Anforderungsdefinition

Fertigstellung Prototypversion von CWF, Assistant und XClient zur Demonstration mit den Türmen von Hanoi



Neutorstraße 13  
5020 Salzburg

AUSTRIA

Tel: +43 (662) 276198-11

Fax: +43 (662) 276198-98

Mail: [office@breanos.com](mailto:office@breanos.com)

|  |  |
| --- | --- |
|  |  |
| Datum | 20.09.2018 |

Inhaltsverzeichnis

[1 Änderungsverzeichnis 4](#_Toc525115367)

[2 Anforderungsbeschreibung 5](#_Toc525115368)

[2.1 Referenzen 5](#_Toc525115369)

[2.2 Einführung 5](#_Toc525115370)

[2.2.1 Inkludierter Umfang 5](#_Toc525115371)

[2.2.1.1 Struktur des Assistant 5](#_Toc525115372)

[2.2.1.2 Schnittstellenbeschreibung KPUs zur Registrierung beim Assistant 5](#_Toc525115373)

[2.2.1.3 Kommunikation zwischen Client und Assistant zum Abholen aller für die Anzeige relevanter Daten 5](#_Toc525115374)

[2.2.2 Exkludierter Umfang 5](#_Toc525115375)

[2.3 Geplantes Vorgehen 6](#_Toc525115376)

[2.4 Aufwand 6](#_Toc525115377)

[2.5 Risiken 6](#_Toc525115378)

[2.5.1 Risiko 1 6](#_Toc525115379)

[2.5.1.1 Beschreibung 6](#_Toc525115380)

[2.5.1.2 Folgen 6](#_Toc525115381)

[2.5.1.3 Risikoeinschätzung 6](#_Toc525115382)

# Änderungsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Verfasser | Beschreibung |
| 19.09.2018 | 1 | Franz Kellner, Breanos Software | Dokument erstellt |

# Anforderungsbeschreibung

## Referenzen

UML in Breanos 🡪 Daipan 🡪 Biks 🡪 Architecture 🡪 Xamarin Client - View&Data retrieval 🡪 Activities

## Einführung

Die folgenden Punkte werden in diesem Dokument hinsichtlich der erforderlichen Arbeitsumfänge beschrieben.

* Struktur des Assistant,
* Schnittstellenbeschreibung KPUs zur Registrierung beim Assistant
* Kommunikation zwischen Client und Assistant zum Abholen aller für die Anzeige relevanter Daten

### Inkludierter Umfang

#### Struktur des Assistant

Der Assistant besitzt derzeit einige Komponenten die hardcoded Modelle / Controller erstellen bzw. beinhalten. Die Endversion des Assistant soll keine fix eingebauten Modelle beinhalten und gänzlich frei von Controllern sein, da diese in den jeweiligen KPUs verbaut sind.

#### Schnittstellenbeschreibung KPUs zur Registrierung beim Assistant

Zur Initiierung der Kommunikation mit dem Assistant muss sich eine KPU registrieren. Der Vorgang wird über das Blackboard abgewickelt, welches die entsprechenden Telegramme weiterleitet.

Das Telegramm zum Registrieren einer KPU könnte dabei wie folgt aussehen:

* Eindeutiger Identifier der KPU
* Liste der von dem Datenmodell der KPU benötigten Assemblies
* Die benötigten Assemblies selbst als byte-Arrays
* Liste der von der View benötigten Dateien zum anzeigen (z.B. Bilder)
* Die mit der KPU verknüpften Views als Paar aus View Identifier und XAML-Text

#### Kommunikation zwischen Client und Assistant zum Abholen aller für die Anzeige relevanter Daten

Zwischen XClient und dem ExternalCommunication-Service des Assistant werden über SignalR-Methodenaufrufe die einzelnen Anfragen verschickt.

Derzeit sieht diese Kommunikation nur vor, sich auf eine View zu registrieren, was zur Folge hat, dass der Client den XAML-Text zum Anzeigen und die derzeitigen Werte des Models geschickt bekommt.

Das Verhalten muss entsprechend dem Aktivitätsdiagramm (s. Referenzen) angepasst werden. Dies beinhaltet in Folge auch seitens des Clients (und des Presenters im Assistant) neue Funktionalität bezüglich des Hereinladens und der Instanziierung des von der KPU bereitgestellten Models, damit das XAML-Binding darauf einfacher handzuhaben ist, bei größerer möglicher Komplexität als es derzeit der Fall ist.

### Exkludierter Umfang

* In späterer Folge soll eine KPU auch die Möglichkeit zur Deregistrierung haben
* Es gibt derzeit noch kein endgültiges Konzept für Mehrfachbeziehungen zwischen KPUs und Views.

## Geplantes Vorgehen

EBE: Schnittstellen KPU-Registrierung KPU – Blackboard & Blackboard – Assistant

FKE: Struktur Assistant & Kommunikation Assistant - XClient

## Aufwand

Der Aufwand wird wie folgt geschätzt.

|  |  |
| --- | --- |
| Aufgabe | Mannstunden |
| Schnittstellen zur KPU-Registrierung | 18 |
| Kommunikation XClient | 6 |
| Struktur Assistant | 6 |
| Backgroundfunktionalität XClient für die geänderte Kommunikation | 6 |
| Gesamt | 36 |

## Risiken

### Risiko 1

#### Beschreibung

Dynamische Assembly-Instanziierung könnte in .NET Standard komplexer sein als unter .NET Core. XClient ist in .NET Standard geschrieben.

#### Folgen

Die Fertigstellung der Backgroundfunktionalität zur Model-Persistierung im XClient könnte aufwändiger/schwieriger sein als erwartet.

#### Risikoeinschätzung

Es ist unwahrscheinlich, dass der Weg zur Instanziierung von Assemblies in .NET Standard wesentlich abgeändert zur Implementierung in .NET Core oder .NET Framework.