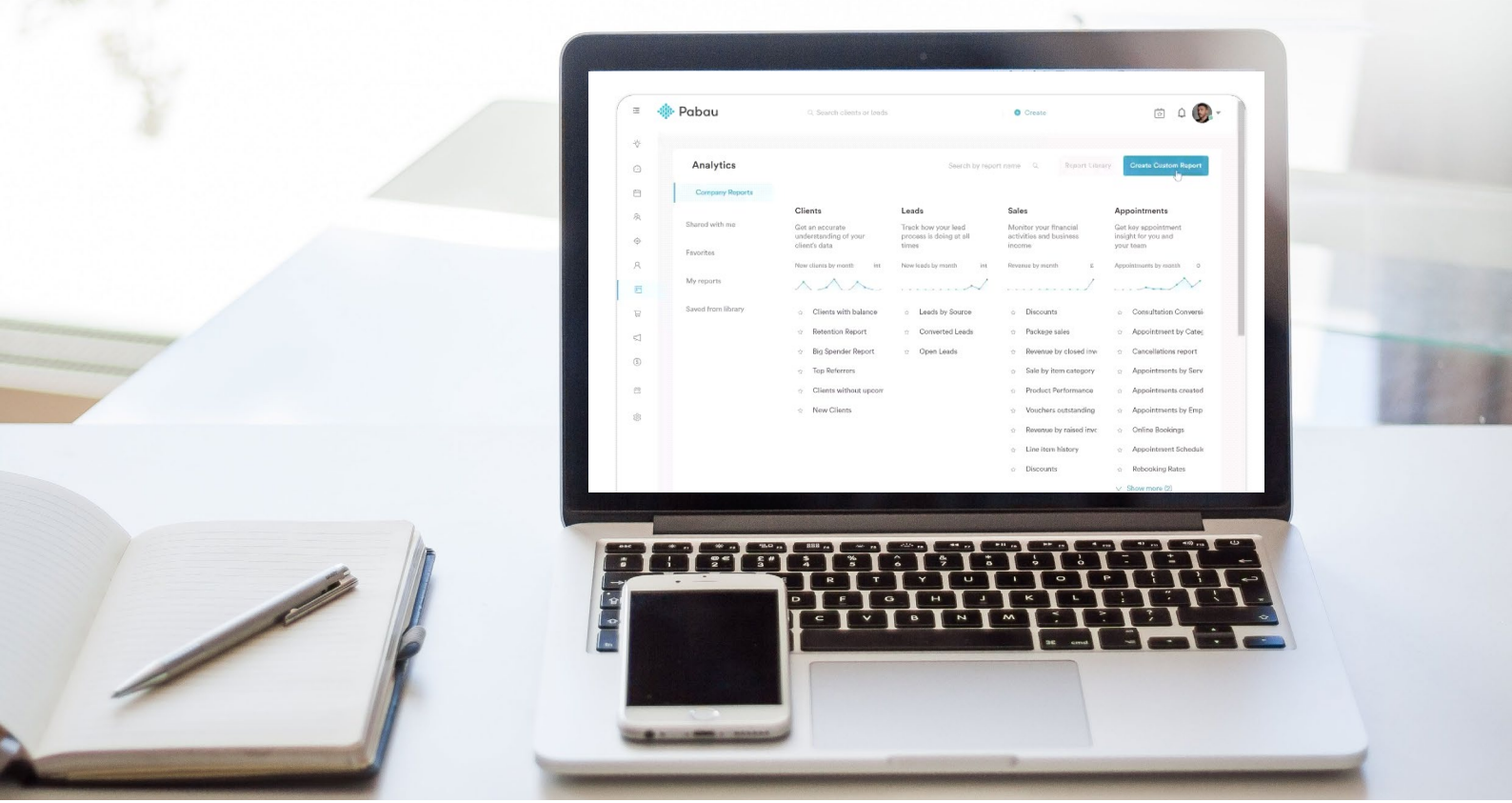


# Support for a self-service platform built on top of Google Cloud

```
File Edit View Navigate Code Refactor Build Run Tests Tools VCS Window Help
403-880A2-FB667D0ADC4 30 8904214 C:\Maths C:\Maths C:\Debug\AnyCPU
C# ReferenceImageGrabBehaviour.cs C# TakeScreenshot.cs C# Maths.cs C# DiagonalGrid.cs
1 // <summary>
2 // <para>Returns the closest power of two value.</para>
3 // </summary>
4 // <param name="value"></param>
5 [return:NotNullIfNotNull(nameof(value))]
6 [MethodImpl(MethodImplOptions.AggressiveInlining)]
7 public static extern int ClosestPowerOfTwo(int value);
8
9 // <summary>
10 // <para>Returns true if the value is power of two.</para>
11 // </summary>
12 // <param name="value"></param>
13 [return:NotNullIfNotNull(nameof(value))]
14 [MethodImpl(MethodImplOptions.AggressiveInlining)]
15 public static extern bool IsPowerOfTwo(int value);
16
17 // <summary>
18 // <para>Returns the next power of two that is equal to, or greater than, the value.</para>
19 // </summary>
20 // <param name="value"></param>
21 [return:NotNullIfNotNull(nameof(value))]
22 [MethodImpl(MethodImplOptions.AggressiveInlining)]
23 public static extern int NextPowerOfTwo(int value);
24
25 // <summary>
26 // <para>Returns the given value from game (1000) to linear color level (0-1000).</para>
27 // </summary>
28 // <param name="value"></param>
29 [return:NotNullIfNotNull(nameof(value))]
30 [MethodImpl(MethodImplOptions.AggressiveInlining)]
31 public static extern float GameToLinearColorLevel(float value);
32
33 // <summary>
34 // <para>Converts the given value from linear to game (1000) color level (0-1000).</para>
35 // </summary>
36 // <param name="value"></param>
37 [return:NotNullIfNotNull(nameof(value))]
38 [MethodImpl(MethodImplOptions.AggressiveInlining)]
39 public static extern float LinearToGameColorLevel(float value);
40
41 // <summary>
42 // <para>Convert a color temperature in kelvins to RGB color.</para>
43 // </summary>
44 // <param name="kelvin">Temperature in kelvin. Range 5000 to 10000 kelvin.</param>
45 // <param name="saturation">Saturation of resulting color. RGB color (0-1000).</param>
46 // <param name="gamma">Gamma correction of resulting color. RGB color (0-1000).</param>
47 [return:NotNullIfNotNull(nameof(kelvin))]
48 [MethodImpl(MethodImplOptions.AggressiveInlining)]
49 public static extern Color GameColorToRGB(float temperatureInKelvin, float saturation, float gamma);
50
51 // <summary>
52 // <para>Converts a color temperature in kelvins to HSV color.</para>
53 // </summary>
54 // <param name="kelvin">Temperature in kelvin. Range 5000 to 10000 kelvin.</param>
55 // <param name="saturation">Saturation of resulting color. RGB color (0-1000).</param>
56 // <param name="gamma">Gamma correction of resulting color. RGB color (0-1000).</param>
57 [return:NotNullIfNotNull(nameof(kelvin))]
58 [MethodImpl(MethodImplOptions.AggressiveInlining)]
59 public static extern Color GameColorToHSV(float temperatureInKelvin, float saturation, float gamma);
60
61 // <summary>
62 // <para>Converts a HSV color to RGB color.</para>
63 // </summary>
64 // <param name="hue">Hue in degrees of circle (0-360).</param>
65 // <param name="saturation">Saturation of resulting color. RGB color (0-1000).</param>
66 // <param name="value">Value of resulting color. RGB color (0-1000).</param>
67 [return:NotNullIfNotNull(nameof(hue))]
68 [MethodImpl(MethodImplOptions.AggressiveInlining)]
69 public static extern Color HSVToGameColor(float hue, float saturation, float value);
70
71 // <summary>
72 // <para>Converts a RGB color to HSV color.</para>
73 // </summary>
74 // <param name="red">Red component of color (0-1000).</param>
75 // <param name="green">Green component of color (0-1000).</param>
76 // <param name="blue">Blue component of color (0-1000).</param>
77 [return:NotNullIfNotNull(nameof(red))]
78 [MethodImpl(MethodImplOptions.AggressiveInlining)]
79 public static extern Color RGBToGameColor(int red, int green, int blue);
80
81 // <summary>
82 // <para>Converts a HSV color to RGB color.</para>
83 // </summary>
84 // <param name="hue">Hue in degrees of circle (0-360).</param>
85 // <param name="saturation">Saturation of resulting color. RGB color (0-1000).</param>
86 // <param name="value">Value of resulting color. RGB color (0-1000).</param>
87 [return:NotNullIfNotNull(nameof(hue))]
88 [MethodImpl(MethodImplOptions.AggressiveInlining)]
89 public static extern Color HSVToGameColor(float hue, float saturation, float value);
90
91 // <summary>
92 // <para>Converts a RGB color to HSV color.</para>
93 // </summary>
94 // <param name="red">Red component of color (0-1000).</param>
95 // <param name="green">Green component of color (0-1000).</param>
96 // <param name="blue">Blue component of color (0-1000).</param>
97 [return:NotNullIfNotNull(nameof(red))]
98 [MethodImpl(MethodImplOptions.AggressiveInlining)]
99 public static extern Color RGBToGameColor(int red, int green, int blue);
100
101 // <summary>
102 // <para>Returns the sine of angle f in radians.</para>
103 // </summary>
104 // <param name="f">The input angle, in radians.</param>
105 // </returns>
106 // <para>The return value between -1 and 1.</para>
107 [return:NotNullIfNotNull(nameof(f))]
108 [MethodImpl(MethodImplOptions.AggressiveInlining)]
109 public static float Sin(float f);
110 {
111     return (float) Math.Sin((double) f);
112 }
113
114 // <summary>
115 // <para>Returns the cosine of angle f in radians.</para>
116 // </summary>
117 // <param name="f">The input angle, in radians.</param>
118 // </returns>
119 // <para>The return value between -1 and 1.</para>
120 [return:NotNullIfNotNull(nameof(f))]
121 [MethodImpl(MethodImplOptions.AggressiveInlining)]
122 public static float Cos(float f);
123 {
124     return (float) Math.Cos((double) f);
125 }
126
127 // <summary>
128 // <para>Returns the tangent of angle f in radians.</para>
129 // </summary>
130 // <param name="f">The input angle, in radians.</param>
131 // </returns>
132 // <para>The return value between -1 and 1.</para>
133 [return:NotNullIfNotNull(nameof(f))]
134 [MethodImpl(MethodImplOptions.AggressiveInlining)]
135 public static float Tan(float f);
136 {
137     return (float) Math.Tan((double) f);
138 }
139
140 // <summary>
141 // <para>Returns the arc-sine of f - the angle in radians whose sine is f.</para>
142 // </summary>
143 // <param name="f">The input value.</param>
144 // </returns>
145 // <para>The return value between -1 and 1.</para>
146 [return:NotNullIfNotNull(nameof(f))]
147 [MethodImpl(MethodImplOptions.AggressiveInlining)]
148 public static float Asin(float f);
149 {
150     return (float) Math.Asin((double) f);
151 }
152
153 // <summary>
154 // <para>Returns the arc-cosine of f - the angle in radians whose cosine is f.</para>
155 // </summary>
156 // <param name="f">The input value.</param>
157 // </returns>
158 // <para>The return value between -1 and 1.</para>
159 [return:NotNullIfNotNull(nameof(f))]
160 [MethodImpl(MethodImplOptions.AggressiveInlining)]
161 public static float Acos(float f);
162 {
163     return (float) Math.Acos((double) f);
164 }
165
166 // <summary>
167 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
168 // </summary>
169 // <param name="f">The input value.</param>
170 // </returns>
171 // <para>The return value between -1 and 1.</para>
172 [return:NotNullIfNotNull(nameof(f))]
173 [MethodImpl(MethodImplOptions.AggressiveInlining)]
174 public static float Atan(float f);
175 {
176     return (float) Math.Atan((double) f);
177 }
178
179 // <summary>
180 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
181 // </summary>
182 // <param name="y">The y-coordinate of the point.</param>
183 // <param name="x">The x-coordinate of the point.</param>
184 [return:NotNullIfNotNull(nameof(y))]
185 [MethodImpl(MethodImplOptions.AggressiveInlining)]
186 public static float Atan2(float y, float x);
187 {
188     return (float) Math.Atan2((double) y, (double) x);
189 }
190
191 // <summary>
192 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
193 // </summary>
194 // <param name="y">The y-coordinate of the point.</param>
195 // <param name="x">The x-coordinate of the point.</param>
196 [return:NotNullIfNotNull(nameof(y))]
197 [MethodImpl(MethodImplOptions.AggressiveInlining)]
198 public static float Atan2(float y, float x);
199 {
200     return (float) Math.Atan2((double) y, (double) x);
201 }
202
203 // <summary>
204 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
205 // </summary>
206 // <param name="y">The y-coordinate of the point.</param>
207 // <param name="x">The x-coordinate of the point.</param>
208 [return:NotNullIfNotNull(nameof(y))]
209 [MethodImpl(MethodImplOptions.AggressiveInlining)]
210 public static float Atan2(float y, float x);
211 {
212     return (float) Math.Atan2((double) y, (double) x);
213 }
214
215 // <summary>
216 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
217 // </summary>
218 // <param name="y">The y-coordinate of the point.</param>
219 // <param name="x">The x-coordinate of the point.</param>
220 [return:NotNullIfNotNull(nameof(y))]
221 [MethodImpl(MethodImplOptions.AggressiveInlining)]
222 public static float Atan2(float y, float x);
223 {
224     return (float) Math.Atan2((double) y, (double) x);
225 }
226
227 // <summary>
228 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
229 // </summary>
230 // <param name="y">The y-coordinate of the point.</param>
231 // <param name="x">The x-coordinate of the point.</param>
232 [return:NotNullIfNotNull(nameof(y))]
233 [MethodImpl(MethodImplOptions.AggressiveInlining)]
234 public static float Atan2(float y, float x);
235 {
236     return (float) Math.Atan2((double) y, (double) x);
237 }
238
239 // <summary>
240 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
241 // </summary>
242 // <param name="y">The y-coordinate of the point.</param>
243 // <param name="x">The x-coordinate of the point.</param>
244 [return:NotNullIfNotNull(nameof(y))]
245 [MethodImpl(MethodImplOptions.AggressiveInlining)]
246 public static float Atan2(float y, float x);
247 {
248     return (float) Math.Atan2((double) y, (double) x);
249 }
250
251 // <summary>
252 // <para>Returns the arc-tangent of f - the angle in radians whose tangent is f.</para>
253 // </summary>
254 // <param name="y">The y-coordinate of the point.</param>
```

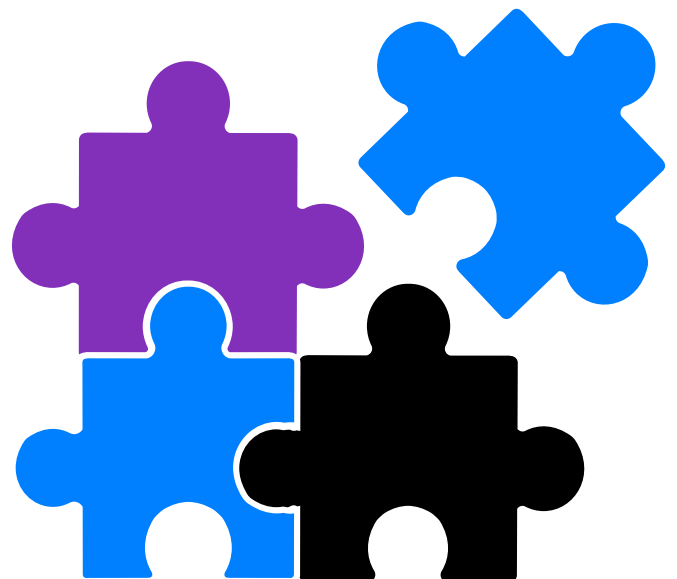


# Customer

A large retail corporation in Vienna, Austria, embarked on a groundbreaking initiative to enhance their development processes. The project's essence was encapsulated in a powerful goal: the creation of a self-service platform built on top of Google Cloud. This platform empowered their developers to seamlessly create, deploy, and monitor applications on Google Cloud. The overarching objective was to dramatically reduce existing application release cycles and minimize the time to market for new applications.

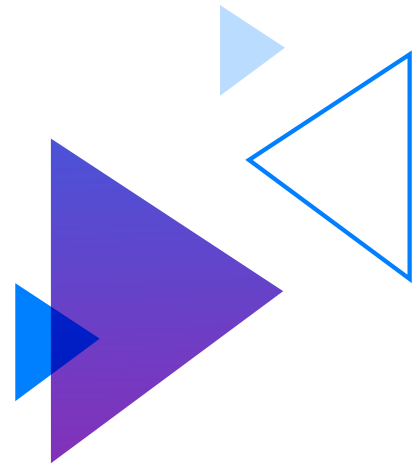
## Technologies:

- DevOps: Google Cloud, Kubernetes, Terraform, Kustomize



# Challenge

The challenges were formidable. The existing development processes needed a substantial overhaul to accommodate the envisioned self-service platform. An integration with Google Cloud presented both technical and logistical hurdles. Additionally, cultural shifts within the development teams were imperative to embrace this paradigm shift in application development, moving towards a more agile and streamlined approach.

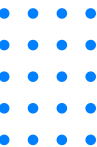


# Solution

In response to the challenges, our team devised a comprehensive solution. The self-service platform was crafted to align with Google Cloud, ensuring optimal performance and efficiency. A multifaceted approach was adopted to address the cultural shift, including tailored training programs and change management strategies. This solution aimed not only at technological integration but also at fostering a mindset change within the client's development teams. By emphasizing a collaborative and adaptive mindset, our approach aimed to empower the client's teams to embrace innovation and navigate future challenges with resilience and agility.



We worked in partnership with WHALEY on a project and we had a huge success. Their team is very proactive and knowledgeable and together we managed to exceed the client's expectations. WHALEY's team revolutionized the development workflows on Google Cloud resulting with sustained cost savings. Their customer experience is something out of the ordinary, we defiantly plan WHALEY as a partner in our future projects

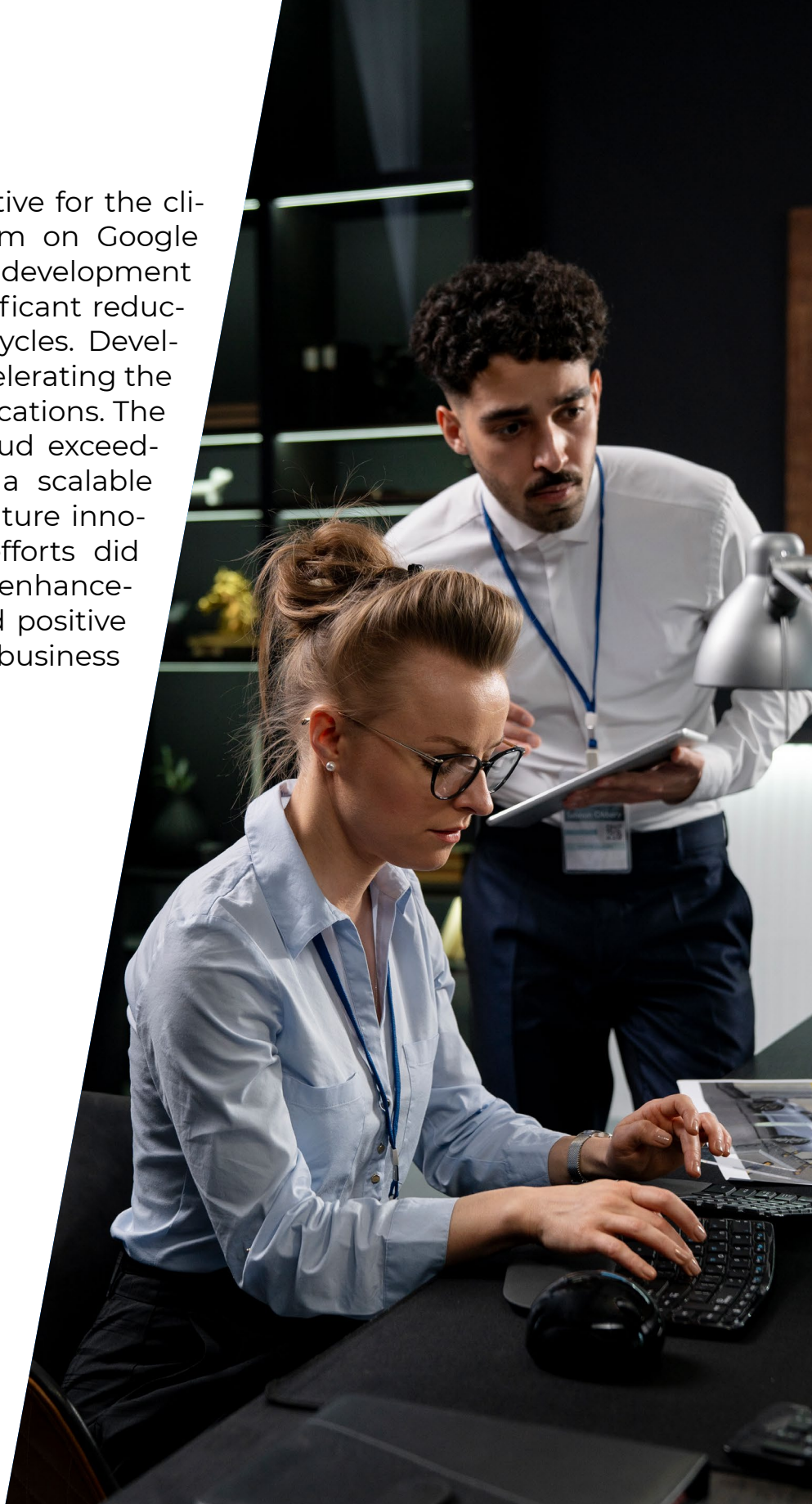


**Damjan Gjuroski**  
*Head of Technology at Posedio*



# Result

The results were transformative for the client. The self-service platform on Google Cloud revolutionized their development workflows, leading to a significant reduction in application release cycles. Developers gained autonomy, accelerating the time to market for new applications. The integration with Google Cloud exceeded expectations, providing a scalable and robust foundation for future innovations. The collaborative efforts did not just result in operational enhancements but made a profound positive impact on the client's overall business trajectory.



**Contact us now to discuss  
your business case**