



Kanton Zug

# Fachschale für Erdwärmesondenbewilligungen


Ein Erfahrungsbericht zur Erstellung einer Fachschale mit Hilfe von  
PostGIS, QGIS und ili2pg

Manuel Kaufmann, QGIS-Anwendertreffen 2017

# Übersicht

1. Ausgangslage
2. Ziele des Fachamts und der GIS-Fachstelle
3. Inhaltliche und funktionelle Anforderungen
4. Datenmodell
5. Eingesetzte Tools (neben QGIS)
6. Anpassungen/Entwicklungen im QGIS-Kern
7. Custom Functions
8. Probleme/Stolpersteine
9. To do...

# 1. Ausgangslage

- Der Bau und die Änderung von **Erdwärmesonden** ist im Kanton ZG bewilligungspflichtig. Zuständig ist das Amt für Umweltschutz (GewG und V GewG).
- Die Erdwärmesonden sind ein kantonales Geobasisdatum (Anhang 2 GeoIV-ZG)
- Es existiert ein in die Jahre gekommenes WebGIS-Tool (basierend auf proprietärer SW) zur Erfassung und Verwaltung der Erdwärmesonden, das abgelöst werden muss.
- Weil das Tool nur wenige Personen (verwaltungsintern) nutzen, wurde entschieden, die Erdwärmesonden neu mit einem Desktop-GIS zu bewirtschaften → 

## 2. Ziele des Fachamts und der GIS-Fachstelle

- Fachamt
  - Neue inhaltliche Anforderungen (Anpassen des Datenmodells)
  - Funktionale Mängel des alten Tools beheben (z.B. flexiblere Anpassungsmöglichkeiten)
- GIS-Fachstelle
  - Modellierung in INTERLIS 2 (Domains als XML-Kataloge)
  - Keine Erfassung mehr von redundanten Informationen, die anderweitig vorhanden sind.
  - In QGIS nicht vorhandene Funktionalitäten sollen nicht über Plugins realisiert werden, sondern wenn möglich in den QGIS-Kern integriert werden (Aufträge an QGIS-Kernentwickler)

### 3. Inhaltliche und funktionale Anforderungen (1)

- Wichtigste Objekte
  - Anlagen (Punkteobjekte) mit zugehörigen Sondengruppen
  - Beteiligte (Eigentümer, Bohrfirma, Geologe)
  - Einschränkungen (Flächenobjekte), wie Geologie, Tiefenbeschränkungen, Bauten
  - Einbezug von Fremddatensätzen (Grundwasser-Schutzzonen, KbS usw.)
  - Zulässigkeitsgebiete (flächendeckende Aussage über die Zulässigkeit von Erdwärmesonden)

### 3. Inhaltliche und funktionale Anforderungen (2)

- Wichtigste funktionale Anforderungen
  - Erfassung der Objekte (mit Ausnahme der Fremddaten)
  - Aus anderen Daten eruierbare Attributewerte sollen nicht erfasst werden müssen (z.B. Gemeinde, Grundstück-Nr., Adresse)
  - Zulässigkeitsgebiete sollen automatisch aus den diversen flächenhaften Einschränkungen erzeugt werden.
  - Allfällige Auflagen und Empfehlungen für die Gesuchsteller bei der Realisierung der Anlagen sollen als Textbausteine verwaltet werden können
  - Diverse Suchmöglichkeiten



## 5. Eingesetzte Tools (neben QGIS)

### ili2pg

- Zur Erzeugung der DB-Struktur direkt aus dem INTERLIS-Modell
- Zum Import der XML-Kataloge

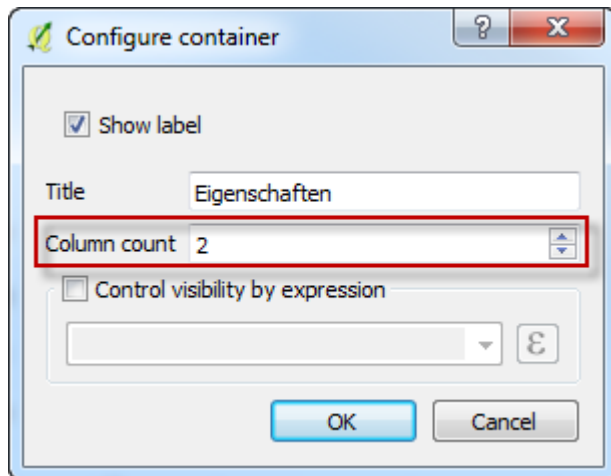
### xls2xtf

- Zur Erzeugung einer Excel-Vorlage (aus dem ILI-Modell) zur Verwaltung der Auswahllisten-Kataloge
- Zur Erzeugung des XTF-Files (XML-Kataloge)



## 6. Anpassungen/Entwicklungen im QGIS-Kern (1)

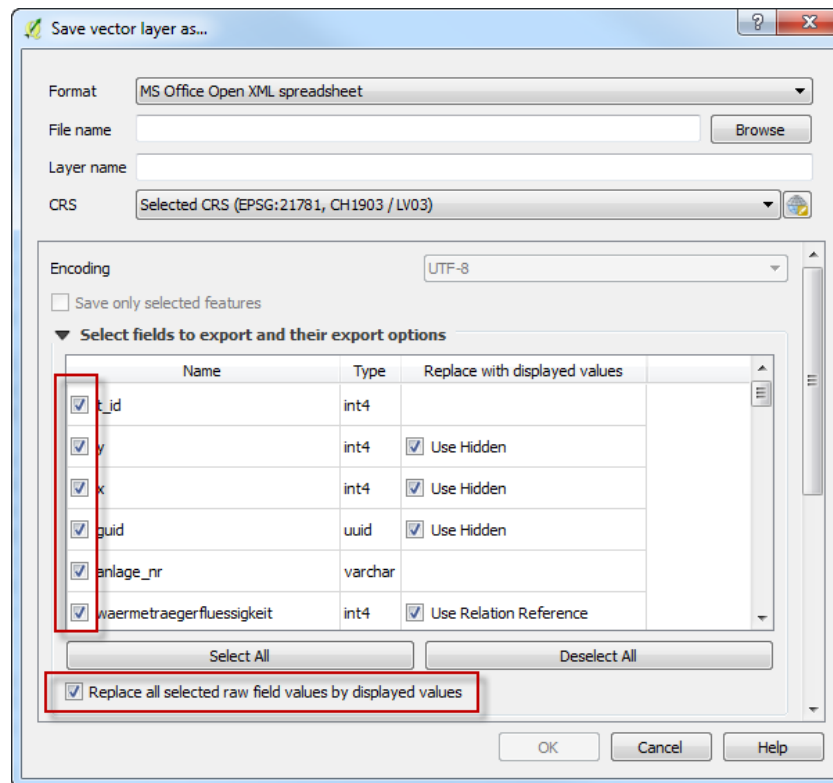
- Forms: Drag&Drop-Designer
  - Keine Tabs oder Group Container mehr notwendig zur Aufnahme von Widgets
  - Mehrere Spalten für die Anordnung der Felder möglich





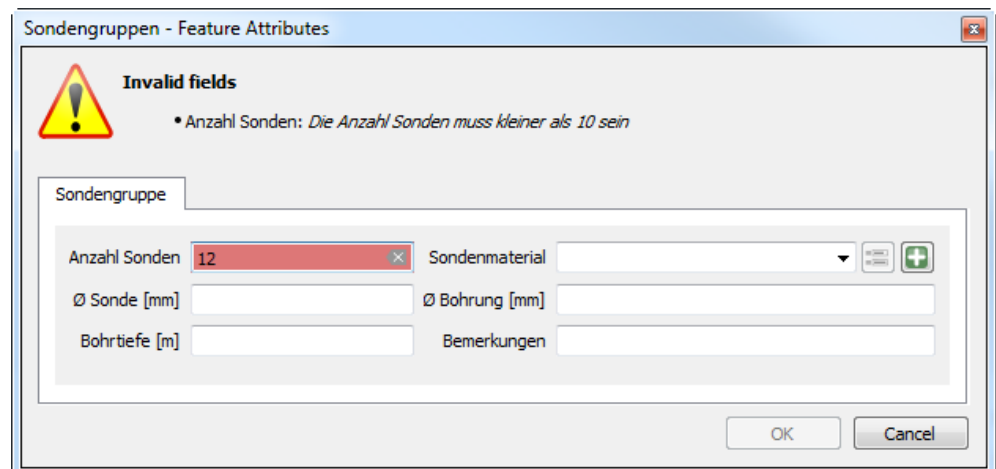
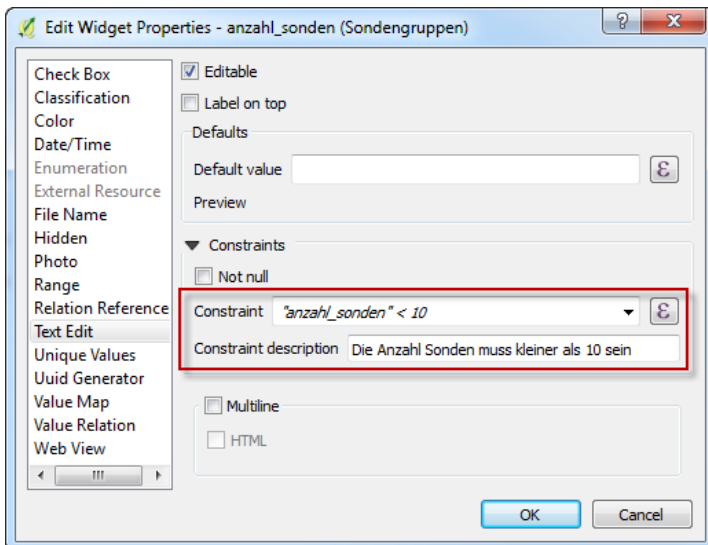
## 6. Anpassungen/Entwicklungen im QGIS-Kern (3)

- Datenexport
  - Wahl der Attribute, die exportiert werden sollen
  - Möglichkeit des "Auflösens" von verlinkten Auswahllistenwerten



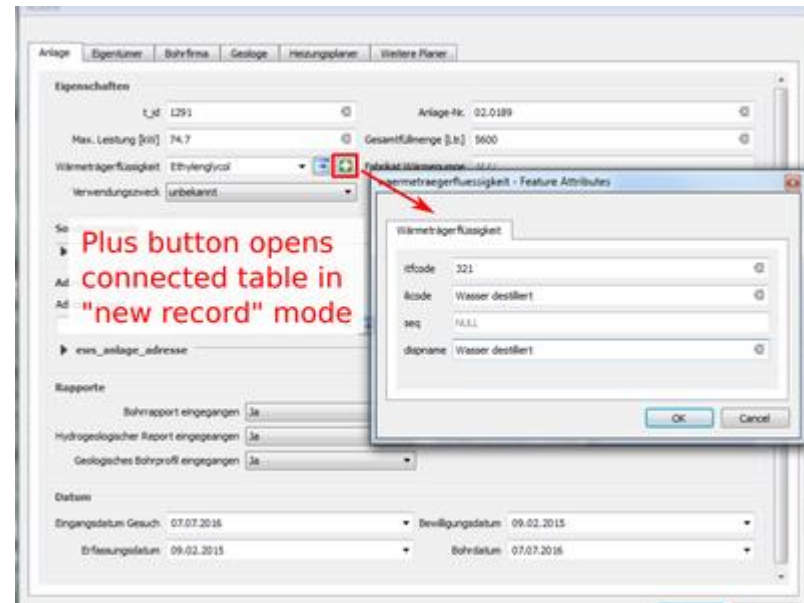
## 6. Anpassungen/Entwicklungen im QGIS-Kern (4)

- Constraints on widgets
  - Constraints für Attributwerte definierbar (mittels QGIS expressions);
  - Tooltip mit Constraint; Beschreibung des Constraints und des Resultats der Auswertung
  - Zusätzlich eine Meldungsbalken zuoberst im Formular



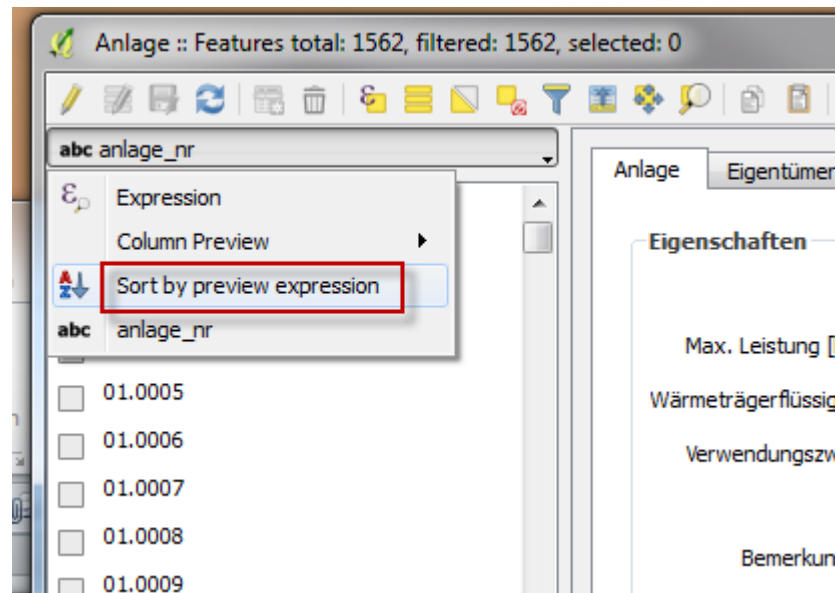
## 6. Anpassungen/Entwicklungen im QGIS-Kern (5)

- Relation reference widget:
  - Möglichkeit zur Erweiterung von verlinkten Auswahllisten direkt im Formular
  - Konfigurierbar in den Widget-Eigenschaften



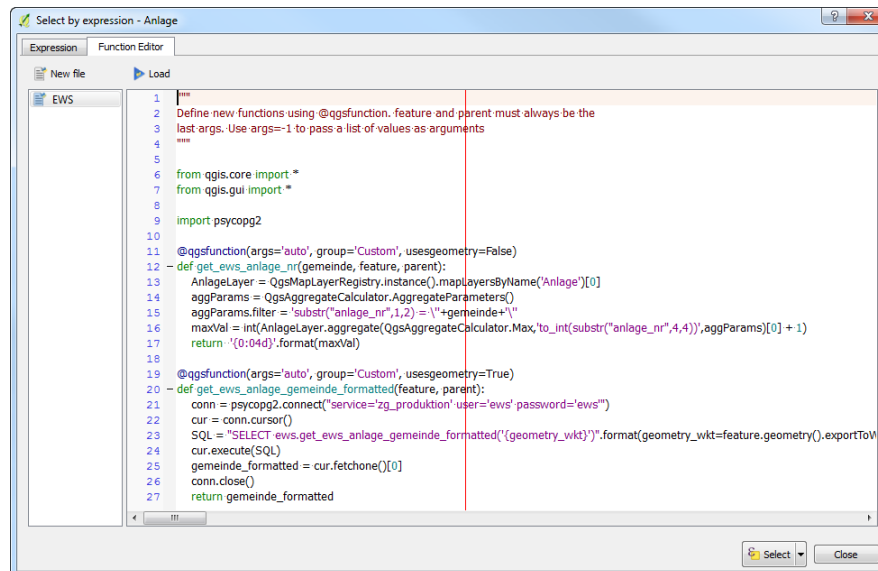
## 6. Anpassungen/Entwicklungen im QGIS-Kern (6)

- Attributtabelle, Formularansicht:
  - Die Objektliste auf der linken Seite in der Formularansicht ist nun sortierbar, ohne in den Tabellenmodus zu wechseln



## 7. Custom functions... (1)

- Beim Absetzen einer Anlage
  - Automatisches Extrahieren der Gemeinde (aus AV-Daten)
  - Ermitteln der nächsten verfügbaren Bewilligungs-Nr.
  - Python Functions definiert im Expression Editor
  - Werden als File im QGIS-User-Verzeichnis  
\.qgis2\python\expressions\\*.py abgelegt



```
1 """
2 Define new functions using @qgsfunction. feature and parent must always be the
3 last args. Use args=-1 to pass a list of values as arguments
4 """
5
6 from qgis.core import *
7 from qgis.gui import *
8
9 import psycopg2
10
11 @qgsfunction(args='auto', group='Custom', usesgeometry=False)
12 - def get_ews_anlage_nr(gemeinde, feature, parent):
13     AnlageLayer = QgsMapLayerRegistry.instance().mapLayersByName('Anlage')[0]
14     aggParams = QgsAggregateCalculator.AggregateParameters()
15     aggParams.filter = 'substr("anlage_nr",1,2) = \''+gemeinde+'\"
16     maxVal = int(AnlageLayer.aggregate(QgsAggregateCalculator.Max,'to_int(substr("anlage_nr",4,4))',aggParams)[0] + 1)
17     return '{:0:04d}'.format(maxVal)
18
19 @qgsfunction(args='auto', group='Custom', usesgeometry=True)
20 - def get_ews_anlage_gemeinde_formatted(feature, parent):
21     conn = psycopg2.connect("service='zg_production' user='ews' password='ews'")
22     cur = conn.cursor()
23     SQL = "SELECT ews.get_ews_anlage_gemeinde_formatted('geometry_wkt')".format(geometry_wkt=feature.geometry().exportToW
24     cur.execute(SQL)
25     gemeinde_formatted = cur.fetchone()[0]
26     conn.close()
27     return gemeinde_formatted
```

## 7. Custom functions... (2)

- Beim Absetzen einer Anlage
  - Automatisches Extrahieren von Grundstücks-Nr. und Adresse (wenn vorhanden) aus den AV-Daten
  - Python Function definiert in den Feldereigenschaften
  - Wird mit dem Projekt gespeichert

Layer Properties - Anlage | Fields

Attribute editor layout: Drag and drop designer

Function name: my\_form\_open

Id	Name	Edit widget	Alias	Type	Type name	Length	Precision	Comment	WMS	WFS
123 6	gesamtfulmenge	Text Edit	Gesamtfüllmenge [Ltr]	int	int4	-1	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1.2 7	max_leistung	Text Edit	Max. Leistung [kW]	double	numeric	7	2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
123 8	verwendungszweck	Value Relation	Verwendungszweck	int	int4	-1	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
abc 9	fabrikat_sachsennummer	Unique Value	Fabrikat_Würkernummer	QString	varchar	100	-1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Python Init function: Provide code in this dialog

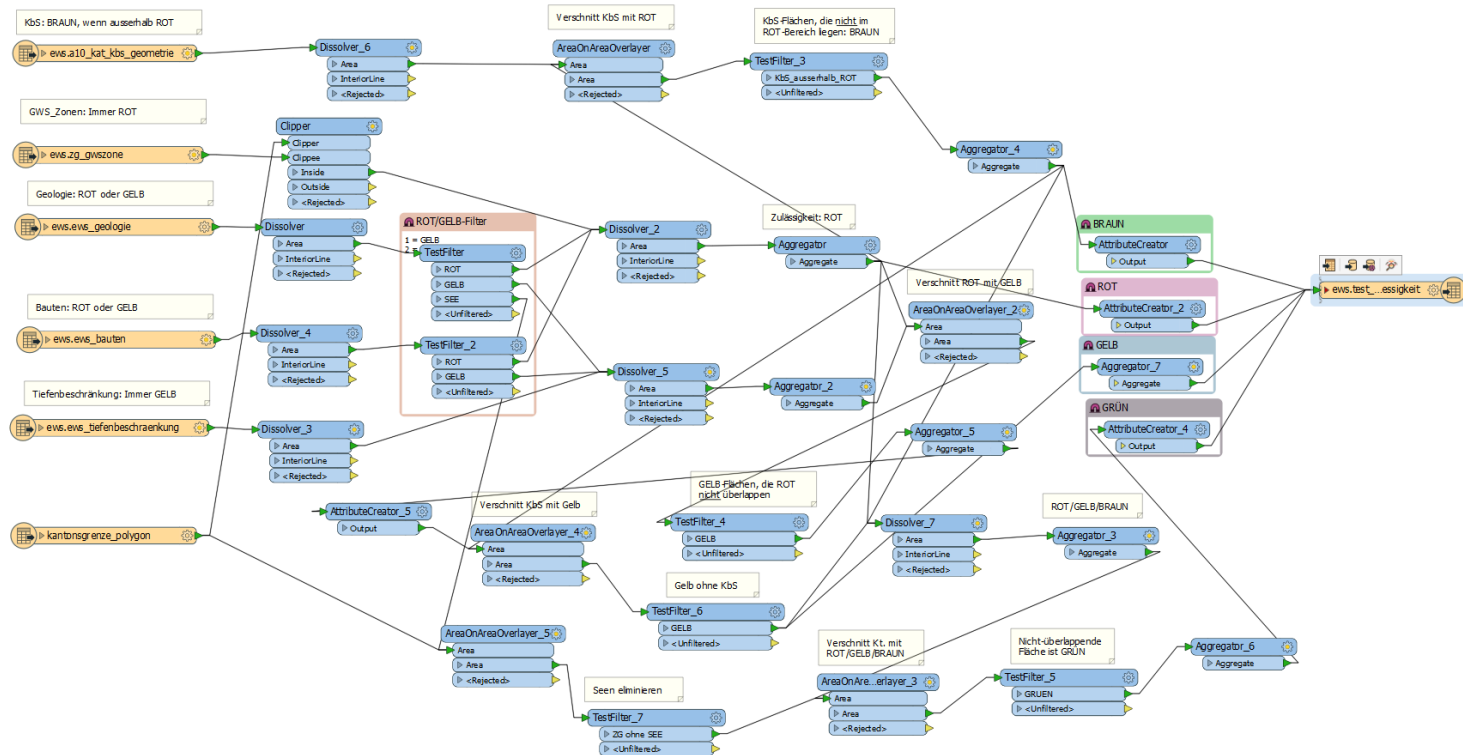
Python init code

```
1 #-*- coding: utf-8 -*-
2 """
3 QGIS-Formulare können eine Python-Funktion haben, die beim Öffnen des Formulars gestartet wird.
4 Hier kann dem Formular Extra-Logik hinzugefügt werden.
5
6
7 Der Name der Funktion wird im Feld "Python-Init-Funktion" angegeben.
8 Ein Beispiel:
9 """
10 from PyQt4.QtGui import *
11 from qgis.utils import *
12 from qgis.gui import *
13 import psychopy
14
15 def my_form_open(dialog, layer, feature):
16     if feature.isValid():
17         #zuerst herausfinden ob bereits Grundstücke verknüpft sind
18         #myForm = QgsForm(dialog, feature, layer)
```



# 7. Custom functions... (3)

- Automatisches Erzeugen der Zulässigkeitsgebiete
  - aus den einschränkenden flächenhaften Datenlayern





## 8. Probleme/Stolpersteine (kleine Auswahl)

- ili2pg
  - Zusammenspiel von ili2pg und QGIS hat noch Verbesserungspotential (z.B. fehlende Fremdschlüssel bei Domains)
  - Die globale DB-Sequenz für die Primärschlüssel (t\_id), die ili2pg anlegt, macht für die Datenmigration mit FME Probleme
- QGIS
  - Handling von n:m-Beziehungen ist noch nicht ideal (Konfiguration der Relationen und der Formulare)
  - QGIS "triggert" die globale DB-Sequenz (von ili2pg) zu oft (grosse Sprünge in den Primärschlüsseln)
  - Durch den Verzicht auf eigene Plugins ist der Handlungsspielraum eingeschränkt;

## 9. To do...

- Zusammenführen der versch. Projektteile
- Erzeugen der Zulässigkeitsgebiete auf Button/Icon legen
- Probleme betreffend Workflow lösen (z.B. Aktualisieren der XML-Kataloge)
- QGIS 3.0 abwarten für einzelne Bugfixes, Verbesserungen
- Suchmöglichkeiten verbessern
- ...

