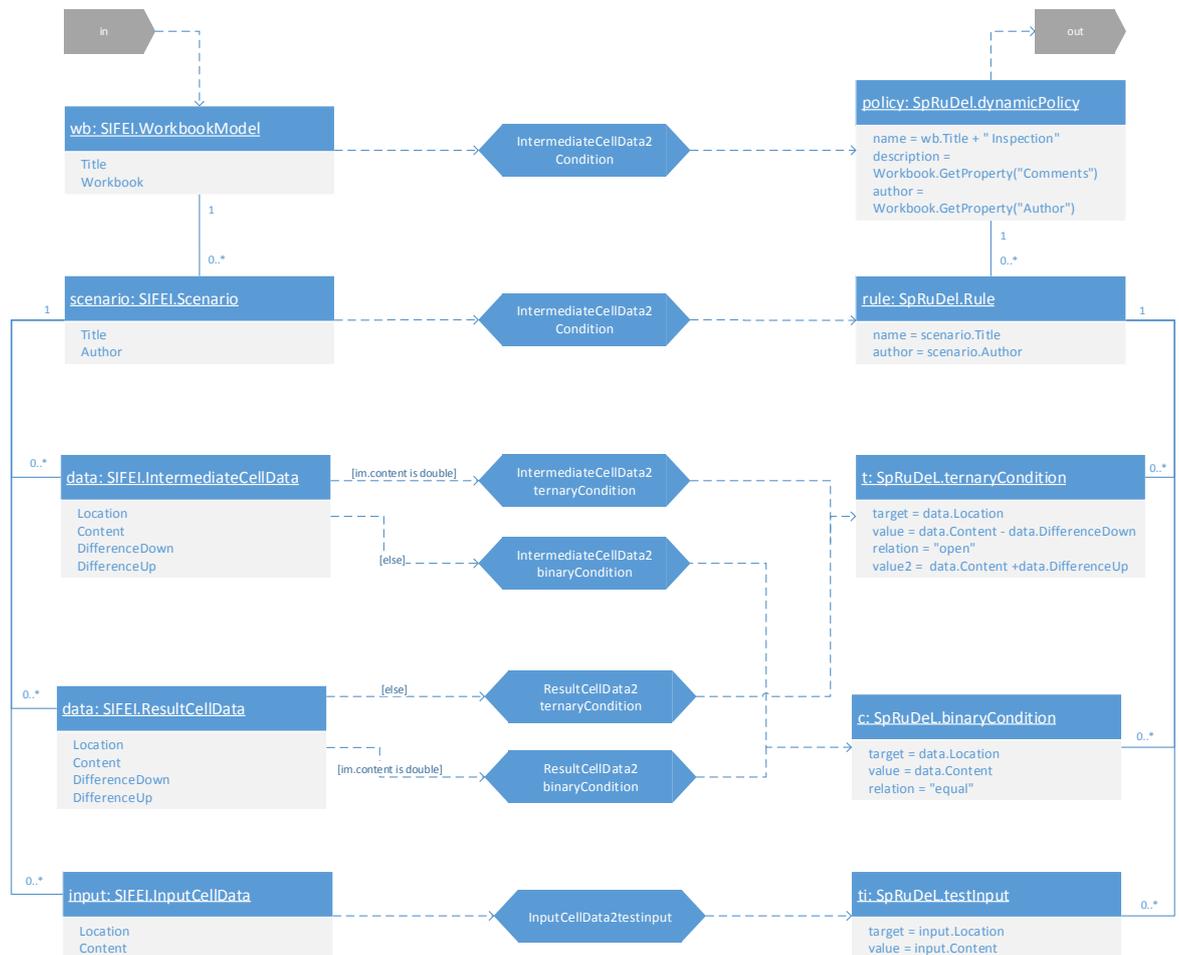


Abbildung 5.7.: Abbildung der Szenarien auf das SpRuDeL-XML-Schema als UMLX-Diagramm. Die Szenariowerte oder erwarteten Ergebnisse sind in den Variable *Content* gespeichert.

5.3.5. Speichern der Szenarien

Die Szenarien und definierten Zellen eines Spreadsheets stehen auch nach dem Beenden von Microsoft Excel zur Verfügung. Beim Speichern des Dokumentes wird das Datenmodell als *Custom XML Part* in der Excel-Datei gespeichert. Excel-Dateien die das Datenmodell der *SIFEI-Szenarienerfassung* enthalten können auch weiterhin von Excel-Instanzen ohne das SIFEI-Addin geöffnet und bearbeitet werden.

5.4. Herausforderungen

Während der Durchführung der Implementierung der *SIFEI-Szenarienerfassung* stellte sich das *Microsoft Office Excel Interop Interface* und das *Microsoft Office Excel Worksheet Interface* als eine Schnittstelle mit unzureichender Funktionalität heraus. In einem Spreadsheet eines Excel-Worksheets können zwar Steuerelemente platziert werden, allerdings können die Steuerelemente nicht mit einer Transparenz versehen werden, so dass die darunterliegenden Zellen teilweise sichtbar sind. Die Steuerelemente können nur in Zellen platziert werden. Auf Grund der technischen Einschränkungen in der Platzierung von Steuerelementen wurde auf das Ausblenden der Zellen verzichtet. Szenariowerte und erwartete Ergebnisse werden durch überlagerte Steuerelemente erfasst.

6. Evaluation

Im Anschluss an die Implementierung wurde eine Benutzerstudie durchgeführt, um die Usability der *SIFEI-Szenarienerfassung* zu validieren. Dieses Kapitel dokumentiert die Ziele, den Versuchsaufbau und die Durchführung der Evaluation der *SIFEI-Szenarienerfassung*. Abschließend werden die Ergebnisse zusammengefasst und die Schlussfolgerungen aus der Evaluation für die Weiterentwicklung von SIFEI dokumentiert.

6.1. Ziele

Ziel der Evaluation ist es zu validieren, ob die *SIFEI-Szenarienerfassung* folgende Eigenschaften erfüllt:

- Gute Erlernbarkeit
- Intuitive Bedienung
- Konzept der Spreadsheet-Prüfung durch SIFEI benötigt nur eine kurze Erklärung

6.2. Versuchsaufbau

Zur Validierung der in Kap. 6.1 genannten Ziele wurden Aufgaben erstellt, die mit SIFEI zu lösen waren. Im Ablauf der Aufgabenbearbeitung wurden Zwischenergebnisse in schriftlicher Form durch den Probanden dokumentiert. Um Synergien im Versuchsaufbau und der

6. Evaluation

Probandenrekrutierung nutzen zu können, wurde die Szenarienerfassung zusammen mit der *SIFEI-Visualisierung* [Dou13] evaluiert.

Die Evaluation gliederte sich in zwei Teile:

Teil I: *SIFEI-Visualisierung* (siehe Evaluation in [Dou13])

Teil II: *SIFEI-Szenarienerfassung*

Die Reihenfolge der beiden Evaluationsteile wurde entgegen des in Kap. 3.4 beschriebenen Ablaufprozesses einer Spreadsheet-Prüfung entworfen. Zuerst wurde die *SIFEI-Visualisierung* evaluiert, um bei der Evaluation der Szenarienerfassung auf eine Vorschau auf die Fehlervisualisierung verzichten zu können.

6.2.1. Aufgaben zur Szenarienerfassung

Grundlage der Aufgaben ist ein Spreadsheet, das bei einer vorzeitiger Rückzahlung des BAföG, die Restschuld berechnet. Studenten die BAföG beziehen können für besondere Studienleistungen einen Rabatt erhalten, wenn das Darlehen vorzeitig zurückbezahlt wird. Für die einzelnen Aufgaben sind in der Excel-Datei bereits Szenarien und definierten Zellen gespeichert.

Zeile	Spalte	Wert	Notiz
4	Darlehenshöhe Erststudium	13.824 €	
5	Darlehenshöhe Zweitstudium	0 €	
8	Monatliche Rückzahlungsrate	105,00 €	Andere Rate als 105€ nur bei geringem Einkommen möglich
9	(Intern: Lookup-INDEX)	6	
11	Teilerlass für besonders kurze Studiendauer:		
13	Erststudium	0,00 €	Mögliche Teilerlasse:
14			2.560 € Abschlussprüfung bestanden mind. 4 Monate vor Förderungshöchstdauer
15	Zweitstudium	0,00 €	1.025 € Abschlussprüfung bestanden mind. 2 Monate vor Förderungshöchstdauer
18	Teilerlass für gute Studienleistungen		
20	Erststudium	0%	Mögliche Teilerlasse (Voraussetzung: unter den besten 30% des Jahrgangs):
21			25% wenn Abschlussprüfung innerhalb der Förderungshöchstdauer bestanden
22	Zweitstudium	0%	20% wenn Abschlussprüfung innerhalb von 6 Monaten nach Ende der Förderungshöchstdauer bestanden
23			15% wenn Abschlussprüfung innerhalb von 12 Monaten nach Ende der Förderungshöchstdauer bestanden
26	Restschuld Erststudium	13.824 €	
27	Restschuld Zweitstudium	0 €	
28	Restschuld Gesamt	10.000 €	
32	Rückzahlungsbetrag bei vorzeitiger Rückzahlung	7.150 €	Hinweis: Steht hier „Err.502“ dann wurde keine monatliche Rückzahlungsrate ausgewählt!

Abbildung 6.1.: Aufgaben-Spreadsheet erstellt von Daniel Kulesz, 2012

Zellen definieren

Die erste Aufgabe hat zum Ziel zu validieren, ob Eingabezellen und Ergebniszellen ohne ausführliche Erklärung definiert werden können. Der Unterschied zwischen einer Ergebniszelle und einer Zwischenergebniszelle ist nicht groß. Dies kann zu Unsicherheiten des Probanden führen, dem der Unterschied nicht klar ist. Diese Störvariable wurde aus der Studie entfernt, indem gefordert war nur Eingabe- und Ergebniszellen zu definieren.

Ein Szenario erstellen

Zu Beginn der zweiten Aufgabe wird der Begriff *Szenario* definiert. Das zur Verfügung gestellte Spreadsheet enthält bereits definierte Eingabe und Ergebniszellen, die in der ersten Teilaufgabe abgefragt wurden. Das Vorgehen stellt sicher, dass der Proband nicht von Fehlern aus der vorherigen Aufgabe beeinflusst wird.

Hinweise wie ein Szenario erstellt wird, wurden in der zweiten Teilaufgabe nicht gegeben. Ziel der Aufgabe ist es die Intuitivität der Umsetzung des Konzeptes der Szenarien (siehe 4.3) zu validieren. Ein Ausschnitt aus einem fiktiven Blogeintrag, der alle erforderlichen Szenariowerte und erwarteten Ergebnisse des Szenarios enthält, wird in der Aufgabe zur Verfügung gestellt. In der folgenden Aufgabenbeschreibung wird angenommen, dass der Blogeintrag ein Tatsachenbericht einer Person über die eigene vorzeitige BAföG-Rückzahlung ist.

Nach der Erstellung des Szenarios soll vom Proband der Titel des zuvor erstellten Szenarios geändert werden. Die Aufgabe 2 dient zur Validierung der Erlernbarkeit der Szenarienübersicht.

Ein Szenario ändern

Das Spreadsheet der dritten Aufgabe enthält bereits 5 Szenarien. Ziel der Aufgabe ist es die Erlernbarkeit der Szenario-Detailansicht zu validieren. Der Proband hat zur Aufgabe ein bestehendes Szenario zu finden und einen bereits erfassten Wert einer Ergebniszelle zu korrigieren.

Abschließend wird mit der Frage nach gefundenen Fehlern für das geänderte Szenario evaluiert, ob einem Szenario Prüfergebnisse zugeordnet werden konnten.

6.2.2. Versuchsumgebung

Der Versuch fand im Studentenpool (Raum 1.014) des Informatikgebäudes der Universität Stuttgart statt. Dem Proband wurde ein Rechner mit dem Betriebssystem Windows 8, Microsoft Office 2013 und SIFEI zur Verfügung gestellt. Die Rechner waren mit einem 24“ Bildschirm ausgestattet. Um jedem Probanden die gleiche Ausgangssituationen zu garantieren wurde für jeden Probanden mit Hilfe einer Checkliste ein eigenes Benutzerkonto eingerichtet. Zur späteren Analyse wurde das Bildschirmbild aufgenommen und auf einen weiteren Bildschirm übertragen um das Handeln des Probanden zu beobachten und um ein Protokoll zu erstellen.

Um bei Probanden, die eine Aufgabe nicht lösen können den Versuch nicht abbrechen zu müssen, wurde für jede Teilaufgabe ein Lösungshinweis (siehe A.1) erstellt, der die Probanden in die Lage versetzt die Teilaufgabe abzuschließen und die folgende Aufgabe regulär zu bearbeiten. Die Benutzung jedes Lösungshinweises wurde im Protokoll dokumentiert.

In einem anschließendem Gespräch wurden die Eindrücke und Verbesserungsvorschläge der Probanden besprochen und im Protokoll dokumentiert.

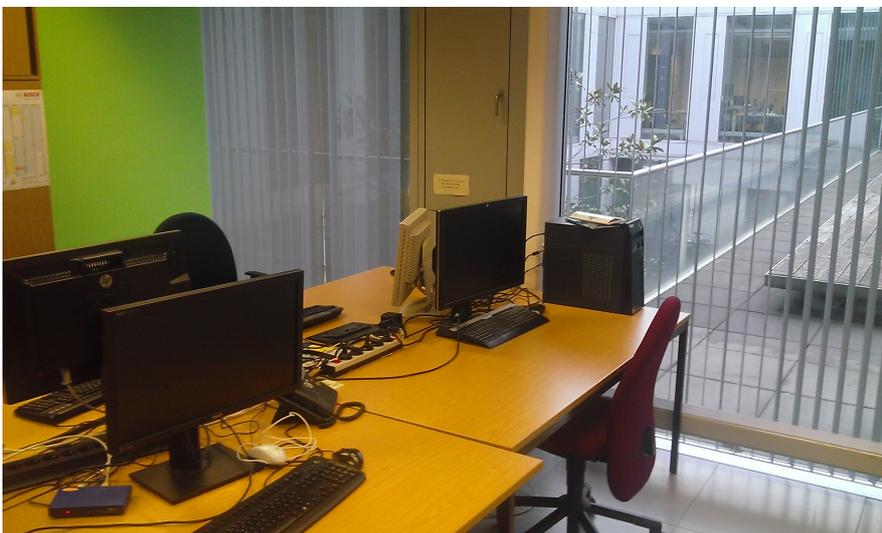


Abbildung 6.2.: Arbeitsplatz der Probanden am hinteren Schreibtisch.

6.3. Durchführung

Vor Beginn der Evaluation wurde eine Pilot-Studie mit drei Probanden durchgeführt. Die Probanden waren zwei Studenten der Softwaretechnik, sowie ein Student aus dem Bereich Maschinenbau. Die Studie selbst wurde mit zwei Probanden des Master-Studiengangs Energietechnik, einem Probanden des Studiengangs Erneuerbare Energien und einem Probanden des Studiengangs Luft- und Raumfahrttechnik durchgeführt.

6.4. Ergebnisse

Die Evaluation zeigt, dass es für die Probanden einfach ist, Zellen als Eingabe- oder Ergebniszelle zu definieren. Drei Probanden markierten jede Zelle einzeln und definierten die Zelle dann als Eingabe- oder Ergebniszelle. Nur ein Proband markierte alle gewünschten Zellen und definierte alle Zellen anschließend gemeinsam als Eingabezellen. Nur ein Proband nutzte die Hilfe.

Die Erstellung eines Szenarios bereitete einigen Probanden Schwierigkeiten. Drei von vier Probanden nutzten die Hilfe. Zwei Probanden gaben die Szenariowerte und erwarteten Ergebnisse zu erst direkt in die Excel-Zellen ein. Nach einigen Versuchen wurde das Szenario richtig erstellt. Bei der Erstellung des Szenarios erstellten zwei Probanden das Szenario ohne die erwarteten Ergebnisse. Das Anzeigen der Details eines Szenarios und das Ändern des Szenario-Namens wurde von alle Probanden erfolgreich ausgeführt.

Die Änderung eines Szenarios wurde von allen vier Probanden erfolgreich ausgeführt. Ein Proband konnte sich auf Grund der zu schmalen Seitenleiste nicht gut orientieren. Die Ergebnisse der Prüfung wurden anschließend von zwei Probanden nicht gefunden. Ein Proband nutzte dazu die Hilfe.

Im Fragebogen gaben drei von vier Probanden an, dass die Realisierung des Konzepts zur Erstellung eines Szenarios nicht einfach zu verstehen war. Drei von vier Probanden gaben an, dass die Ergebnisse der Prüfung nicht einem Szenario zugeordnet werden konnten. Im

Gegensatz dazu war für drei von vier Probanden der Ablauf einer Prüfung verständlich. Drei von vier Probanden können sich vorstellen SIFEI für eigene Spreadsheets zu verwenden.

6.5. Schlussfolgerungen

Die Evaluation zeigt, dass sich Szenarien eignen, um Spreadsheet-Prüfungen zu konfigurieren. Das Erfassen eines Szenarios der grafischen Oberfläche einer Spreadsheet-Ablaufumgebung ist für Benutzer einfach durchzuführen. Aus den Beobachtungen der Evaluation lässt sich schlussfolgern, dass der Ablauf der Prüfung nicht komplett verstanden wurde und Benutzer Unterstützung bei der Durchführung der ersten Prüfung benötigen. Alle Schlussfolgerungen aus den Ergebnissen der Evaluation sind auf Grund der geringen Anzahl von Probanden als eine Tendenz einordnen. Für belegbare Aussagen ist eine größere Stichprobe nötig.

Aus der Evaluation lassen sich einige Verbesserungen ableiten, die an der *SIFEI-Szenarienerfassung* durchgeführt werden sollten. Die folgende Tabelle listet benötigten Änderungen priorisiert und mit den veranschlagten Kosten auf.

6. Evaluation

Nr.	Komponente	Änderung	Grund	Kosten	Pr.
1	Zellen definieren	Hinweis auf Mehrfachselektion	Proband markieren jede Zelle einzeln	0,25 h	5
2	Szenario erfassen	Ablauf der Erstellung beschreiben	Probanden verstehen das Eingabekonzept nicht	1,00 h	4
3	Seitenleiste Szenario-Details	Breite anpassen	Übersicht verbessern	0,75 h	3
4	Seitenleiste Szenarios	Prüfstatus anzeigen	Probanden fanden die Ergebnisse der Prüfung nicht	1,00 h	3
5	Seitenleiste Szenario-Details	Prüfergebnis anzeigen	Probanden fanden die Ergebnisse der Prüfung nicht	3,00 h	2
6	Szenario erfassen	Assistent zur Erstellung eines Szenarios entwickeln	Probanden verstehen das Eingabekonzept nicht	7,00 h	1

Tabelle 6.1.: Priorisierte Liste von Änderungen, die sich aus der Evaluation von SIFEI ergeben.

7. Zusammenfassung und Ausblick

Die Konfiguration einer Spreadsheet-Prüfung durch Szenarien hat sich als ein vielversprechender Ansatz herausgestellt um den Bedarf an Experten zur Prüfung eines Spreadsheets durch das Spreadsheet Inspection Framework zu senken.

Das in dieser Arbeit entwickelte Konzept ermöglicht es Benutzern Spreadsheets im Rahmen der Spreadsheet-Erstellung zur prüfen. Die im Rahmen der Konzeption verwendete Methodik der Szenarien setzt keine Kenntnisse zur Prüfung von Spreadsheets voraus.

Die Konfiguration einer Spreadsheet-Prüfung durch das Spreadsheet Inspection Framework in Excel ermöglicht vielfältige Einsatzmöglichkeiten. Nach der Entwicklung eines Spreadsheet-Layouts können bereits Szenarien zur Prüfung der noch zu erstellenden Formeln angelegt werden. Im Rahmen des Wartungsprozesses eines Spreadsheets helfen die gespeicherten Szenarien Integrationsprüfungen [Lud13, S. 457] durchzuführen. Die Spreadsheet Inspection Framework Excel Integration bietet über den gesamten Lebenszyklus eines Projektes die Möglichkeit neue Funktionalitäten zu prüfen und nachzuweisen, dass alte Funktionalitäten von den Änderungen nicht beeinflusst werden.

Mit dem aktuellen Entwicklungsstand von SIFEI lassen sich nun erstmals alle Arbeitsschritte einer Spreadsheet-Prüfung die das Spreadsheet Inspection Framework zur Verfügung stellt in einer Spreadsheet-Ablaufumgebung durchführen.

Die Evaluation zeigt, dass sich Szenarien gut eignen um eine Spreadsheet-Prüfung zu konfigurieren. Erstbenutzer von SIFEI benötigen allerdings noch mehr kontextbezogene Unterstützung zur Verständlichkeit zum Ablauf einer Prüfung mit SIFEI.

Unberücksichtigt blieb in dieser Arbeit die Skalierbarkeit der in grafischen Oberfläche von SIFEI umgesetzten Konzepte. Für Benutzer die Spreadsheets mit mehreren tausend unterschiedlichen Formeln prüfen wollen, muss die Verwaltung der erstellten Szenarien weiter verbessert werden.

7.1. Zukünftige Arbeiten

Im Bereich von SIF und SIFEI besteht Potential in der Zukunft Verbesserungen vorzunehmen:

- Evaluation mit einer größeren Stichprobe, um genauere Aussagen zur Qualität von SIFEI machen zu können.
- Implementierung von weiteren statischen Prüfungen für mehr Feedback zur Qualität des geprüften Spreadsheets.
- Konfiguration der statischen Prüfung über SIFEI durch ein besonderes statisches Szenario. Im Datenmodell von SIFI ist diese Funktionalität bereits vorgesehen. Das SpRuDeL-XML-Schema bedarf einer Erweiterung.
- Visualisierung des Berechnungsbaumes (Funktionalität teilweise in Excel verfügbar).
- Automatisches Ausführen der Prüfungen während der Bearbeitung eines Spreadsheets.
- Visualisierung der Prüfvollständigkeit.
- Implementierung von Sprachpaketen

A. Anhang

A.1. Evaluation

Aufgabenstellung

Evaluation von SIFEI (Spreadsheet Inspection Framework Excel Integration)

Proband Nr: _____

Informationen zum Ablauf

- Diese Studie besteht aus zwei Aufgabenteilen. Zu jedem Aufgabenteil werden Sie anschließend in einem Fragebogen befragt. Nach der Bearbeitung der beiden Aufgabenteile werden wir Sie noch allgemein zu Ihrer Excel-Nutzung befragen.
- Sie finden auf dem Desktop ein Verzeichnis mit dem Namen „Experiment“, das alle benötigten Dateien enthält.
- Sollten Sie beim Lösen einer Aufgabe auf größere Probleme stoßen, nutzen Sie bitte die Hinweise auf den Hilfeblättern (aber bitte nur wenn Sie die Aufgabe sonst nicht lösen können).
- Bitte beachten Sie, dass in dieser Studie nicht alle Funktionen der Software evaluiert werden.

Worum geht es in dieser Studie?

Am Institut für Softwareentwicklung wurde SIFEI, ein System zur Prüfung von Spreadsheets („Excel-Programmen“ / „Excel-Tabellen“) entwickelt. SIFEI prüft Spreadsheets anhand von allgemeinen und benutzerdefinierten Regeln, zu jeder Regel kann es eine beliebige Anzahl von Verstößen geben.

In dieser Studie soll evaluiert werden, wie gut sich SIFEI für die Prüfung eines konkreten, realen Spreadsheets eignet.

Hintergrund

Sie haben vor einem Jahr ihr Studium an der Universität Stuttgart beendet, und sind nun in einem Unternehmen im Großraum Stuttgart beschäftigt. In absehbarer Zeit steht die Rückzahlung ihres „BAföG-Darlehens“ an. Für Anfang nächsten Jahres liebäugeln Sie, mit der Anschaffung eines neuen Autos. Sie wissen jedoch nicht, ob es eine gute Idee ist, sich das Auto zu kaufen, da Ihnen in Kürze auch die Rückzahlung Ihres „BAföG-Darlehens“ ins Haus steht.

Ein ehemaliger Kommilitone hat vor einiger Zeit ein Spreadsheet erstellt, das den Rückzahlungsbetrag für das „BAföG-Darlehen“ ermittelt. Das Spreadsheet berücksichtigt dabei auch den Rabatt, den Sie durch eine vorzeitige Rückzahlung erhalten würden.

Doch kann man diesem Spreadsheet vertrauen? Sie hoffen mit Hilfe von SIFEI eine Antwort auf diese Frage zu finden.

TEIL II: Szenarien

In Teil I der Evaluation wurden Fehler im Spreadsheet analysiert. Neben den Prüfungen der Leserichtung, Zellkomplexität und „Konstanten in Formeln“ wurden im Teil I auch bereits Szenarien geprüft, die von einem Nutzer selbst erstellt wurden.

Im nun folgenden Teil II sollen Szenarien erstellt werden, mit denen die Formeln des Spreadsheets geprüft werden können. Ein Szenario besteht aus einem Eingabewert für jede Eingabezelle und einem erwarteten Ergebnis mit einer Abweichung für jede Ergebniszelle.

Die folgende Abbildung beschreibt einen beispielhaften Prüfungsvorgang.



Abbildung 1: Ablauf einer Prüfung

Multifunktionsleiste (Ribbon)

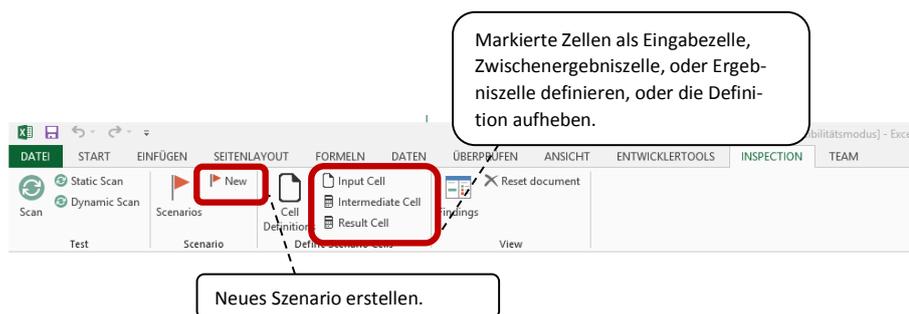


Abbildung 2: Inspection-Abschnitt in der Multifunktionsleiste von Excel 2013 - Szenarien

Aufgabe 4: Zellen definieren

- 1.) Öffnen sie aus dem Aufgabenverzeichnis die Excel Datei „BAföG_4.xls“.
- 2.) Die Zellen in Spreadsheets werden unter anderem als Eingabezelle oder Ergebniszelle genutzt. Entscheiden Sie, wie die einzelnen Zellen im BaföG Rechner verwendet werden und definieren Sie diese jeweils als Eingabezelle, oder Ergebniszelle.
- 3.) Speichern Sie das Dokument ab.

Aufgabe 5: Ein Szenario erstellen

Szenario - Ein Szenario besteht aus einem Eingabewert für jede Eingabezelle und einem erwarteten Ergebnis mit einer Abweichung für jede Ergebniszelle. Wird die Prüfung mit Klick auf Scan durchgeführt wird geprüft ob die Ergebnisse, die eine Formel einer Ergebniszelle aus den Werten der Eingabezellen berechnet in den Grenzen liegen, die in dem Szenario für diese Ergebniszelle festgelegt wurden.

- 1.) Öffnen sie aus dem Aufgabenverzeichnis die Excel Datei „BAföG_5.xls“.
- 2.) Dieses Spreadsheet enthält bereits Zell Definitionen. Welche Zellen sind als Eingabezelle definiert? Geben Sie die Zellen als Liste an (z.B. A1, B2, C7)

Zellen: _____

- 3.) Damit sie sich von der Korrektheit des Spreadsheet überzeugen können, ist es sinnvoll das Spreadsheet mit sehr vielen unterschiedlichen Szenarien zu prüfen, um eine möglichst großen Anwendungsbereich abzudecken.

Im Internet haben Sie einen Bericht eines Ingenieurs gefunden, der in seinem Blog über seine BAföG-Rückzahlung berichtet:

„... Nachdem ich mein Bachelor-Studium 2 Monate vor Ende der Förderungshöchstdauer beendet hatte, begann ich an der Universität in Stuttgart ein Master-Studium. Im Bachelor-Studium wurde ich mit einem Darlehen von 8.523 € gefördert. Während meines Master-Studiums betrug die Förderung nur 2.100 € Meine Master-Arbeit beendete ich 5 Monate nach Ende der Förderungshöchstdauer. Mein Master-Abschluss war unter den 30 % der besten Studenten meines Jahrgangs. Nach der etwas langwierigen Bearbeitung meines Antrags auf vorzeitige Rückzahlung betrug mein Rückzahlungsbetrag betrug meine Restschuld für das Bachelor-Studium 7.498 € und die Restschuld für mein Master-Studium 1.680 €. Ich habe alle meine Restschulden vorzeitig zurückgezahlt. Die Restschuld bei vorzeitiger Rückzahlung betrug 6.777 €.“

Wir gehen davon aus, dass sich dabei um einen Tatsachenbericht handelt und möchten prüfen, ob der BAföG-Rechner die gleichen Ergebnisse liefert.

Erstellen Sie ein Szenario um damit das Spreadsheet prüfen zu können. Für die Ergebniszellen wählen sie eine Abweichung von 5 (€, ohne Einheit!) nach unten und eine Abweichung von 10 (€) nach oben.

- 4.) Das SIFEI hat für dieses Szenario automatisch einen Namen erstellt. Wie heißt dieser?

Name: _____

- 5.) Benennen Sie das Szenario in „Flotter Bauigel“ um.
- 6.) Speichern Sie das Dokument ab.

Aufgabe 6: Ein Szenario ändern

- 1.) Öffnen sie aus dem Aufgabenverzeichnis die Excel Datei „**BAföG_6.xls**“.
- 2.) In den Spreadsheet befindet sich das Szenario „Student Jan Hofer“.
Wer hat das Szenario erstellt?

Name: _____

- 3.) In dem Szenario „Student Jan Hofer“ wurde vom Ersteller des Szenarios für die Zelle ‚B32 ‘ ein falsches erwartetes Ergebnis angegeben. Nach der Prüfung stellen Sie fest, dass der Wert nicht stimmt (Es sollte 5900 statt 62 sein).

Finden und korrigieren Sie diesen Fehler. Legen Sie zusätzlich eine erlaubte Abweichung nach oben und unten auf 15 (€, ohne Einheit!) fest.

- 4.) Scannen Sie das Spreadsheet. Tritt ein Fehler für das Szenario „Student Jan Hofer“ noch auf?

ja

nein

- 5.) Speichern Sie das Dokument ab.

Hilfestellungen

TEIL II: Hilfestellung

Aufgabe 4

2.)

Markieren Sie eine oder mehrere Zellen und klicken Sie auf ‚Input Cell‘ oder ‚Result Cell‘.



Wählen Sie ‚Input Cell‘ für Zellen mit Eingabewerten (ohne Formeln).

Wählen Sie ‚Result Cell‘ für Zellen mit Formeln.

Eingabezellen – Zellen im Spreadsheet, die konkrete Werte enthalten und zur Berechnung von Formeln in anderen Zellen dienen.

Ergebniszellen – Zellen im Spreadsheet, deren Wert sich aus einer Formel berechnet. Das Ergebnis wird von keinen weiteren Zellen genutzt

TEIL II: Hilfestellung

Aufgabe 5

2.) Zell Definitionen

Diese Seitenleiste zeigt eine Übersicht der definierten Eingabe-, Zwischenergebnis-, und Ergebniszellen an. Klicken Sie auf das Icon oder den Text um zu der Zelle zu springen.



3.) Szenario erstellen

Szenario – Ein Szenario besteht aus einem Eingabewert für jede Eingabezelle und einem erwarteten Ergebnis mit einer Abweichung für jede Ergebniszelle.

Klicken Sie auf ‚New‘ um die Erstellung eines neuen Szenarios zu starten.

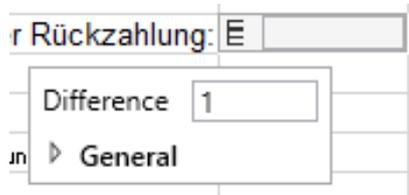


Für jede definierte Eingabe- oder Ergebniszelle wird anstatt der Zelle ein Symbol für den Zelltyp (Eingabezelle oder Ergebniszelle) und ein Eingabefeld angezeigt.

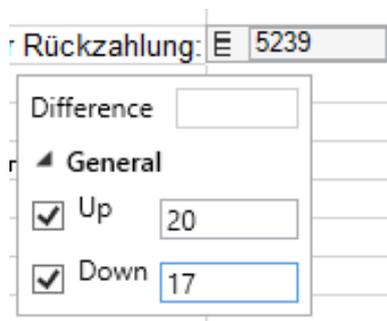


Geben Sie in die Eingabefelder die passenden Werte ein. Beachten Sie die Zellbeschriftungen auf der linken Seite.

Um eine Abweichung des Wertes einer Ergebniszelle festzulegen, klicken Sie mit der rechten Maustaste auf das Icon. Sie können nun die Abweichung nach oben und nach unten einstellen (im Beispiel ist eine Abweichung nach oben und unten von 1 eingestellt).



Mit Klick auf den Expander öffnen sich die Eingabefelder, um die Abweichung nach oben und unten explizit anzugeben. Aktivieren Sie dazu die beiden Checkboxes.

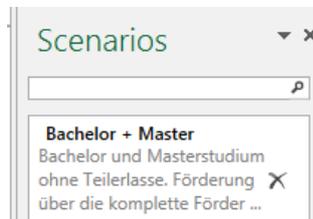


4.) Name des Szenarios

Klicken Sie auf Szenarios um eine Übersicht der Szenarien zu erhalten.

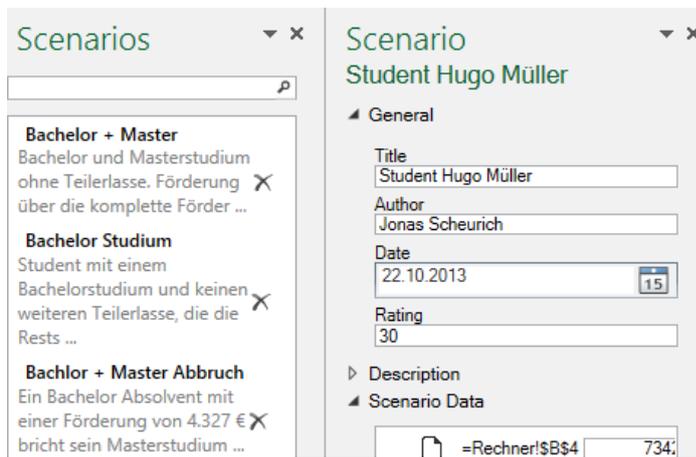


Es öffnet sich die Szenario Übersicht. Diese Seitenleiste listet die erstellten Szenarien mit Titel und Beschreibung auf. Mit Klick auf ein Szenario kann dieses Szenario in der Detailansicht geöffnet werden.



5.) Szenario umbenennen

Klicken Sie auf das Szenario „New Scenario - <Datum>“. Darauf öffnet sich die Detailsicht des Szenarios.



Ändern Sie unter Titel den Namen.

TEIL II: Hilfestellung

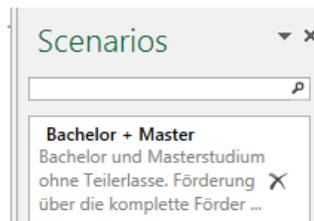
Aufgabe 6

2.) Szenario „Student Jan Hofer“

Klicken Sie auf Szenarios um eine Übersicht der Szenarien zu erhalten.



Es öffnet sich die Szenario Übersicht. Diese Seitenleiste listet die erstellten Szenarien mit Titel und Beschreibung auf. Mit Klick auf ein Szenario kann dieses Szenario in der Detailansicht geöffnet werden.

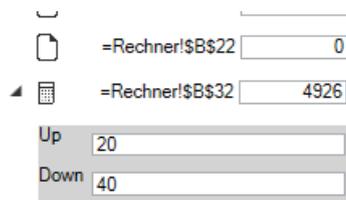


Klicken Sie auf das Szenario „Student Jan Hofer“. Daraufhin öffnet sich die Detailansicht.

Diese Seitenleiste zeigt die Beschreibung eines Szenarios, sowie die erfassten Werte an. Mit Klick auf den Expander eines Wertes für eine Zwischenergebnis- oder eine Ergebniszelle können die Grenzen der Abweichung nach unten und oben angezeigt werden.

Ändern Sie den Wert mit der Beschreibung „=Rechner!\$B\$32“

Klicken Sie auf den Expander. Nun öffnen Sie die beiden Abweichungen nach oben und unten. Ändern sie die beiden Werte.



A.2. Inhalt der Daten-CD

Diese Arbeit ist auf einer CD dokumentiert, die folgenden Inhalt enthält:

- Projektplan
- Implementierung
- Installationsdateien
- Installationsanleitung
- Handbuch
- Checkliste zur Vorbereitung der Evaluation
- Protokoll der Evaluation
- Videoaufzeichnung der Evaluation
- Ausarbeitung im PDF-Format
- Kurzfassung und Abstract als Plain Text

Abkürzungsverzeichnis

CER Cell Error Rate

ISTE Institut für Softwaretechnologie

SaaS Software as a Service

SIF Spreadsheet Inspection Framework

SIFEI Spreadsheet Inspection Framework Excel Integration

SpRuDeL Spreadsheet Rule Definition Language

VPL Visual Programming Language

WYSIWYT What You See Is What You Test

Literaturverzeichnis

- [BNV12] T. Berberich, A. B. Nguyen, M. Vetter. Audit-Werkzeuge für Spreadsheets. Fachstudie Softwaretechnik: Universität Stuttgart, Institut für Softwaretechnologie, Software Engineering, 2012. (Zitiert auf Seite 14)
- [Buwo4] H. Buwalda. Soap opera testing. *Better Software*, 6(2):30–37, 2004. (Zitiert auf Seite 17)
- [CFM⁺12] J. Cunha, J. P. Fernandes, P. Martins, J. Mendes, J. Saraiva. Smellsheet detective: A tool for detecting bad smells in spreadsheets. In *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*, S. 243–244. IEEE, 2012. (Zitiert auf Seite 14)
- [CFRS12] J. Cunha, J. P. Fernandes, H. Ribeiro, J. Saraiva. Towards a catalog of spreadsheet smells. In *Computational Science and Its Applications–ICCSA 2012*, S. 202–216. Springer, 2012. (Zitiert auf Seite 14)
- [Dou13] E. Doust. *Visualisierung von Fehlern in Spreadsheets*. Bachelorarbeit, Universität Stuttgart, Institut für Softwaretechnologie, Software Engineering, erscheint 2013. (Zitiert auf den Seiten 8, 18, 24, 37 und 42)
- [eus] European Spreadsheet Risks Interest Group. URL <http://www.http://eusprig.org/>. (Zitiert auf Seite 7)
- [IEE10] IEEE Standard Classification for Software Anomalies 1044-2009, 2010. (Zitiert auf Seite 13)

- [Kul13] O. J. Kulesz, D. Practical Challenges with Spreadsheet Auditing Tools. Proceedings of the EuSpRIG 2013 Conference, 2013. (Zitiert auf Seite 7)
- [Lem13] M. Lemcke. *Dynamische Pruefung von Spreadsheets*. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, 2013. (Zitiert auf den Seiten 8, 18, 19, 20, 21 und 24)
- [Lig09] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Spektrum Akademischer Verlag, 2009. (Zitiert auf Seite 13)
- [Lud13] H. Ludewig, J.; Lichter. *Software Engineering - Grundlagen, Menschen, Prozesse, Techniken*. dpunkt.verlag, Heidelberg, 2013. (Zitiert auf den Seiten 14, 31 und 49)
- [Pano8a] R. R. Panko. What We Know About Spreadsheet Errors. *Journal of End User Computing's*, 1998 (revidiert 2008). (Zitiert auf den Seiten 7, 13 und 14)
- [Pano8b] R. R. Panko. Reducing Overconfidence in Spreadsheet Development. *CoRR*, S. 49 – 66, 2008. (Zitiert auf Seite 14)
- [Pano8c] R. R. Panko. Spreadsheet Errors: What We Know. What We Think We Can Do. *CoRR*, S. 9, 2008. (Zitiert auf Seite 13)
- [Rot98] L. L. D. C. B. M. Rothermel, G. What you see is what you test: a methodology for testing form-based visual programs. *Proceedings of the 20th International Conference on Software Engineering*, 1998. (Zitiert auf den Seiten 15 und 17)
- [SGP13] B. L. Stephen G. Powell, Kenneth R. Baker. A critical review of the literature on spreadsheet errors. *Decision Support Systems*, S. 11, 2013. (Zitiert auf Seite 14)
- [Tea10] Team23GmbH. International OpenOffice market shares, 2010. URL <http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html>. (Zitiert auf Seite 11)
- [Zit12] S. Zitzelsberger. *Fehlererkennung in Spreadsheets*. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, 2012. (Zitiert auf den Seiten 8 und 18)

Alle URLs wurden zuletzt am 25. 11. 2013 15.00 Uhr geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift