

# Web-Enabling von fecher Mit frischer Oberfläche stark im Geschäft

## Finanzlösung SASKIA.H2R begeistert nach Web-Enabling-Projekt von fecher neue Anwenderkreise

*SASKIA hat eine bewegte Geschichte hinter sich: Vor gut 30 Jahren als Kommunale Informationsverarbeitung Südsachsen gegründet, entwickelte das Unternehmen die bis heute landesweit führende HKR- und IFR-Finanzsoftware für die öffentliche Hand. Mit dem Erfolg in weiteren Bundesländern erfolgte 2001 die Umbenennung in SASKIA Informations-Systeme GmbH, 2006 der Einstieg des Rechenzentrum-Partners KISA und im Jahr 2014 die Übernahme durch Robotron Datenbank-Software GmbH. Nachdem die ursprünglich in Gupta SQLWindows geschriebene Software 2011 in einem Porting-Projekt von fecher bereits auf die .NET-Plattform migriert worden war, stand 2022 erneut ein Technologiewechsel an. Diesmal galt es, das Front-End von einer klassischen WinForms-Anwendung in eine moderne Web-Oberfläche zu konvertieren. Erneut konnte fecher helfen – mit einem Web-Enabling-Projekt, das zum Jahresende 2023 erfolgreich abgeschlossen wurde. Unter dem Namen SASKIA.H2R kam die neue Softwaregeneration im Frühjahr 2024 schließlich auf den Markt.*

„Funktional braucht sich unsere Software nach 30 Jahren am Markt mit permanenter Weiterentwicklung wahrhaft nicht zu verstecken“; berichtet Ralf Köhler, Bereichsleiter der Softwareentwicklung bei SASKIA. Von der simplen Rechnungsstellung über



**Als IT-Spezialist mit Hauptsitz in Chemnitz betreut die 1993 gegründete SASKIA Informations-Systeme GmbH über 400 Kommunalverwaltungen und öffentliche Ämter.**

den Grundsteuerbescheid bis hin zur kompletten Haushaltsführung im Doppik-Verfahren deckt die Lösung alle Anforderungen ab, die Landkreise, Städte und Gemeinden an ein modernes Finanzwesen stellen. Optisch allerdings konnte man der Lösung ihre Herkunft aus unterschiedlichen Jahrzehnten und Technologiegenerationen schon ansehen. So waren in den letzten Jahren die Stimmen aus Marketing und Vertrieb lauter geworden, die eine neue Benutzeroberfläche forderten. „Wir waren definitiv nicht die Lösung, die in Ausschreibungen und Präsentationen auf den ersten Blick punkten konnte“, erinnert sich Köhler selbstkritisch.

Anfang 2022 startete er deshalb ein Evaluierungsprojekt, das den Aufwand prüfen sollte, die mittlerweile in C# auf .NET laufende Software mit einem Browser-Frontend auszustatten. „Web-Technologie war gesetzt. Noch einmal auf eine Desktop-Oberfläche zu setzen, kam für uns nicht in Frage“, berichtet der Bereichsleiter. Die eigenen Versuche mit dem Angular-Framework kamen jedoch bald wieder zum Erliegen. „Angular gilt zwar als Standard und bietet gute Perspektiven, ist aber auch ausgesprochen komplex“, so seine Erfahrung. „Weder Performance noch Entwicklerproduktivität waren auf dem Niveau, das wir gewohnt waren und weiterhin gebraucht hätten. Obendrein findet man kaum Entwickler auf dem Arbeitsmarkt, die diese speziellen Technologien beherrschen.“

### Mit Vollgas ins Web

Bei rund 1,8 Millionen Codezeilen und gut 1.500 Masken hätte die Neuentwicklung eines Web-Frontends auch viel zu viele Ressourcen verbraucht. „Im Prinzip hätten wir jahrelang parallel entwickelt, bis wir überhaupt wieder beim heutigen Funktionsumfang angekommen wären“, weiß Köhler. „Um die volle Funktionalität zu erhalten und das Zeitfenster für die Umstellung möglichst kurz zu halten, kam für uns deshalb nur eine Portierung in Frage.“

Eben diese bot fecher an, der Spezialist für Anwendungsmodernisierung, der 2011 bereits die Portierung von Gupta nach .NET durchgeführt hatte. Ein sogenanntes Web-Enabling sollte die vorhandenen WinForms-Masken in einem weitgehend automatisierten Prozess auf das Wisej.NET-Framework bringen, das eine Ausführung im Web-Browser ermöglicht. Parallel dazu war ein Redesign der Benutzeroberfläche geplant. Als Ergebnis würde eine echte Web-Anwendung mit modernem Erscheinungsbild

entstehen, die neben Windows auch alle anderen gängigen Client-Plattformen unterstützt.



**Ralf Köhler,**  
**Bereichsleiter**  
**der Software-**  
**entwicklung**  
**bei SASKIA**

„fecher kannten wir ja bereits aus dem vorherigen Projekt, das für uns mehr als zufriedenstellend verlaufen war“, erläutert Köhler. „Und Wisej.NET hat uns rasch überzeugt, als wir uns damit beschäftigt haben. Der größte Vorteil für unser Team bestand darin, dass wir weiterhin mit C# in Visual Studio entwickeln würden – auch die Web-Oberfläche – und uns somit kaum umstellen mussten.“ Die Geschäftsleitung und Robotron als Gesellschafter begeisterte vor allem, dass das Web-Enabling von fecher zum Festpreis und mit garantierten Fertigstellungstermin angeboten wurde. „So hatten wir kein Risiko zu tragen und konnten mit unserem Team in der Zwischenzeit bereits die Weiterentwicklung voranbringen.“

## Eine besondere Zusammenarbeit

Im November 2022 erfolgte schließlich der Auftrag. Beim gemeinsamen Kick-off für das Projekt wurde die Abnahme für Ende September 2023 geplant. Nachdem SASKIA Anfang Januar den Quellcode der Software an fecher geliefert hatte, arbeitete das dortige Team zunächst an der toolbasierten Umsetzung der Web-Oberfläche und den notwendigen manuellen Nacharbeiten, bis im Februar eine erste Version ohne Fehlermeldungen durch den Compiler lief. Parallel dazu erstellte das SASKIA-Team zahlreiche Screen-Videos, die die Bedienung der Anwendung zeigten und von fecher später verwendet werden konnten, um die korrekte Funktion der neuen Lösung zu testen.



**SASKIA.H2R: Fortschrittliche Finanzsoftware mit vertrauter Funktionalität und modernster Webtechnologie.**

„Charakteristisch war, dass wir immer Hand in Hand gearbeitet und gemeinsam auf die Lösungsfindung hingewirkt haben. Dabei standen für uns alle immer die Effizienz, die Einfachheit und die Wartbarkeit im Vordergrund“, berichtet Köhler und nennt als Beispiel die Grid-Funktionalität: „Dafür hatten wir eigentlich schon die Komponentenbibliothek eines

Drittherstellers ins Auge gefasst, weil wir diese funktional und auch optisch sehr ansprechend fanden. Aber nach Tests mit nur 30.000 Datensätzen hat sich einfach herausgestellt, dass das Grid unter der bei uns typischen Last nicht zuverlässig funktioniert. Gemeinsam mit fecher haben wir deshalb entschieden, die eingebaute Funktionalität von Wisej.NET zu verwenden. Der Funktionsumfang des Wisej-Datagrids ist zwar nicht ganz so vielfältig. Dafür haben wir aber die Garantie, dass es auch bei großen Datenmengen performant arbeitet.“

Die enge Abstimmung der Projektpartner hat sich im weiteren Verlauf fortgesetzt und vor allem in der Testphase bewährt. Hier meldete das SASKIA-Team gefundene Fehler nämlich häufig gar nicht erst an fecher, sondern behob diese wo immer möglich einfach selbst. „Dass fecher uns noch vor der Abnahme Zugriff auf das Repository gab, war natürlich ein großer Vertrauensbeweis“, weiß Köhler. „Da unsere Zusammenarbeit schon so lange besteht, konnten wir diesen Weg gehen, der es allen Beteiligten viel leichter gemacht hat.“

## Zwischendurch noch ein Redesign

Ein Problem, das sich nicht so einfach lösen ließ, ergab sich während des Projekts noch aus der Entscheidung, bei der Umstellung auf die Web-Oberfläche die Beschriftung jeweils über die Eingabefelder setzen zu lassen. „Das hatten wir anfangs ziemlich spontan so festgelegt und es war auch in 80 Prozent der Fälle die beste Lösung“, erinnert sich Köhler. „Leider hat es aber eine ganze Reihe von Masken mit vielen Feldern durcheinandergebracht.“ Deshalb mussten alle Masken manuell nachbearbeitet werden. Aus der Not wurde kurzerhand eine Tugend gemacht und entschieden, eine zweimonatige Redesign-Phase einzuschieben. „Das hat zwar den Projektfahrplan etwas verzögert, sich im Ergebnis aber mehr als ausgezahlt.“ Auch

hier arbeiteten die Teams von fecher und SASKIA Hand in Hand, um die Zusatzarbeiten rasch zu bewältigen.

Die so entstandene neue Softwaregeneration SASKIA.H2R bekamen die ersten Kunden bei einer Roadshow im Frühjahr 2024 zu sehen. „Für unsere bestehenden Anwender ist natürlich am wichtigsten, dass sie SASKIA.H2R in vertrauter Form bedienen können, ohne sich umstellen zu müssen.“ Um das sicherzustellen, hat die Web-Oberfläche weitestgehend die vertraute Tastaturbedienung erhalten, auch wenn diese aufgrund der Abhängigkeit vom Browser vom Desktop abweicht. Überhaupt ist der Aufbau der Benutzeroberfläche trotz des neuen, frischen Looks nahezu gleichgeblieben. „Über die typischen Mobil-Apps mit drei Feldern kann der Sachbearbeiter in der Verwaltung nur lachen“, so Köhler. „Bei ihm sind es 30 Felder, die er alle auf einen Blick sehen und bearbeiten muss. Genau für solche transaktionalen Anwendungen im Browser ist Wisej.NET ausgelegt und war für uns deshalb genau die richtige Lösung.“

Zugleich sind Vertrieb und Marketing mit dem Erscheinungsbild des neuen SASKIA.H2R mehr als zufrieden. „In der Verwaltung werden auf absehbare Zeit weiterhin Desktop-Arbeitsplätze zum Einsatz kommen. Aber auch dort ist der Generationswechsel zu spüren. Viele neue Mitarbeiter sind die üblichen Standards einer modernen Web-Anwendung einfach gewohnt“, fasst Köhler zusammen. „Und spätestens wenn der Bürgermeister in der Präsentation sitzt, ist eine frische und aufgeräumte Oberfläche gleich ein dicker Pluspunkt.“



*Bei einer Roadshow im Frühjahr 2024 erlebten die ersten Kunden die neue Softwaregeneration SASKIA.H2R.*

### In jeder Hinsicht zukunftssicher

Für die Zukunft kann Köhler sich weiterhin vorstellen, dass etwa für die Inventarisierung oder als Kassenterminal auch einmal Mobilgeräte zum Einsatz kommen, die er mit der Web-Oberfläche nun direkt bedienen kann. Selbst Linux-Arbeitsplätze will er langfristig nicht ausschließen. Am allerwichtigsten für ihn ist jedoch, dass er mit seinem Team in gewohnter Produktivität weiterentwickeln kann. „Gemeinsam mit fecher haben wir eine moderne Lösung geschaffen, ohne uns mit den Komplexitäten der Web-Entwicklung herumschlagen zu müssen“, zieht er zufrieden Bilanz.

#### IMPRESSUM & KONTAKT

**Herausgeber:** fecher GmbH  
 Otto-Lilienthal-Str. 12, 63322 Rödermark, Germany  
**Telefon:** +49 6074 80577-00  
**E-Mail:** info@fecher.eu  
**Web:** www.modernizing-applications.de  
**Geschäftsführer:** Günter Hofmann, Andreas Glomm

**fecher.**