

# Projekt 1: TextureSync

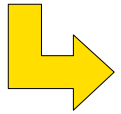
Texturen-Sammlungen einfach und sicher verwalten



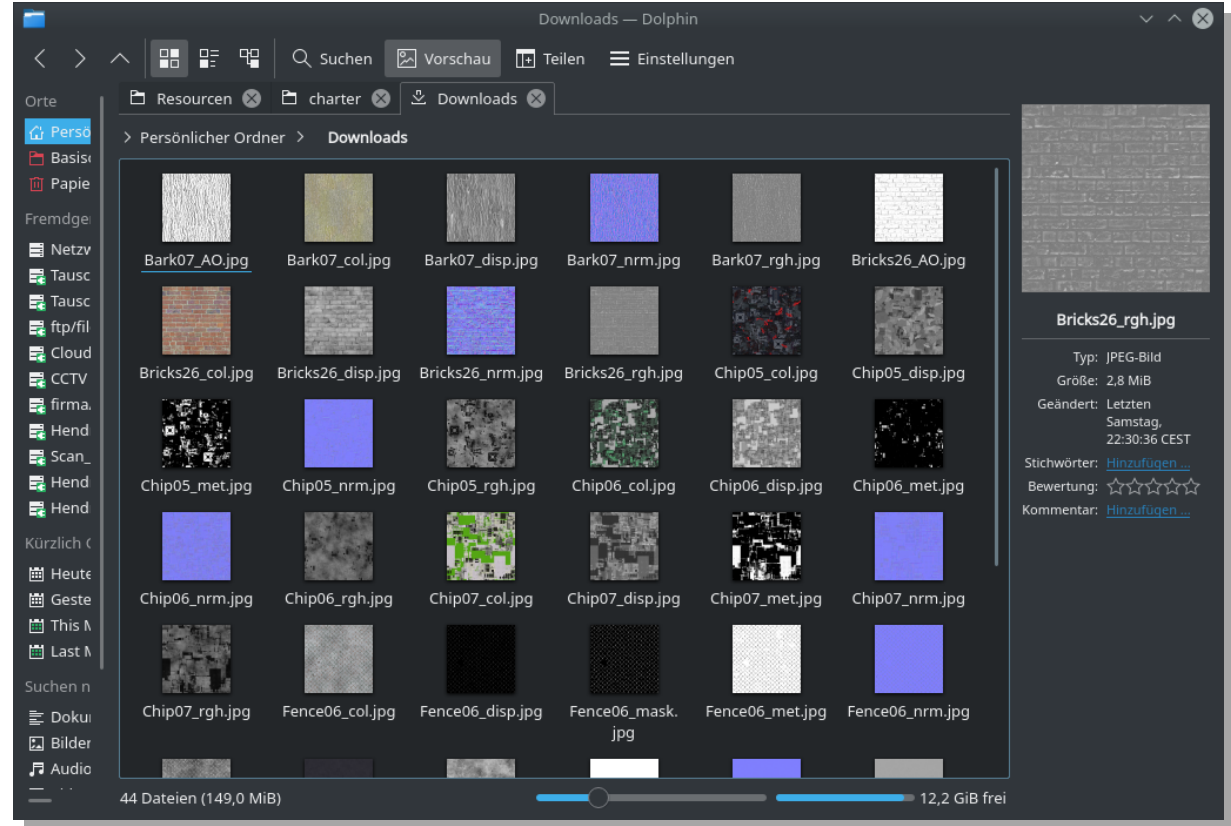
- Problem
- Lösung
- Artefakte
  - Lastenheft
  - Pflichtenheft
  - Grobdesign
  - Tests
- Technologien
- Fazit
- Demo

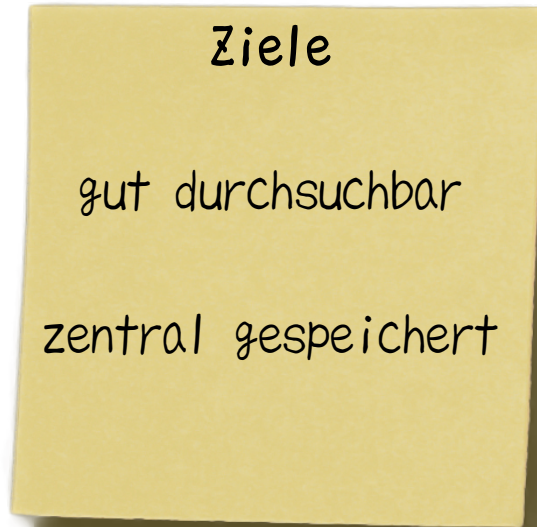


- nicht durchsuchbar
- keine Sortierung
- dezentral gespeichert
- keine Sicherung

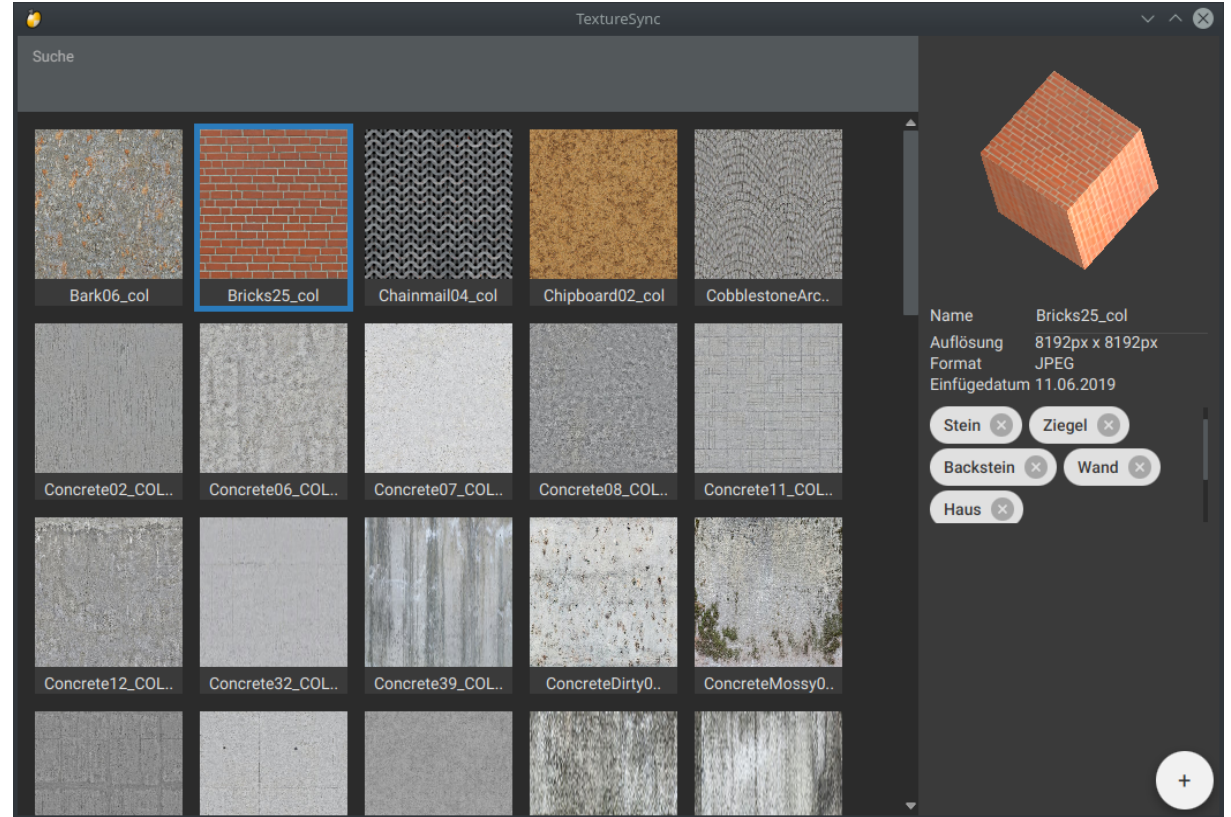


**Verwaltungsaufwand**



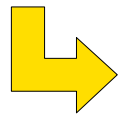


<https://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png>

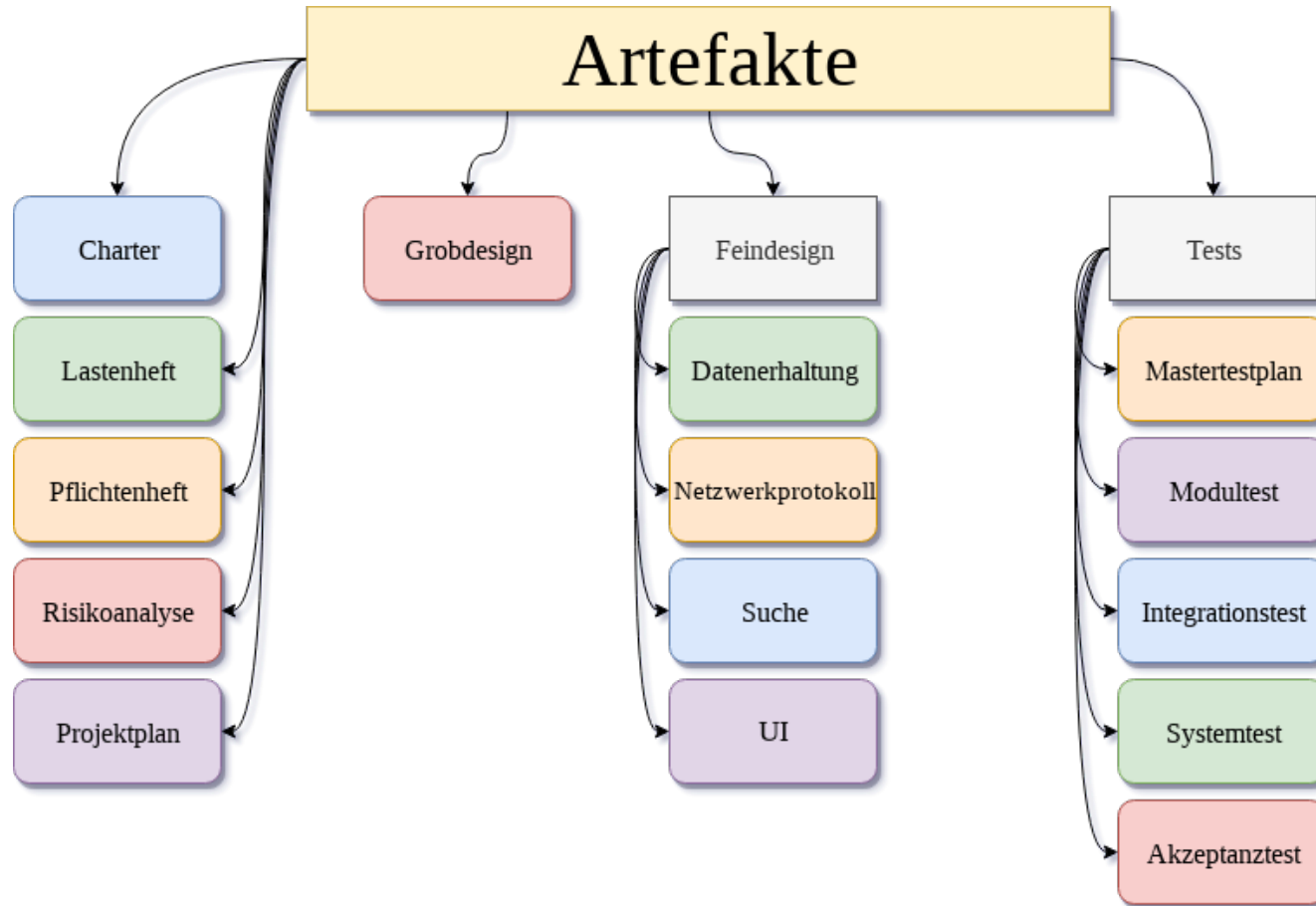


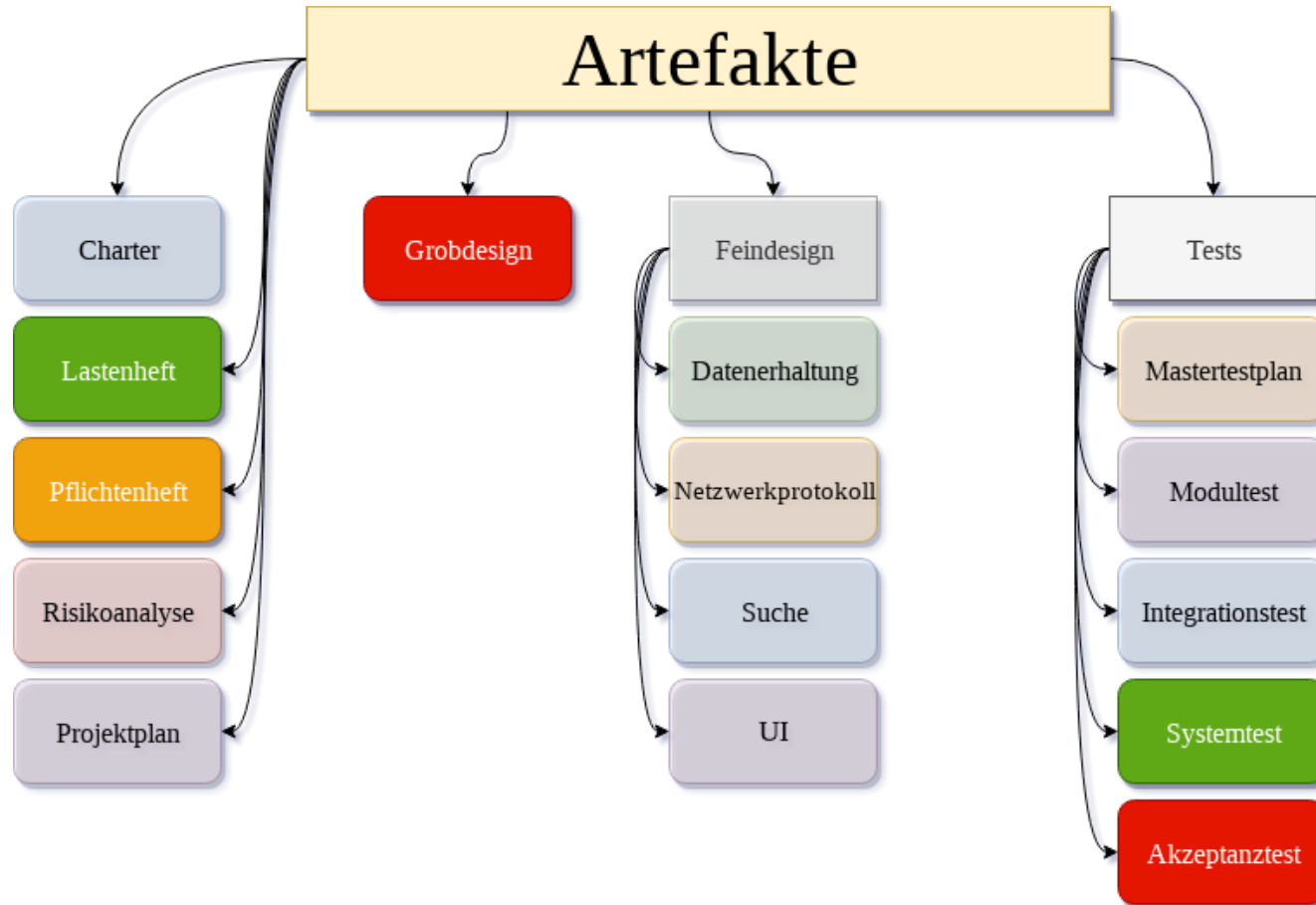
## Auszug aus Projekt-Charta:

Kriterien	
Wichtigste Funktionen	<ul style="list-style-type: none"><li>• Texturen anzeigen und verwalten</li><li>• eine Preview beim Durchsuchen der Texturen</li><li>• ein Tag-System, um Texturen zuzuordnen</li><li>• Filter für Metadaten und Tags</li><li>• Synchronisation mit zentralem Server</li></ul>
Akzeptanzkriterien	<ul style="list-style-type: none"><li>• Das Durchsuchen darf nicht länger als 1 Sekunde bei 1000 Texturen dauern.</li><li>• mindestens 10 Clients gleichzeitig aktiv</li></ul>



Kunden: Animationsstudios, 3D-Designer, Grafikagenturen



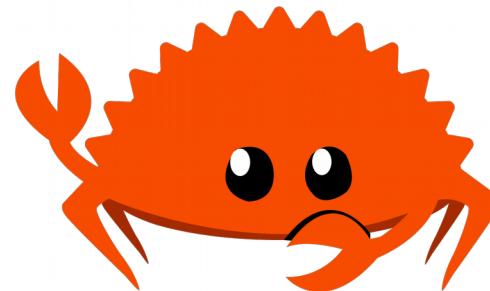


## Server: Rust

- Compilersprache mit ähnlicher Performance wie C++
- Compiler verhindert gängige Fehler in Programmen  
z.B. Seiteneffekte, use after free, race conditions, Pufferüberlauf
- cargo als Toolchain (ist Teil von Rust)  
übernimmt Kompilierung, Testausführung, Abhängigkeitsverwaltung

## Bibliotheken:

- image: Generierung von Vorschaubildern
- serde: Speicherung der Texturdaten in Json-Datei



<https://rustacean.net/assets/rustacean-flat-gesture.png>



## Client: Kotlin

- Syntax ähnlich wie Java
- Sprachfeatures die Java nicht bietet (when mit expressions)
- übersetzt in JVM Bytecode → läuft überall
- Native code (LLVM)
- IntelliJ IDEA (IDE), gradle (build tool)



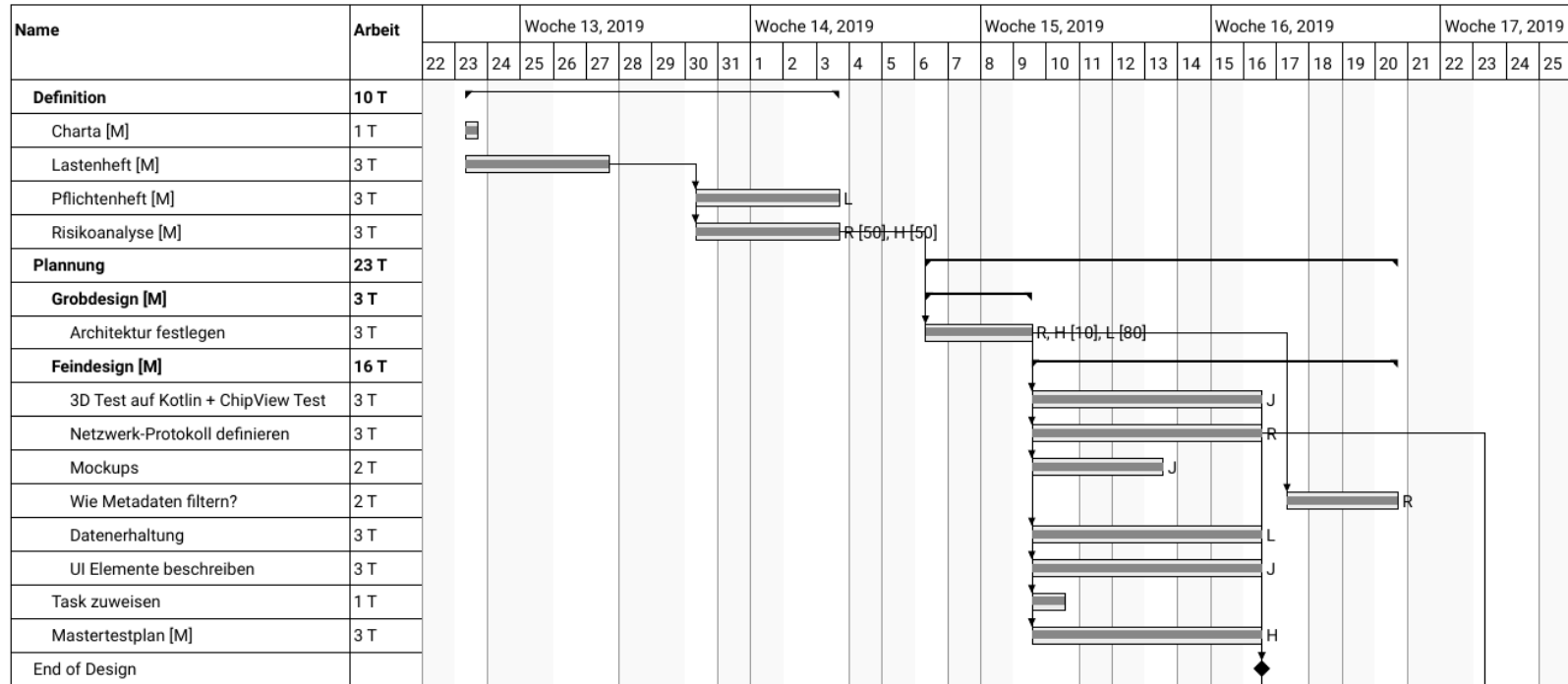
[https://logos-download.com/wp-content/uploads/2016/10/Kotlin\\_logo\\_wordmark.png](https://logos-download.com/wp-content/uploads/2016/10/Kotlin_logo_wordmark.png)

## Bibliotheken:

- GUI Framework: TornadoFX (JavaFX für Kotlin)
- Design lässt sich durch css ändern
- Jfoenix: Material Design GUI Elemente (ChipView, etc.)

# Pläne sind nichts, Planung ist alles.

(Dwight D. Eisenhower, ehemaliger US-Präsident)



## Was hat uns geholfen:

- vorhandenes Wissen
- Issue-Tracker (59 Issues)
- Versionsverwaltung (360 commits)
- Test

### #28 show texture image on importview

**Closed** opened 1 week ago by [localhorst](#) · 1 comments



localhorst commented 1 week ago

you should see the image while brainstorming for tags

- localhorst added the **enhancement** label 1 week ago
- CodeSteak was assigned by localhorst 1 week ago
- localhorst added this to the **client 1.0** milestone 1 week ago



CodeSteak commented 1 week ago

Collaborator

3D-Preview if possible

The screenshot shows the GitHub repository interface for 'TextureSync'. At the top, there are navigation tabs for 'Code', 'Issues 4', 'Pull-Requests 0', 'Releases 2', 'Wiki', and 'Aktivität'. Below the tabs is a search bar with 'Suche...' and a 'Suche' button, and a 'Neues Issue' button. The main content area shows a list of issues. Issue #28 is highlighted, showing it is 'Closed' and was opened 1 week ago by 'localhorst'. Issue #43 is also visible, titled 'UI Glitch bei resize', with labels 'bug', 'client', and 'low priority'. The interface includes filters for 'Label', 'Meilenstein', 'Zuständig', 'Typ', and 'Sortieren'.

**DEMO**