

12. Deutsches GeoForum 2024, DDGI 8.11.2024

Vorteile von Low Code Geo-Plattformen im Vergleich zu Pro Code

Ralph D. Humberg
Tama Group GmbH
2024.11.08

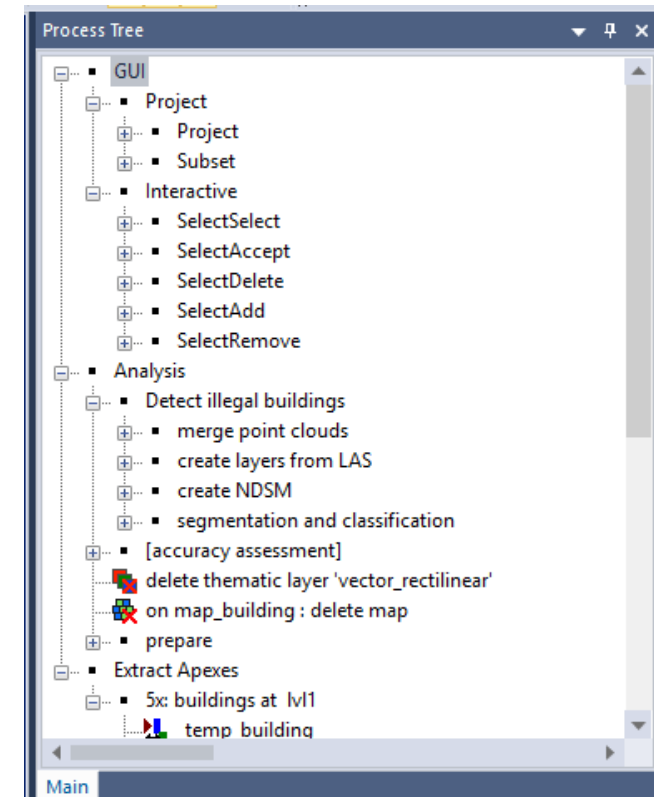


Low Code – was ist das

Low-Code ist eine **Methode der Anwendungsentwicklung**, bei der das Programmieren nicht mehr auf der Basis von Textzeilen, sondern mit visuellen Hilfsmitteln erfolgt. Anstatt in einer technischen Programmierumgebung arbeitet Low-Code mit einer modellgesteuerten **Drag-and-Drop-Oberfläche**.

Pro-code refers to the use of traditional programming languages such as Java, JavaScript, C#, Python, etc. to create applications. Pro-code allows development of complex prototypes and production systems from the ground up.

Google Definition, 4.11.2024



Beispiel von Low Code in eCognition

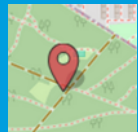
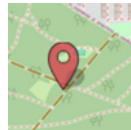
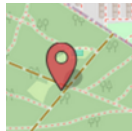
Ob Pro Code oder Low Code hängt von der Sichtweise ab

Aus Sicht der Software Entwickler

- Codieren ist schneller in ascii 10 Finger
- Zugang zu guten Libraries
- Modularisieren großer Projekt / Teamwork

Aus Sicht der Anwender

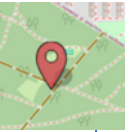
- Versteht der Codierer mein Problem
- Wie schnell bekomme ich Änderungen
- Wie stabil ist die Software



Kernargumente für Low Code

Aus Sicht der Software Manager

- Was kostet die Entwicklung, das Testen, das Debuggen, die Tools, das Refactoring, das alles
- Habe ich ein Team
- Habe ich USPs



Aus Sicht der IT-Administratoren

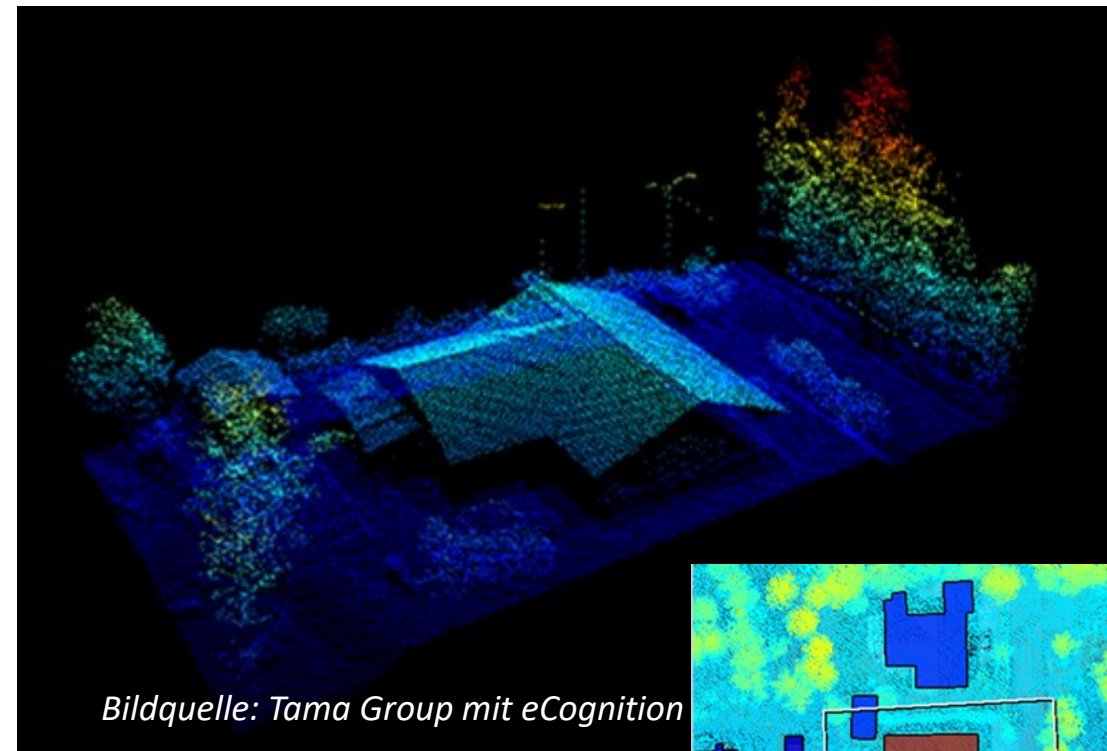
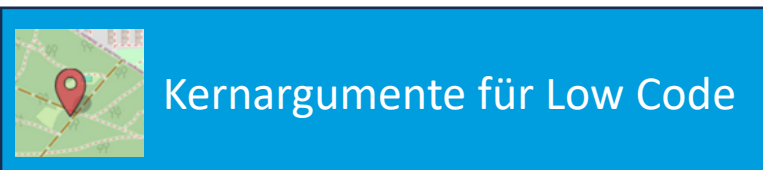
- Virtualisierbarkeit, Sicherheit, Offline-Betrieb, Lizenzierungs-Modelle
- Updates, Patches, APIs

Aus Sicht von ChatGPT et al

- Passt ins Sprachmodell
- Freier Zugang zu code

Versteht der Codierer mein Problem

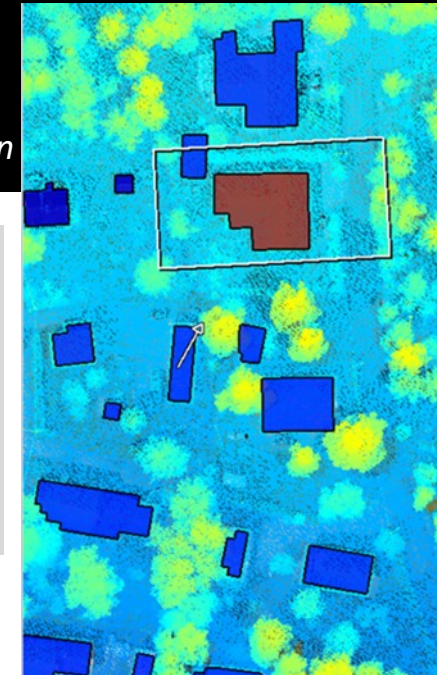
- Schnelle Feedback loops bei der Entwicklung von ‚customized solutions‘
- Anwender und technischer Consultant (=Low Code Programmierer) benötigen keinen Produkt Manager um die Ziel-Lösung zu besprechen
- Oft für Rapid Prototyping verwendet = erste Lösung für ein Problem
- Der technische Consultant kann ein Fachexperte für Bauwesen, PV-Anlagen, Forstwirtschaft, Umweltplanung, oder oder sein.



Bildquelle: Tama Group mit eCognition

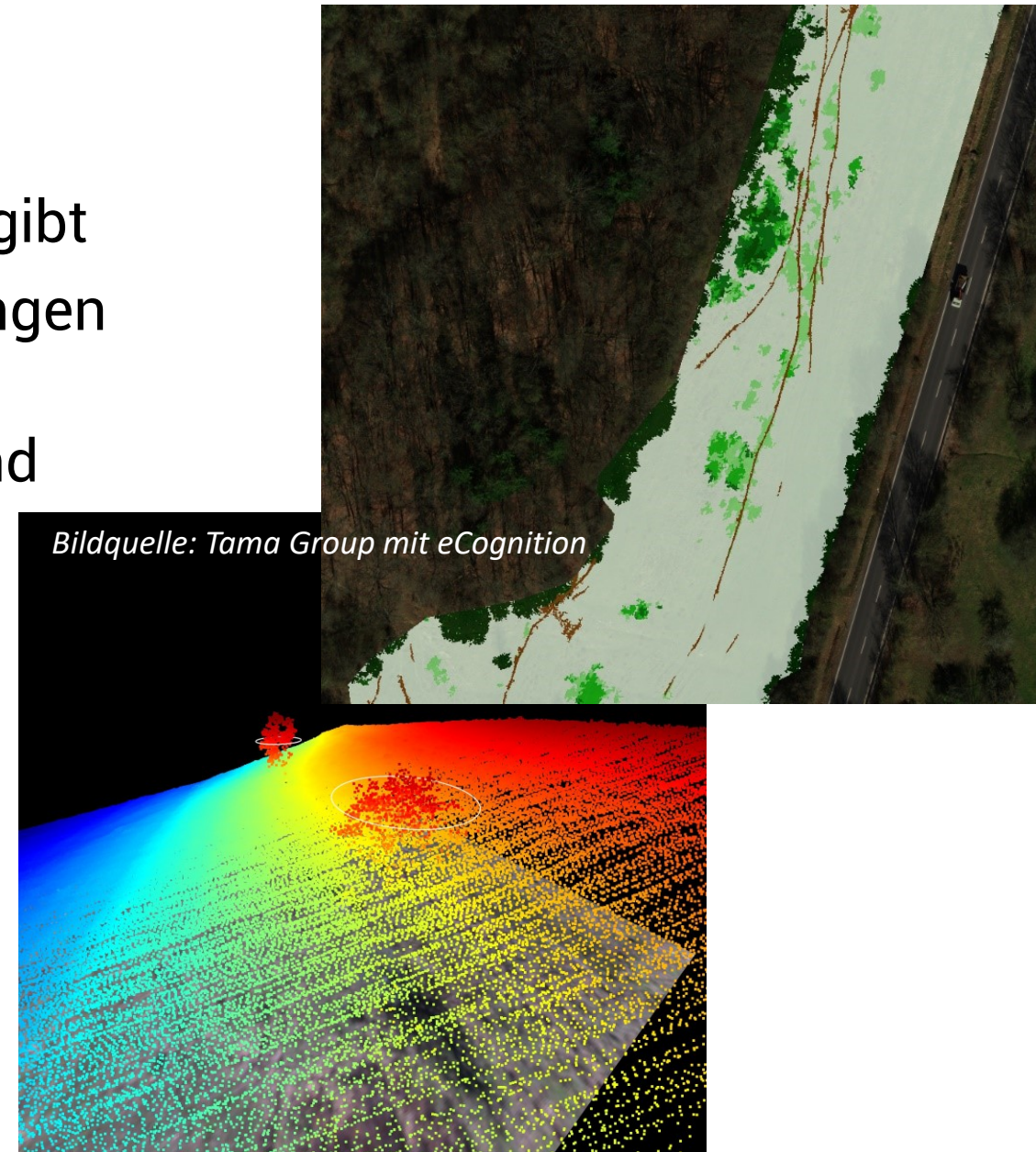
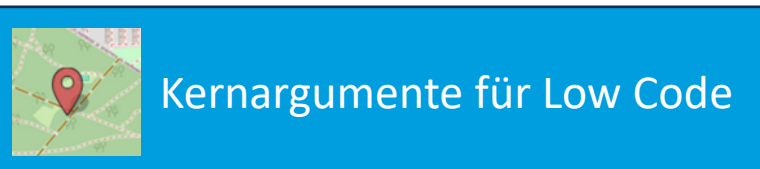
Spezielle Fragen wie

- Anschlüsse für PV Anlage
- Abstand zu 20m hohen Bäumen
- Differenzen zu Eingabeplan
- Firstlänge
- uvm

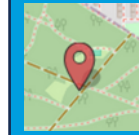


Was kostet das alles zusammen

- Vor allem wenn es nur wenige Anwender gibt
- Wenn es sich um spezifische Fragestellungen handelt
- Viele verschiedene Datenquellen nötig sind um Antworten zu erhalten
→ Datenfusion für Einzelfragen
- Möglichst viel inhouse ablaufen sollte
- Komplexität der Daten wie auch der Fragestellung eine ‚große Lösung‘ teuer und langsam macht

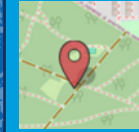


Fazit „Low Code versus Pro Code“



Kernargumente für Low Code

- Beide Methoden haben ihre Vor- und Nachteile
- **LowCode punktet insbesondere bei Rapid Prototyping und kleinen Anwendergruppen, dann auch für große Datenmengen und komplexer Sensorik**
- ProCode ist das Mittel der Wahl bei großen Programmiererteams und standardisierten Fragestellungen
- In jedem Fall sollte die persönliche Präferenz von Entwicklern nicht den alleinigen Ausschlag geben.
- Gesamtkostenrechnungen sind „fies“: Es reicht nicht, nur die Kosten von Software-Lizenzen zu vergleichen weil reichlich indirekte Kosten entstehen (zum Beispiel beim verteilten Testen, im Produkt-Management sowie das Lösen von Anwender-Problemen), die häufig schwer zu erfassen sind. Trotzdem sollten sowohl Anwender wie auch Manager einen Vergleich durchführen.



Kernargumente für Low Code

Vielen Dank für Ihre Zeit



CALL FOR PAPERS, Ralph D. Humberg, Tama Group GmbH, 2024.07

Vorteile von Low Code Geo-Plattformen im Vergleich zu Pro Code

in wissenschaftlichen Bereichen sind Pro Code Entwicklungen, häufig mit Python und JavaScript, eine beliebte Methode um Probleme zu lösen. Die meisten Entwickler nutzen Libraries um Zeit zu sparen, und arbeiten mit Plattformen wie GitHub um Code zu verwalten. Während diese Methoden für viele Themen hervorragend funktionieren, entstehen aus Management-Sicht einige Herausforderungen, die zunächst nicht offensichtlich sind. Bei genauerer Betrachtung entsteht zumeist ein zweischichtiges Entwicklungsmodell: Der Code ist von und für Entwickler gemacht, die Anwendung von Domänen-Experten, wie Hydrologen, Agronomen, Geologen oder Förstern. Bedauerlicherweise findet sich bei dem notwendigen Übersetzungsprozess einiges in 'lost in translation' - die Fachanwender können oft nicht beurteilen ob ein Problem nicht sauber gelöst wurde weil der Entwickler kein entsprechendes Template in der Library gefunden hat oder weil die Datengrundlage mangelhaft ist oder weil die Problemdefinition unzureichend ist oder oder. Im Rahmen einer Geo-Masterarbeit an einer Universität können wir noch erwarten dass die Masteranden sowohl im Code wie auch in der Fachdisziplin tief genug stecken, doch gilt diese Annahme im Berufsleben kaum mehr.

Abhilfe schaffen hier sogenannte Low Code Plattformen wie eCognition, die Geo-Anwendern ermöglichen direkt herauszufinden ob eine Fragestellung besser wissensbasiert, mit maschinellem Lernen, mit neuronalen Netzen, mit vorhandenem Python-Code oder mit einer beliebigen Kombination daraus lösbar ist. Der erfreuliche Nebeneffekt für alle Geo-Manager in Industrie, Behörden und der Wissenschaft, ist der sichtbare Produktivitätsfortschritt im Bereich von 'rapid prototyping' und bei zentralen Produktionsschritten.