

---

## Hilfe zur Selbsthilfe: Gupta-Anwendungen nach .NET gebracht

### Kooperatives „in-house“-Portierungsprojekt von fecher macht 25 Jahre alte Gupta-Anwendungslösungen von Lang Finanzsoftware fit für die kommenden Jahrzehnte

---

Finanzierungsverwaltung und -kontrolle, Cash-Management, Liquiditätsplanung sowie Wertpapier- und Trading-Programme - hier liegen seit über 25 Jahren die Kernkompetenzen der Lang Finanzsoftware GmbH im oberösterreichischen Freistadt. Vor allem Wohnungsbaugenossenschaften, Immobilienunternehmen und Banken, aber auch Kunden aus der Privatwirtschaft im ganzen deutschsprachigen Raum setzen auf den CASH MANAGER und den KREDIT MANAGER, die dieses Know-how in erfolgreiche Applikationen gießen. Während die Software in zweieinhalb Jahrzehnten permanent weiterentwickelt wurde, war die technische Basis - der Gupta Team Developer - mittlerweile allerdings in die Jahre gekommen. CEO Michael Riesner und Software Developer Simon Voggeneder berichten, wie sie mit einem kooperativen „in-house“-Porting-Projekt von fecher ihre Anwendung auf .NET umstellen und damit fit für die kommenden Jahrzehnte machen konnten.

#### **Wo lagen denn die Probleme mit der alten Gupta-Plattform?**

**Riesner:** Ich will unsere frühere Plattform gar nicht schlecht reden, sie hat uns ja jahrzehntelang gute Dienste geleistet. Aber mal angenommen, ich befinde mich in einer Kongresshalle mit eintausend Softwareentwicklern und ich bitte alle aufzuzeigen, die Gupta Team-Developer programmieren können. Dann werden da fünf bis sechs Hände hochgehen. Wenn ich aber die Frage stelle, wer C# entwickeln kann, dann wird wahrscheinlich die Hälfte der Anwesenden aufzeigen, wenn nicht mehr. Das war für uns das Ausschlaggebende: Wir wollen ein attraktiverer Arbeitgeber für unser Entwicklungsteam sein und uns leichter damit tun, neue Entwickler am freien Markt zu rekrutieren.



Hauptprodukt von Lang Finanzsoftware ist der KREDIT MANAGER

#### **Welche Optionen für eine Portierung kamen in Frage und wie haben Sie sich letztlich entschieden?**

**Voggeneder:** Wir verfügen ja über eine sehr gute Basis, haben mit einem kleinen Team immer schon viel erreicht. Das wollten wir erhalten und in die neue Umgebung mitnehmen, ohne alles neu schreiben zu müssen. Nachdem wir bereits Applikationsteile mit der C#-Brücke implementiert und in die Gupta-Anwendung eingebunden hatten, war für uns C# gegenüber Java der deutliche Favorit. Außerdem waren und sind wir, ebenso wie beinahe alle unsere Kunden, ohnehin bereits im Microsoft-Universum zuhause.

**Riesner:** Dabei war uns klar, dass wir externe Unterstützung hinzuholen wollten. Ein Name, der dafür in der Gupta-Szene immer wieder gefallen ist, war fecher. Wenn wir mit Entwicklerkolleginnen und -kollegen gesprochen haben, hieß es überall, dass dort das größte Know-how und die besten Werkzeuge vorhanden wären.

Allerdings wollten wir das Portierungsprojekt nicht ganz aus der Hand geben. Für andere Kunden mag eine schlüsselfertige Gupta-Portierung ideal sein, aber wir haben uns ganz bewusst für das kooperative Modell „in-house“ entschieden. Dabei sollte fecher nur die grundlegende Portierung bis zum kompilierbaren Code vornehmen und uns dann dabei anleiten, die restlichen Schritte selbst durchzuführen. Bei aller Komplexität, die in unseren Applikationen steckt, wollten wir von vornherein sicherstellen, dass wir den portierten Code wirklich beherrschen, und das dazu notwendige Wissen von Anfang an intern aufbauen.



*Michael Riesner,  
CEO der Lang Finanzsoftware GmbH*

### **Wann und wie ist das Projekt dann gestartet?**

**Riesner:** Die ersten Kontakte waren im Oktober 2018, von da an ging alles ganz schnell. Wir haben den Quellcode unserer drei kommerziellen Produkte, CASH MANAGER, KREDIT MANAGER und ASSET MANAGER, an fecher gegeben. Im Januar 2019 kam direkt die erste Version des portierten Codes zurück.



*Simon Voggeneder, Software Developer  
bei Lang Finanzsoftware GmbH*

**Voggeneder:** Damit fing die Arbeit für uns an. Beginnend mit der für uns wichtigsten Applikation, dem KREDIT MANAGER, haben wir die nächsten sechs Monate damit verbracht, alles durchzutesten und die noch erforderlichen Anpassungen vorzunehmen. Wir wollten ja möglichst schnell durchkommen, um die alte Gupta-Version nicht länger als unbedingt notwendig parallel weiterführen zu müssen. Notfalls hätten wir das sicherlich gemacht, aber natürlich war das nicht Sinn und Zweck der Sache.

### **Und wieviel Nacharbeit war am Ende tatsächlich nötig?**

**Voggeneder:** fecher hatte die Software wie vereinbart mit seinen Werkzeugen portiert und so weit fertig gestellt, dass der entstandene C#-Code sich fehlerfrei kompilieren ließ. Wir konnten die Anwendungen also grundsätzlich starten und es war dann an uns, die Oberfläche nachzubearbeiten, wo sie sich nicht pixelperfekt hatte übertragen lassen. Meist ging es um Kleinigkeiten: Schriftarten anpassen, Eingabefelder verschieben, die Größe von Buttons justieren. Im Einzelnen war das nicht der große Aufwand, aber bei weit mehr als einer Million Lines of Code hat es sich zusammensummiert, bis alles wieder so aussah, wie es sollte.

Größere Probleme hatten wir eigentlich nur mit einem Dialog im KREDIT MANAGER, der einen sehr komplexen Aufbau hat. Mit überlappenden Feldern, die dynamisch ein- und ausgeblendet werden, integrierten Navbars und ähnlichen Elementen hat er nach der Portierung überhaupt nicht ausgesehen, wie er eigentlich sollte. Da waren wir sehr froh, dass fecher uns tatkräftig bei der Finalisierung unterstützt und die Layout-Logik so für uns angepasst hat, dass alles zum größten Teil wieder passte.

**Riesner:** Dieser Dialog ist in unserer Applikation aber auch wirklich die „eierlegende Wollmilchsau“, in der ganz viel Komplexität drinsteckt. Nachdem wir das in Kooperation mit fecher lösen konnten, funktioniert die Anwendung in ergonomischer Hinsicht besser als zuvor, muss man sagen.

*„fecher hat uns bei der Finalisierung tatkräftig unterstützt.“*

Ebenfalls einige Nacharbeit erforderten die gut 200 Auswertungen, die wir ursprünglich mit dem Gupta Report Builder erstellt hatten. fecher hatte mehrere Reportingwerkzeuge für .NET vorgeschlagen, in die automatisiert portiert werden konnte, und unsere Wahl war schließlich auf Crystal Reports gefallen. Da lief es ähnlich wie mit den Dialogen: Die Portierung hat grundsätzlich sehr gut funktioniert, und überall da, wo eine Formatierung mal nicht so ausgeschaut hat wie vorher, haben wir uns in der Finalisierung um den letzten Feinschliff gekümmert. Sicher, es war viel Aufwand, aber wir hatten uns ja bewusst für den Projekttyp „in-house“ entschieden, um die Applikationen selbst so zu finalisieren, wie wir es uns vorgestellt haben.

Diese verschiedenen Nacharbeiten, bei denen ja immer irgendwo etwas zu implementieren oder am Code zu ändern war, hat natürlich unsere Entwicklung machen müssen. Aber den ganzen Aufwand für die Tests der Software, die vorher und nachher stattfanden, haben wir im Unternehmen aufgeteilt. Also das Testen, ob das richtige Ergebnis herauskommt, das Überprüfen, wie die Dialoge aussehen und ob die Texte alle auch lesbar sind, da hat jeder seinen Part dazu beigetragen.

### **Wie lange hat es gedauert, bis die Software an die Kunden ausgeliefert wurde, und wie war die Resonanz?**

**Voggeneder:** Nach einem knappen halben Jahr waren wir mit allen drei Anwendungen fertig. Nachdem wir sicher waren, dass die grafische Benutzeroberfläche perfekt zu benutzen ist, alle Reports funktionieren und die Qualitätssicherung auch sonst keine Fehler mehr gefunden hat, haben wir an die ersten Kunden ausgeliefert. Das war im Oktober 2019, also circa zehn Monate nach Beginn des Projekts. Was uns sofort aufgefallen ist, war die höhere Performance bei Datenbankzugriffen, dass das Laden von Daten oft deutlich schneller geht. Wenn wir zum Beispiel einen Kunden in der Software neu einrichten, wird bei der Installation das ganze Datenbankschema angelegt. Das dauerte auf einmal nur mehr ein Zehntel so lang wie früher, das war beeindruckend!

**Riesner:** Alle Kunden, die bei uns einen Wartungsvertrag besitzen, haben die Applikation kostenfrei im portierten Zustand erhalten. Das ist einfach die nächste Softwaregeneration geworden, und die Anwenderinnen und Anwender haben auch gleich verschiedene Vorteile bemerkt, die sie mit der alten Gupta-Version nicht hatten. Zum Beispiel ist durch die native .NET-Integration die Ansteuerung von Office-Produkten viel komfortabler geworden ist, ohne extra Umwege bei Excel-Tabellen gehen zu müssen.

Nicht zuletzt ist das Design schöner und zeitgemäßer, das ist ja auch nicht unwesentlich. Unsere Applikationen sind sehr listen- und tabellenorientiert, das lässt sich nicht einfach im Kacheldesign abbilden. Mit C# und Visual Studio bestehen aber zahlreiche Möglichkeiten, auch umfangreiche Funktionalität optisch ansprechend zu präsentieren. Und es gibt eine riesige Auswahl an kommerziell verfügbaren Bibliotheken, so dass man nicht alles selbst machen muss.

### ***Die neue C#-Version hat also auch Vorteile für Ihre Entwicklung?***

**Voggeneder:** Unbedingt. Heute entwickeln wir wesentlich komfortabler und haben erheblich bessere Möglichkeiten etwa beim Debugging. Durch die Portierung hat sich uns die ganze .NET-Welt erschlossen mit all ihren Programmkomponenten, die vorher nicht wirklich zur Verfügung standen. Endlich sind wir mit unserer Entwicklungsumgebung *state of the art*, angefangen davon, wie der Code aussieht, bis zur Frage, in welchen Foren man sich dazu austauschen kann oder wo man Entwickler dafür findet. Im .NET-Universum gibt es ja auch diese NuGet-Pakete, mit denen einem alle möglichen Code-Fragmente zur Verfügung stehen. Das hilft immens, wir konnten dadurch viel einfacher Projekte umsetzen, in denen wir mit Webservices kommunizieren. Auch für Kryptografie-Themen stehen uns jetzt ganz neue Möglichkeiten zur Verfügung.

Nicht zuletzt hat sich als großer Mehrwert herausgestellt, dass die Applikationen nicht mehr der riesige Monolith sind, der sie früher waren. Zum Beispiel war der KREDIT MANAGER, unsere Hauptanwendung, im Grunde eine einzige große Datei. Im Zuge der Portierung haben wir das alles in Funktionsbausteine aufgeteilt und es sind vermehrt einzelne Klassen und einzelne, ausgelagerte Dialoge entstanden. So haben wir weniger Gefahr irgendwelcher Konflikte und können viel genauer nachverfolgen, wo im Code tatsächlich Änderungen passiert sind.

### ***Bei aller Begeisterung: Was hätten Sie im Rückblick anders gemacht?***

**Riesner:** In einer Hinsicht haben wir bereits nachkorrigiert. Nachdem die Kunden-Applikationen erfolgreich portiert waren, haben wir uns entschieden, auch noch den Kundenmanager, unser internes, selbstentwickeltes CRM-System, ebenfalls mit Unterstützung von fecher zu portieren. Ursprünglich war das gar nicht vorgesehen. Wir haben es aber zum Jahreswechsel 2020/21 durchgezogen, weil es uns durch den Produktivitätsgewinn eine erhebliche Ersparnis bei der Weiterentwicklung bringt.

*„Mit unserem heutigen Wissen würden wir die Sache früher angehen.“*

Definitiv bewährt hat sich die Kooperation mit fecher. Die war von Anfang an sehr partnerschaftlich und immer auf Augenhöhe. Wenn wir noch eine Gupta-Anwendung hätten, würden wir sie wieder von fecher portieren lassen.

**Voggeneder:** Das Einzige, was wir sicher anders machen würden: Wir würden die Portierung bereits viel früher angehen.

**Riesner:** Dem kann ich nur beipflichten.

#### *Impressum & Kontakt:*

Herausgeber: fecher GmbH  
Otto-Lilienthal-Str. 12  
D-63322 Rödermark

Telefon: +49 6074 80577-00  
Telefax: +49 6074 80577-99

E-Mail: [info@fecher.eu](mailto:info@fecher.eu)  
Website: [www.fecher.eu](http://www.fecher.eu)  
Geschäftsführer: Günter Hofmann