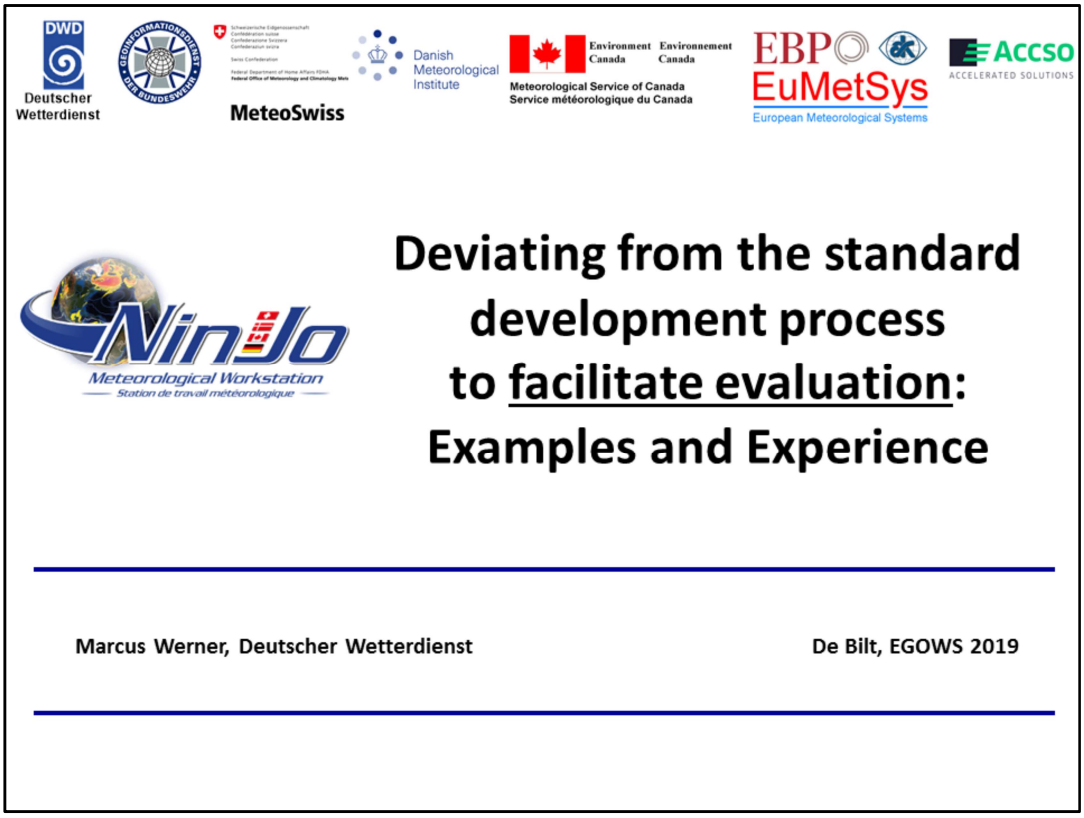

Deutscher Wetterdienst
MeteoSwiss
Danish Meteorological Institute
Environment Canada
Environnement Canada
Meteorological Service of Canada
Service météorologique du Canada
EBP
EuMetSys
European Meteorological Systems
Accso
ACCELERATED SOLUTIONS
NinJo
Meteorological Workstation
Station de travail météorologique
Deviating from the standard development process to facilitate evaluation: Examples and Experience
Marcus Werner, Deutscher Wetterdienst
De Bilt, EGOWS 2019

None

Agenda

- Status & development for operations
- A new stakeholder group at DWD (with new types of requirements)
- Reflecting our current development process
- Case studies: Trials using a new way
- Outside = Inside or "How to avoid duplication of work"
- Conclusions

none

At DWD: NinJo is used as a key working tool which fulfills majority of expectations

- Operationally DWD relies almost **100% on NinJo**
- All **important** functionality and **data** are **supported** in a sufficient way

- For pure operational visualization aspects, we are "mostly" in maintenance mode and **just add "bells and whistles“**

- **Production tools**, i.e. map editors like warning process, etc. **are actively extended** (-> AWEM Europe, Week Weather, AutoWarn-E, WRP, ...)
    - **Workflow** related stuff is a big topic for all consortium partners,
    - We would like to improve **collaboration** support

- We work on technical items (under the user radar, to ensure maintenance and the platform future, e.g. we will not discuss the need for higher test coverage with our users :-))

NinJo works in operations…

User / stakeholder satisfaction regarding operational usage

- stability is high

- performance is OK (complaints are marginal)

- System downtime is almost non-existing (also due to trained staff, # operated sites, good hardware and stable network)

Workload of the team

- Over **years still very high**

- **~4500 OPEN Jira** issues (we discuss things like Kanban, to reduce # active entries per developer)

- We maintain **3 major NinJo versions in parallel**

  *(3.X), 4.X, 5.X and soon start development of NinJo 6.X

- Active maintenance and re-work / **re-design of older components** (e.g. BaseGrid re-design -> Streamline Talk from Anna)

- Special technical work - almost completed **migration to Java 11** (Oracle Java license cost, security requirements, Jigsaw new module system)

Challenges

People approach us with…

- rough / unstable and half thought-out ideas

- in general stakeholders tell us **their solution** rather then their "needs“

- they provide **minimal** (useful) **documentation**

- for complex problems, fixed and dedicated user experts are missing

- Sounds familiar to you ???

Quality checked Software – ready for operations

We use a **formal heavy development workflow** driven by JIRA

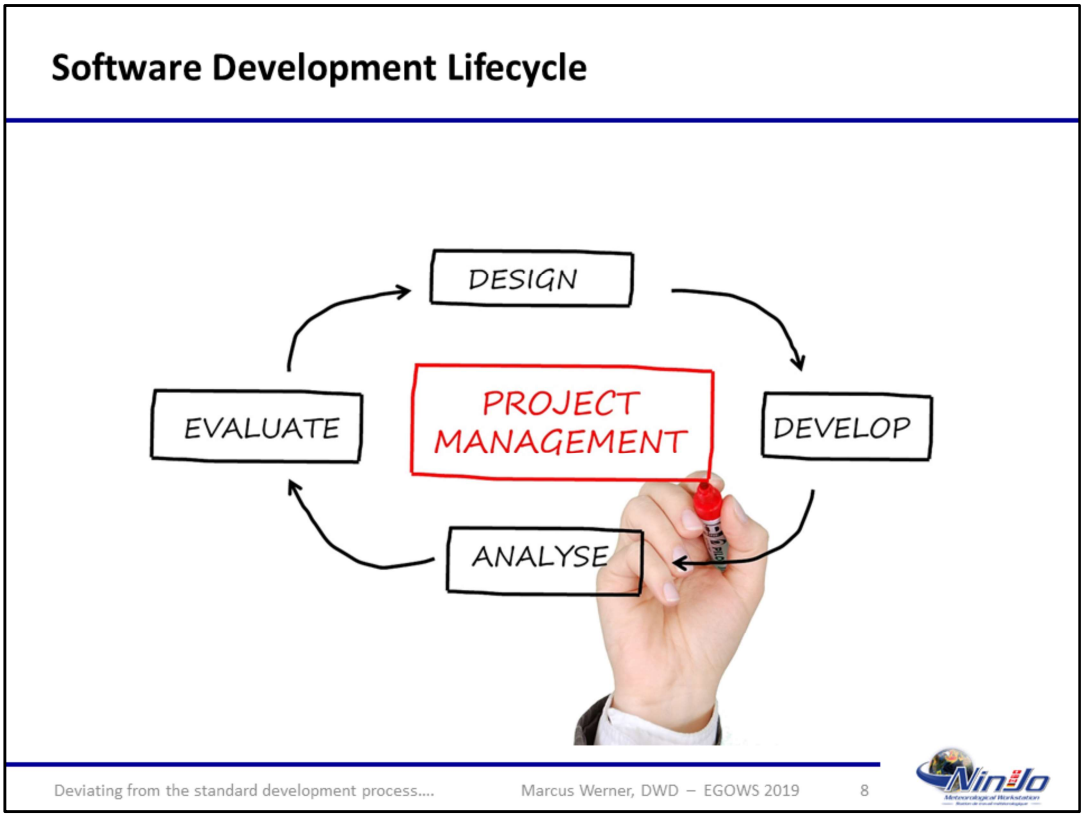
Developers need to go through a lot of steps and deliver many artifacts (incl. agreed design, unit tests, BTS tests, documentation,…, accepted reporter test)

BUT WE STILL BELIEVE it is necessary to keep our "operational" quality high

Pitfall: it lacks flexibility for "evaluation needs“

We also have an alternative way within NinJo (prototype creation)

- We can create Client Plugins
- We can create Server Plugins
- But still requires at least one experienced NinJo Developer
- Overall it could require work from other domain partners (<-> could be challenging within tight timelines)



Our SDLC - System Development Lifecycle

- **Mixture** between **classical approach / Waterfall** (..., Analysis Document, Design Document, Development, Test,...)

- Modern **agile methods** (iterative incremental, communication rich, ...)

- We suffer from **misunderstood & unproductive JIRAs** (mostly a series of New Features & Bugs,...)
    - **JIRAs** might contain discussions with **30-50+ comments**
    - Often **10 or more** of these entries **are linked**
    - These issue are likely to be shifted from release to release
    - Unwrapping this, requires **time-consuming consolidation work**

- Technical collaboration by **Wiki** shared document creation (often NinJo internal)

Our NinJo User (stakeholder) community (DWD view)

- Operational forecaster (*)

- Meteorological specialists (improve data and work methods, workflow, products...)
- Data Producer / Researcher / other Developers
- Training & education staff
- Sales (Data and Products)

- The DWD customers (implicit)

(ranges from "the general public" to specific professional customers (e.g. area aviation))

- Our board of directors (DWD strategy)

Strong influences of our development direction (previous)

Our "previous" main development stakeholders (focus operational needs)

- **Operational Forecaster (main driver)**
- NinJo Evaluation Group (**REG**)

- Experienced **people with a vision**

- External factors (e.g. ICAO changes, data migrations like a new model, new super-computer, ...)

New stakeholders and their needs

Recent influence factors (changed in the **last 2-3 years**)

- **Multiple new research projects** (30 – 40+ new researchers at DWD) **<– attention by board of directors**

- New data types (polygon vs. raster, NowCast Ensembles,…)

- Intermediary results need to be visualized

- DWD NinJo Team (FE21) is now a full part of the DWD research department

How to deal with a way in the fog

Continue …

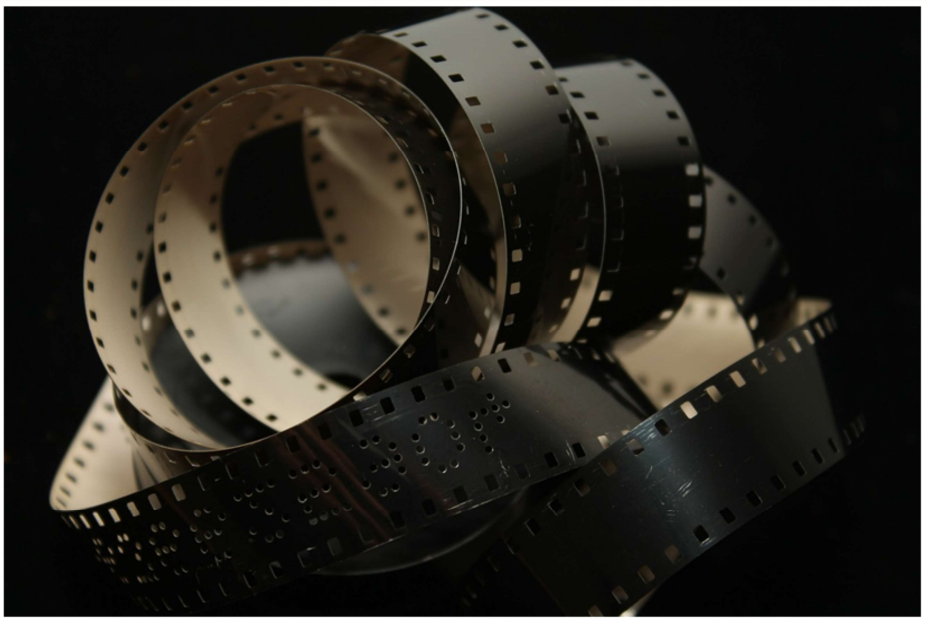
- We now have to deal with “much more uncertainty and a lot of open questions, ...“

- Idea, goals and targets, **get more concrete as we all move forward** …

- Researchers need early feedback from our experienced forecasters and **our forecasters want to see data in NinJo**

- Most people want to use layers to combine new data **with existing meteorological data** (-> advantage of NinJo vs. other tools used at DWD)

New formats & content



- Polygon data, classified or feature data are preferred in comparison to raster data

- Dimension reduction & simplification

- We discuss and work with a wide range of file / content formats
    - netcdf (with CF 1.6 convention), hdf5 ??
    - variety of XML-, GeoJSON -> tags and content (not very specific)
    - CAP 1.2

    - OGC and WMO standards do not really help us in this case

Wishes and complaints

Stakeholders:

- We need a workable solution **ASAP (fast like yesterday)**, the **NinJo team** is likely to **deliver too slow**

- They mix up **features integrated in an "official NinJo release"** (requires full heavy NinJo SDLC) **versus a slim approach** for "evaluation needs"

Use case #1: Request to support EVAL for a new radar process

Konrad3D

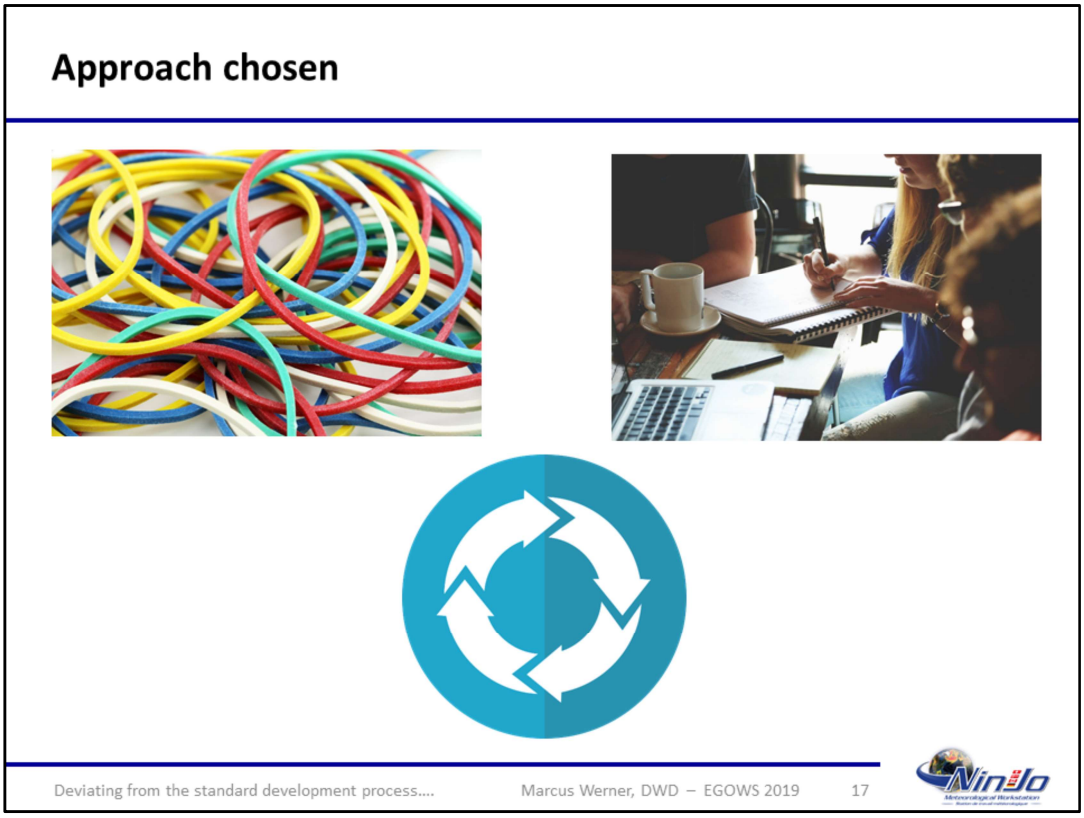
- Cell detection and classification based on hi-res radar sweeps (3D input data)
- 2D output (met-object data)
- Motion tracking (past)
- Movement and strength prediction

Konrad3D – and you can imagine…

A very **tight schedule** by customers and decision makers

Our input data for the **Konrad3D** process (format & content) was in the middle of development, a step away from being finalized – ~**"construction set"**

Old "Konrad" was supposed to be retired ASAP



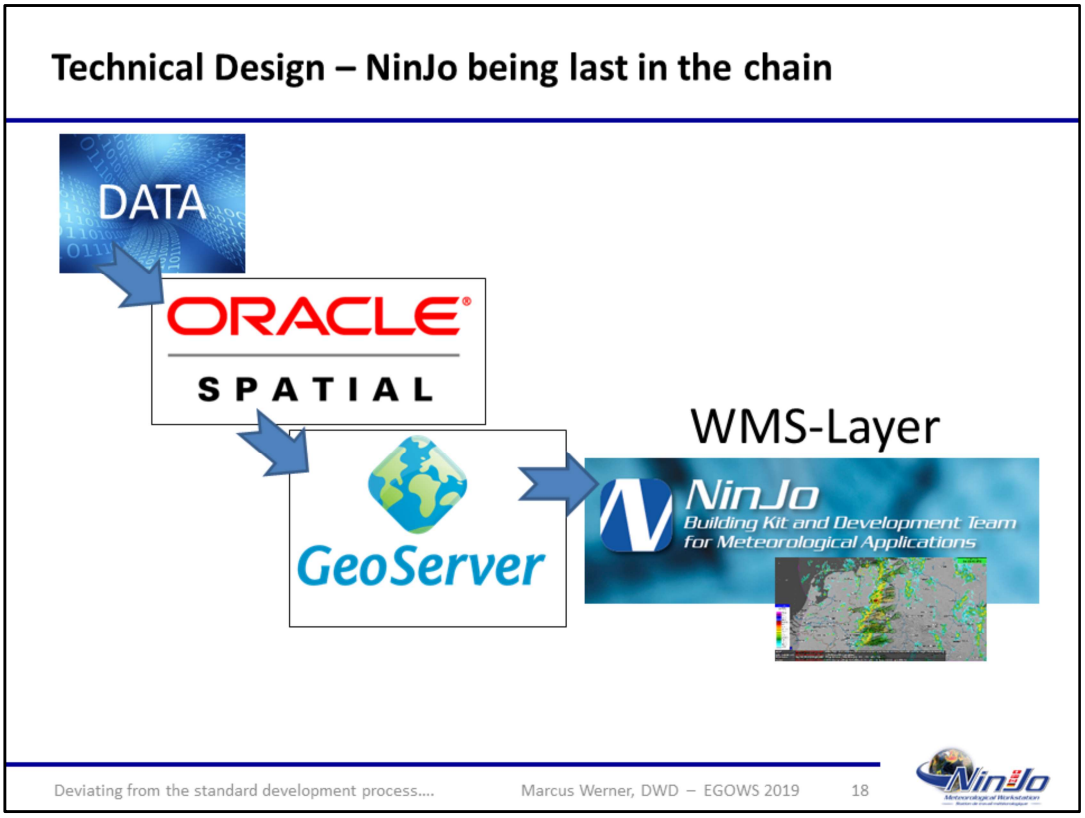
So we looked for….

- **Flexible way** according to **data format & content** and visualization results

- **Small fixed team**, establish a **close communication** between Radar Developers, NinJo Developers and Evaluators/Users

- Can be implemented / supported outside NinJo's Core Team (e.g. minimal work by NinJo framework or key developers)

- Able to work within in **quick iteration cycles** (feedback, change, deliver, test)



SQL import for Oracle

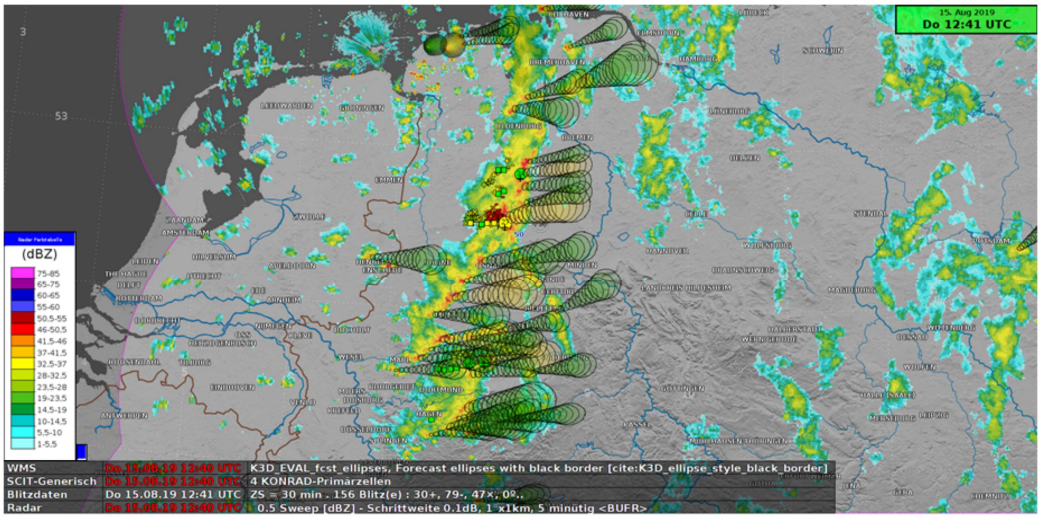
GeoServer layer definitions by SLD-files

NinJo WMS-Layer

NinJo running on a remote xrpd-Server (central maintenance possible)

Evaluation user works with Windows 7 / 10 Remote Desktop

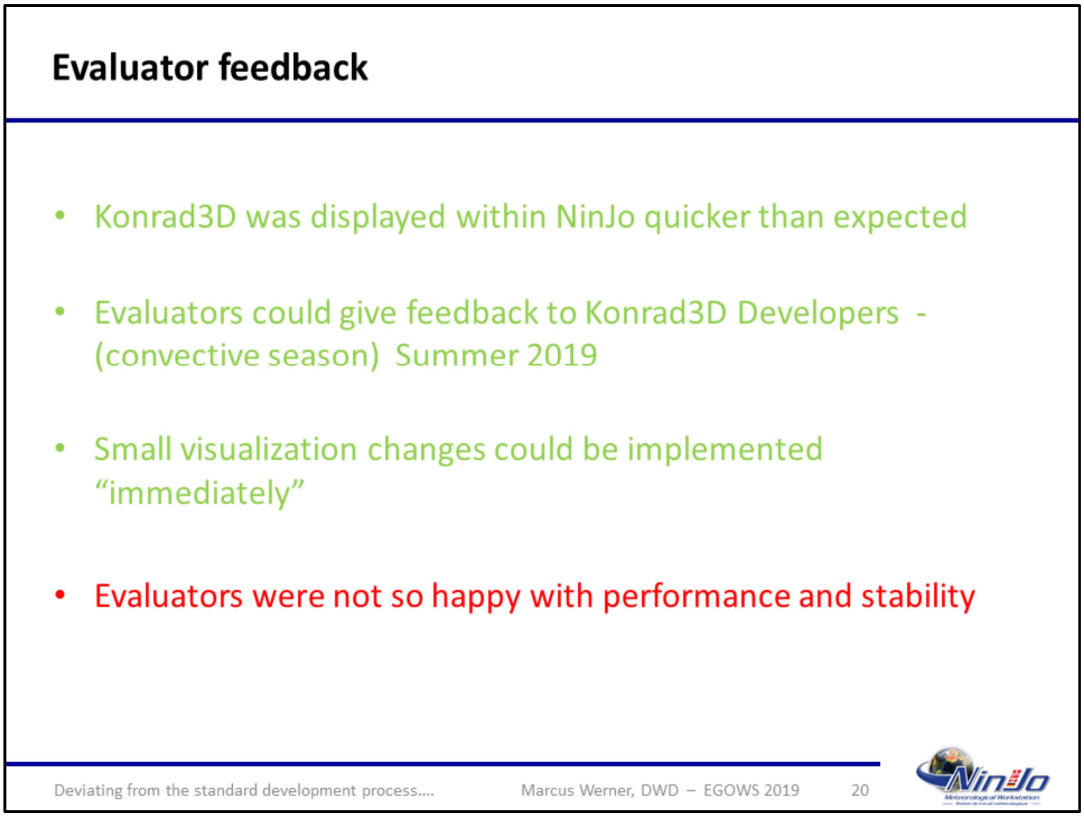
WMS-Layer with Konrad3D and underlying other data



Konrad3D cell forecast ellipses shown in NinJo

Underlying data

- Lightning data
- Radar reflection data
- Old “Konrad” data as reference



The **approach** was completely **new** for our team

There was an Oracle V12.X to **Oracle V18.Y migration** during the evaluation period

Test GeoServer was hosted and **installed by us** and not by the DWD GeoServer SME team

Usage of **NinJo Remote** does not improve reactiveness and hence user experience (compared to the NinJo FAT Client)

There was **no focus on performance tuning** or any other optimization

There are a lot of untested tuning mechanisms possible (SQL side, GeoServer and WMS-Layer)

Use case 2: Evaluating aviation data with external customers

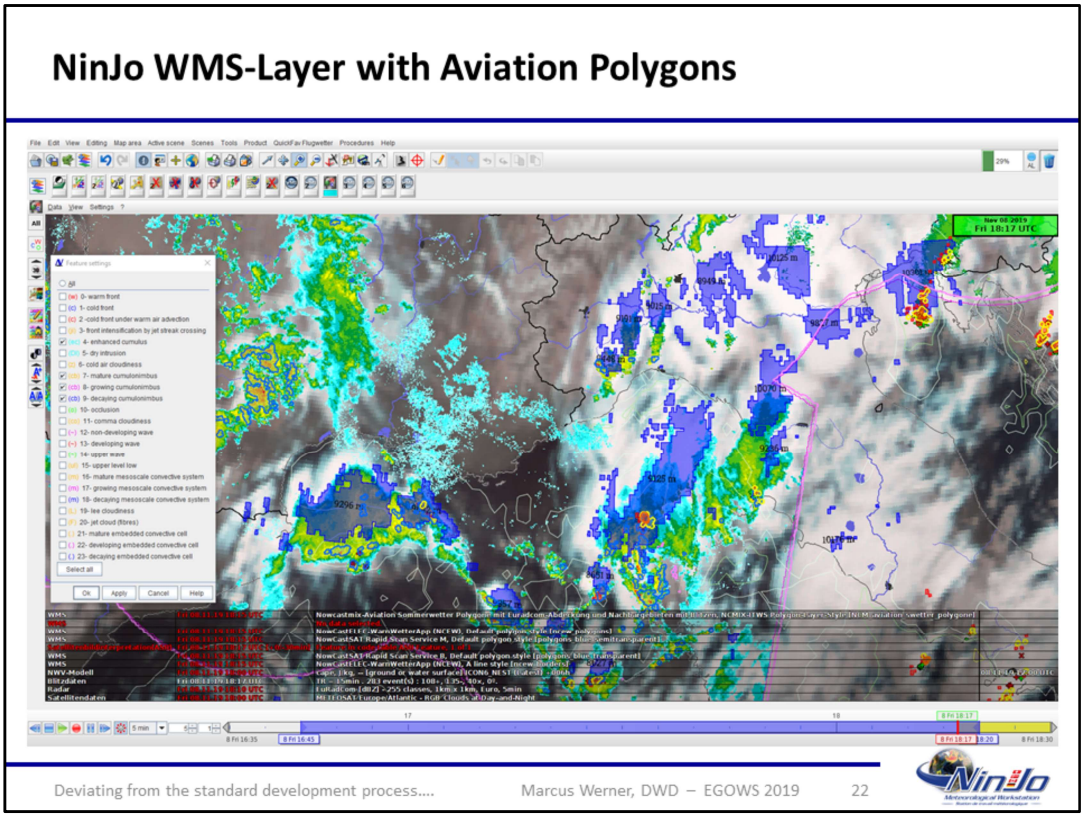
External data and internal usage

- Our aviation department provided special data for aviation customers by GeoServer WMS / WFS

- Data could be displayed by DWD special Web Pages (Flugwetter, Heliportal, ...)

- Forecasters / advisers need to explain the data to customers (want to see "exactly" the same on their NinJo screen)

- Duplicated the data on an internal GeoServer ("load protection and speed optimization")

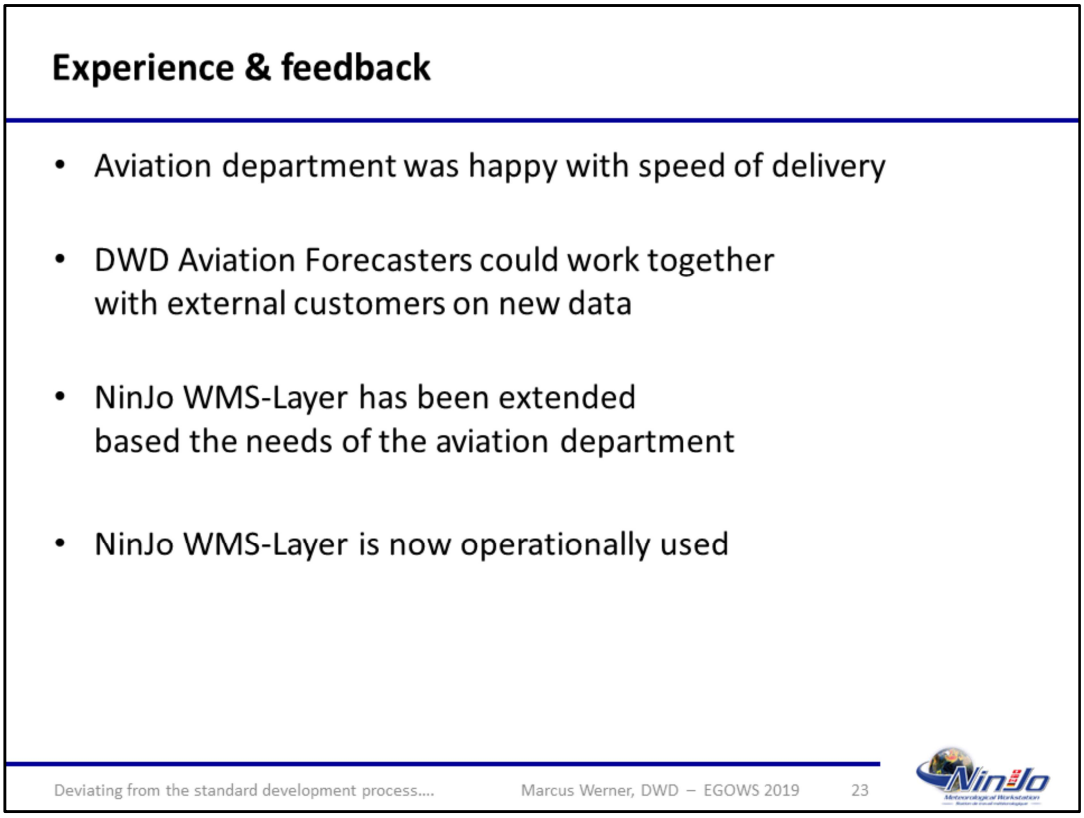


NinJo WMS Layer was extended with **Auto-Update** functionality and general better time behavior support

Be able to **access password-protected data** on GeoServer

Show speaking / **self explaining Server name** (compared to just showing the technical URL)

Show **feature info** display (Text & http)

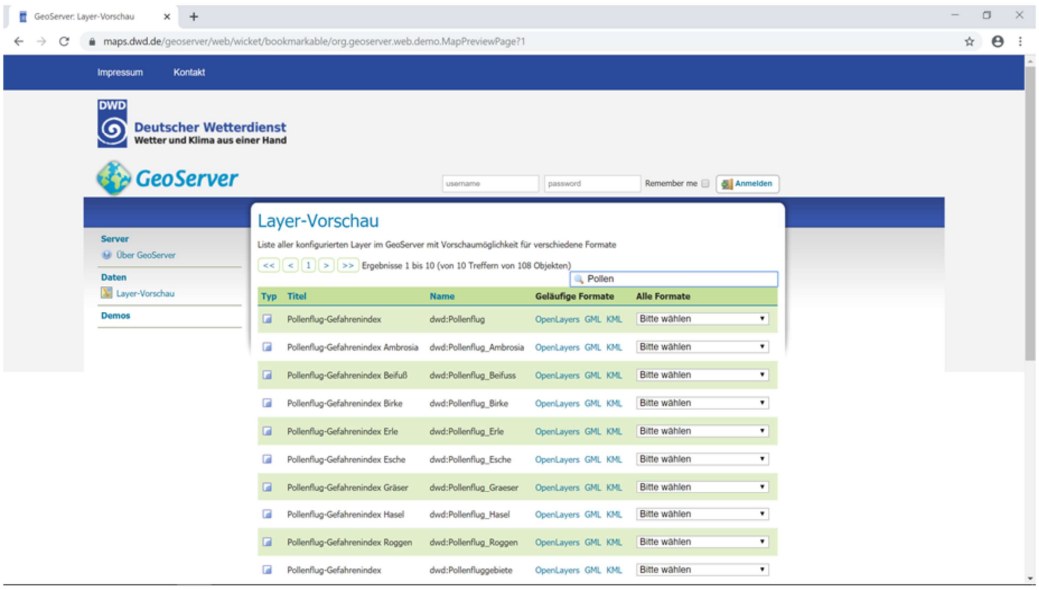


The NinJo WMS-Layer has now started to evolve from a prototype component to a full-grown NinJo-Layer

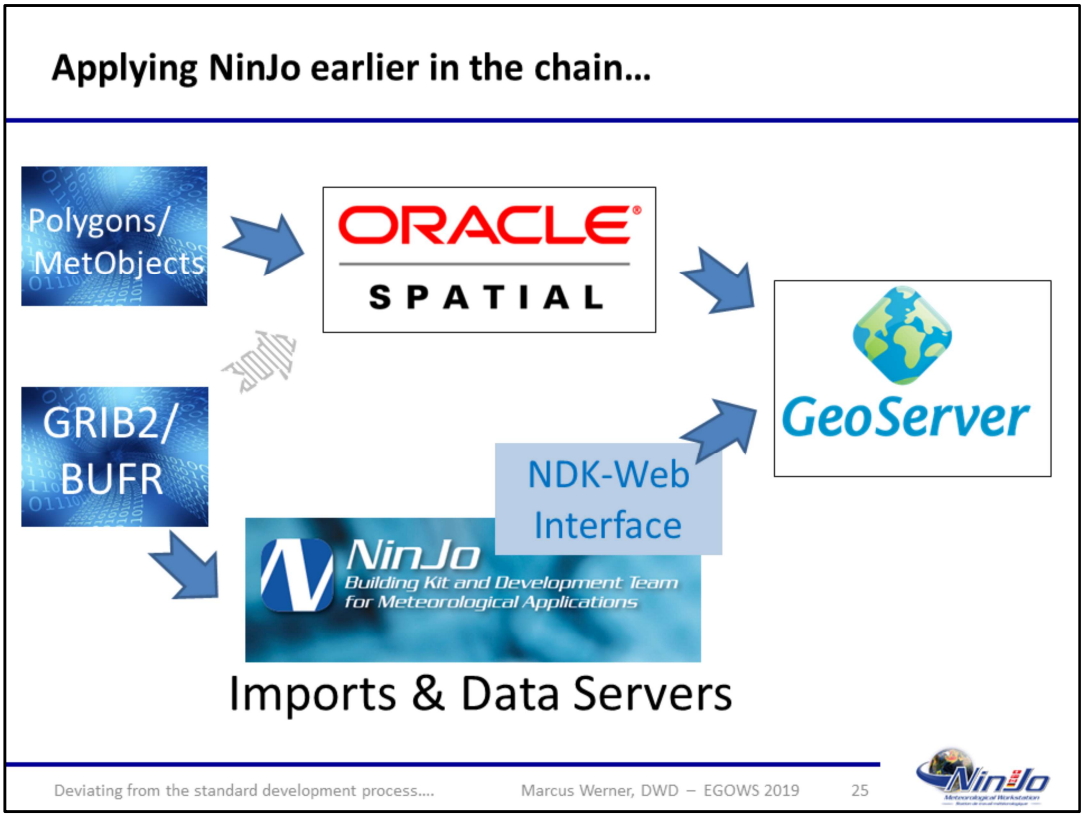
The re-work has started

We have now a dedicated developer as component owner

We try to avoid duplicate work



- DWD fulfilling INSPIRE is slowly progressing
- We hope to speed it up, with the help of NinJo



All important data for DWD exist within NinJo (esp. Server part)

GRIB1 / 2, BUFRs and other standard meteorological data types can go via NinJo into the GeoServer

Spatial referenced polygon data can go via Oracle 18 ff

There is no need to bring data already loaded into NinJo into Oracle

NinJo can help with the NDK-Web Inferface (DMI already uses this interface to feed their Web page dmi.dk)

NinJo WMS-Layer might be again last in the chain

Conclusion

- DWD will increase the usage of WMS & WFS
- GeoWeb Services within NinJo are likely to extended (-> WFS)
- DWD will apply NDK-Web for GeoServer (like DMI for dmi.dk)
- NinJo Team (FE21) and DWD GeoServer Team (TI16) will continue to link both processing streams
- NinJo will stay for the majority of use cases with the traditional 2-tier approach, using a “fat Java client”

none

Special thanks to my DWD colleagues ….

Having contributed to the exercise „Konrad3D Eval. 2019“

- Kathrin Feige (now WV14) & Markus Bork (FE21)
- Manuel Werner (FE23) & Robert Feger (FE23)
- Dirk Fuchs (TI12) & Rene Poch (TI12)
- Thorsten Schulz (TI16) & Rudolf Mohr (TI16)

none

References

- Images from pixabay.com
  - What is allowed?
  - ✓ All content on Pixabay can be used for free for commercial and noncommercial use across print and digital, except in the cases mentioned in "What is not allowed".
  - ✓ Attribution is not required. Giving credit to the contributor or Pixabay is not necessary but is always appreciated by our community.
  - ✓ You can make modifications to content from Pixabay.

Screenshots from DWD owned applications & data