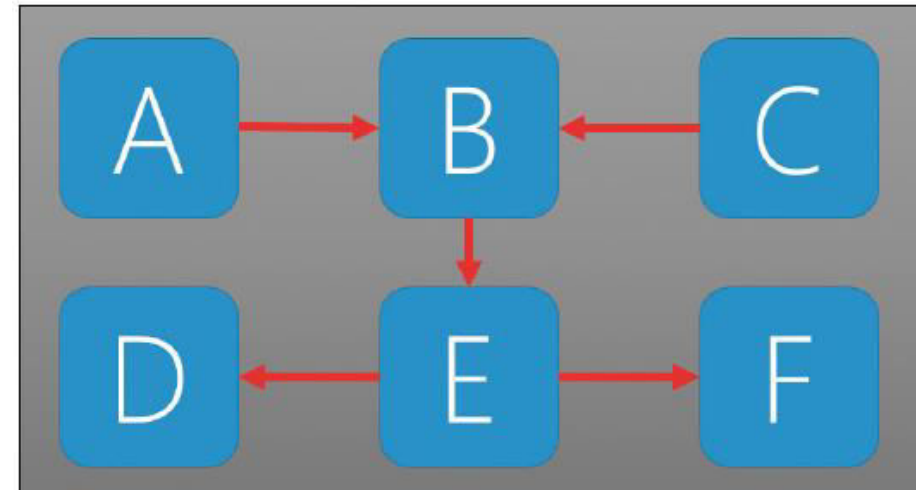
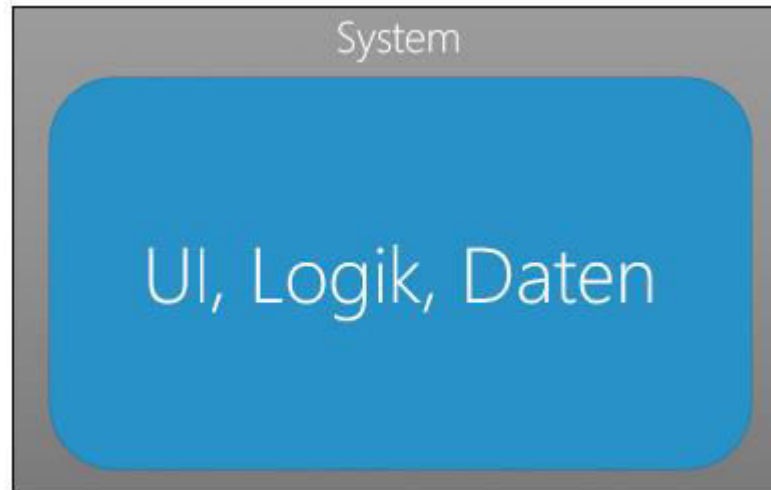


Applikationsentwicklung

Eine Standard-Architektur für die Praxis

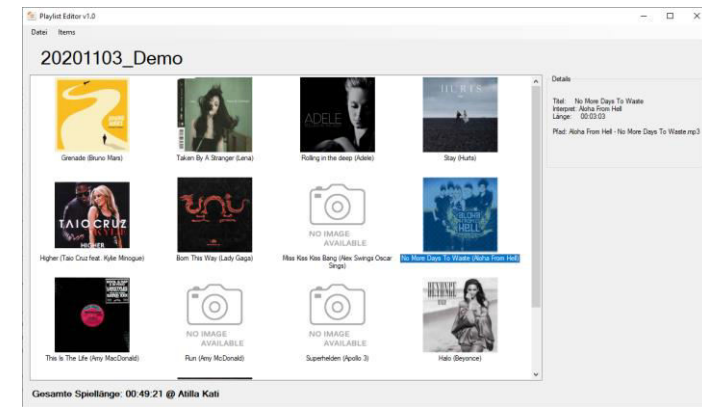
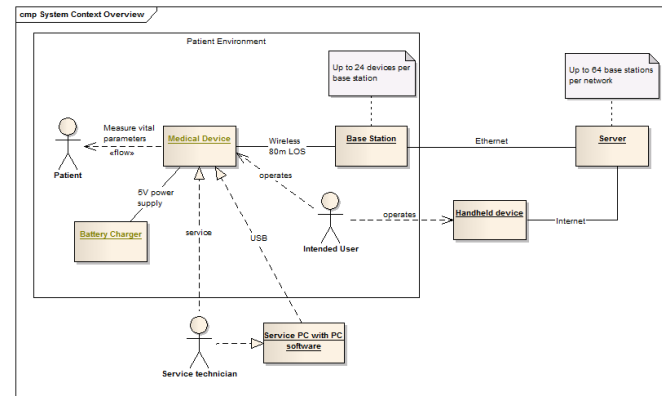
Problemstellung & Ziele

- Was ist falsch mit dem funktionalen Fokus?



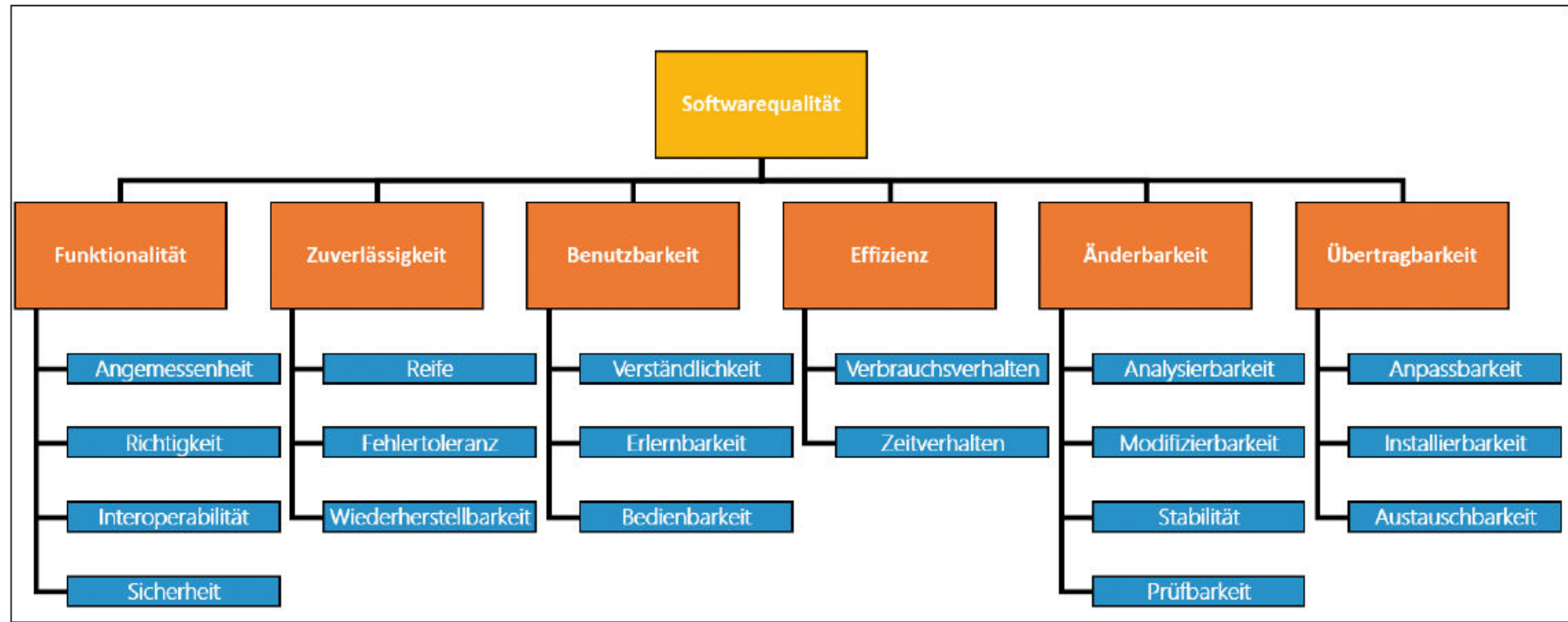
Problemstellung & Ziele

- Was ist falsch mit dem funktionalen Fokus?
- Grundlagen zu Software Qualität und Architektur Patterns
- Umsetzung an einem praktischen Beispiel



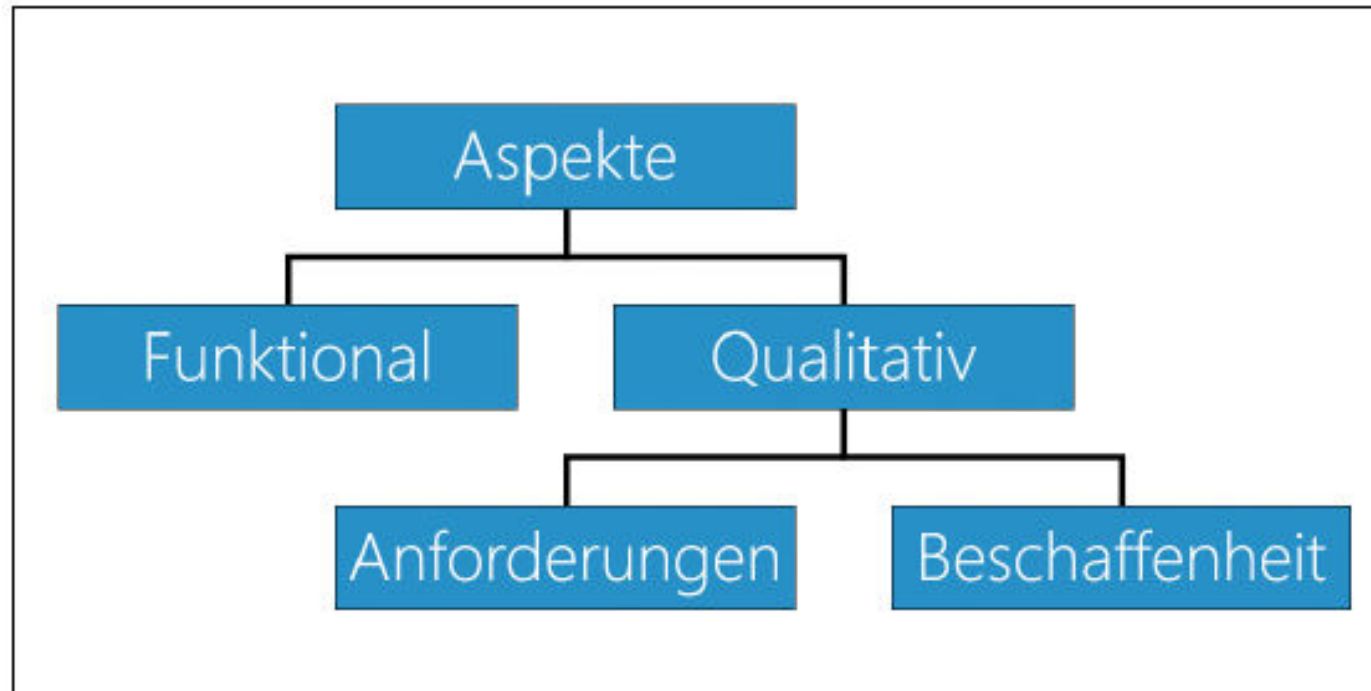
Das Software Qualitäts-Modell

- ISO 25010/2011 (ISO/IEC 9126)



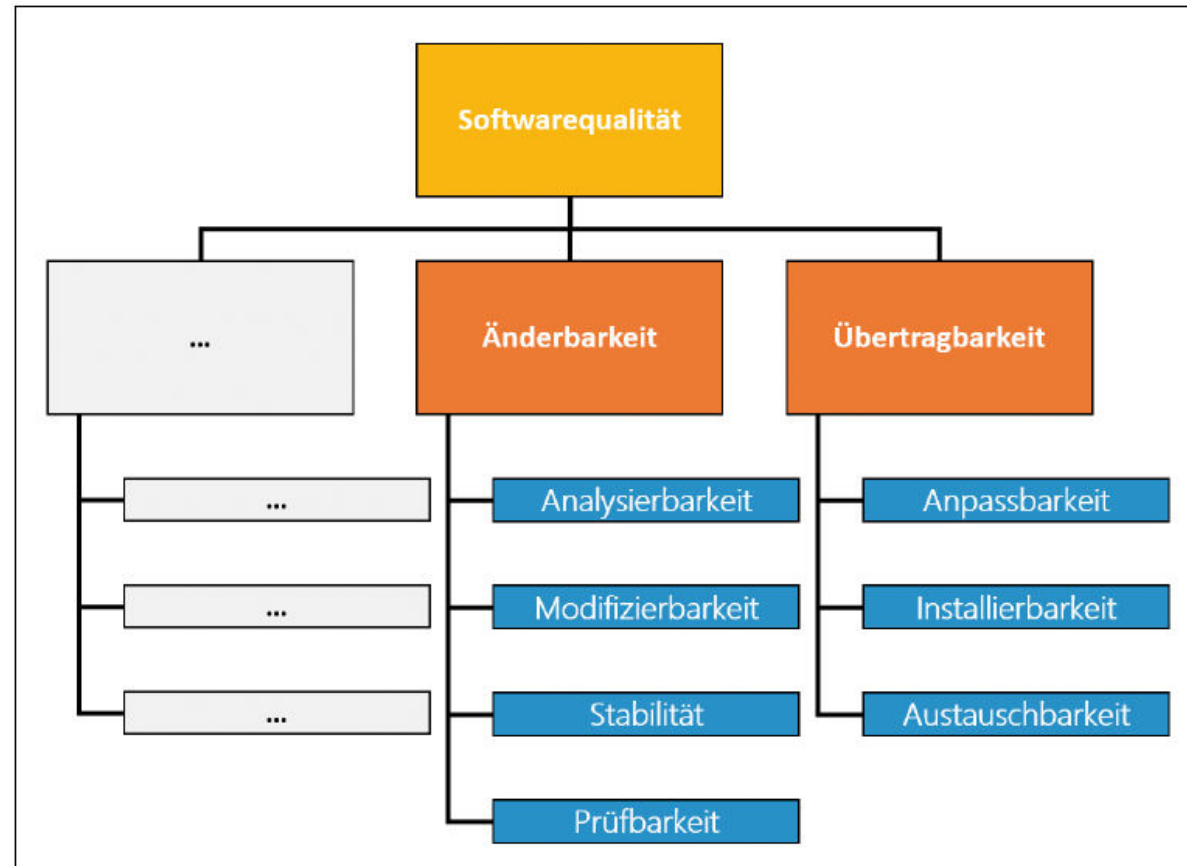
Das Software Qualitäts-Modell

- ISO 25010/2011 (ISO/IEC 9126)



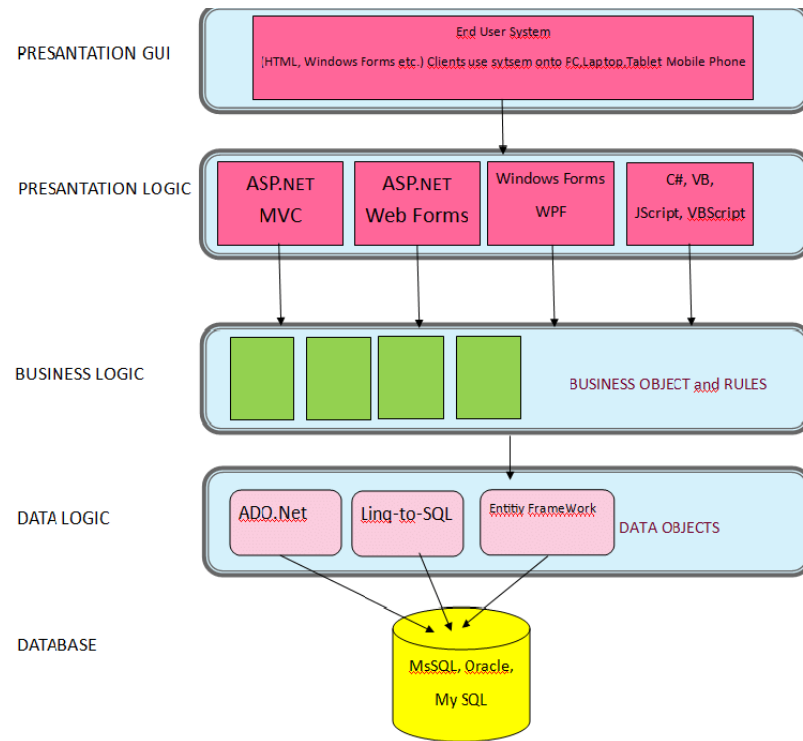
Das Software Qualitäts-Modell

- Konstante Qualitätsmerkmale



Architektur – Design Patterns

- Warum Architektur?

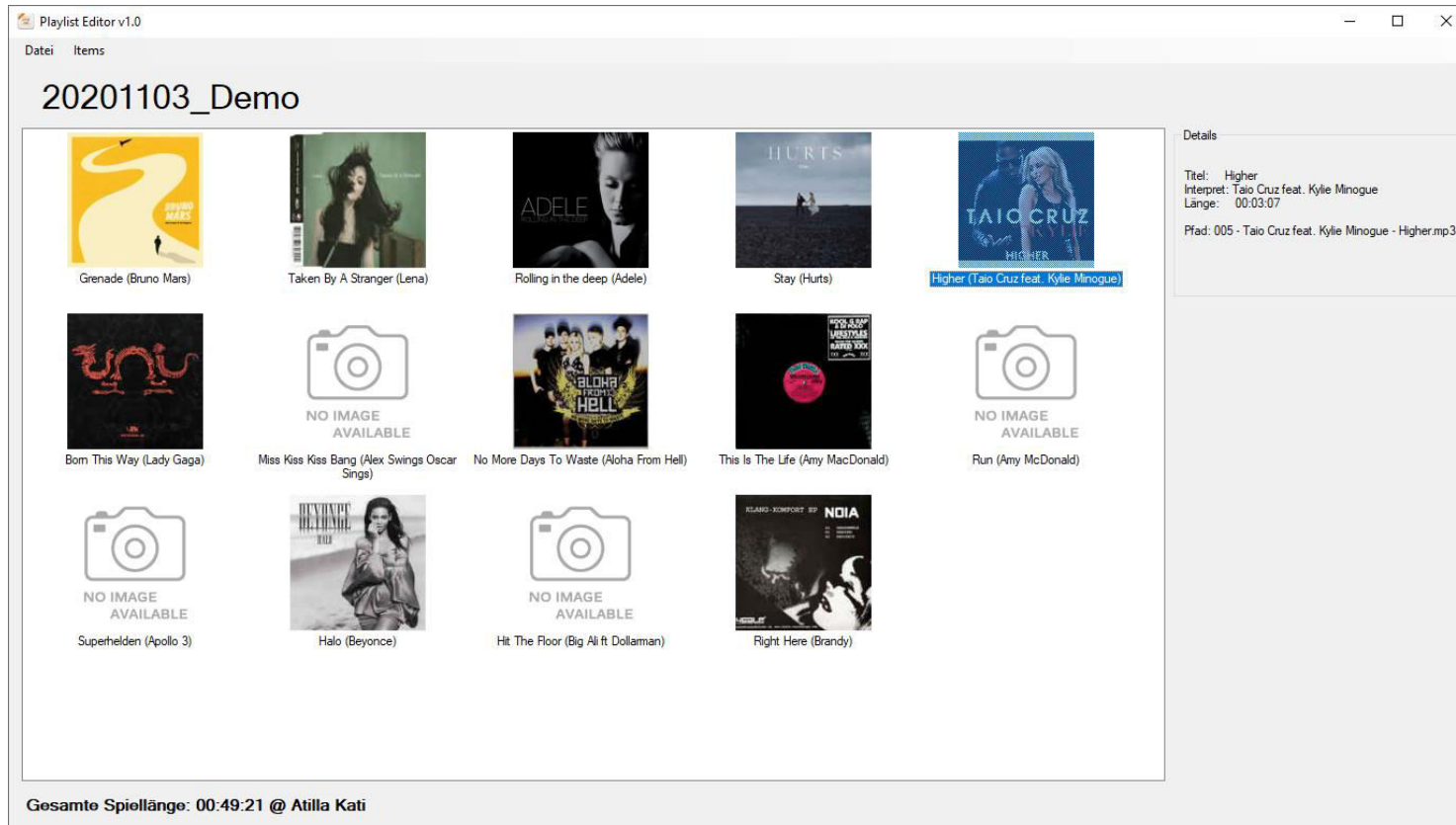


Architektur – Design Patterns

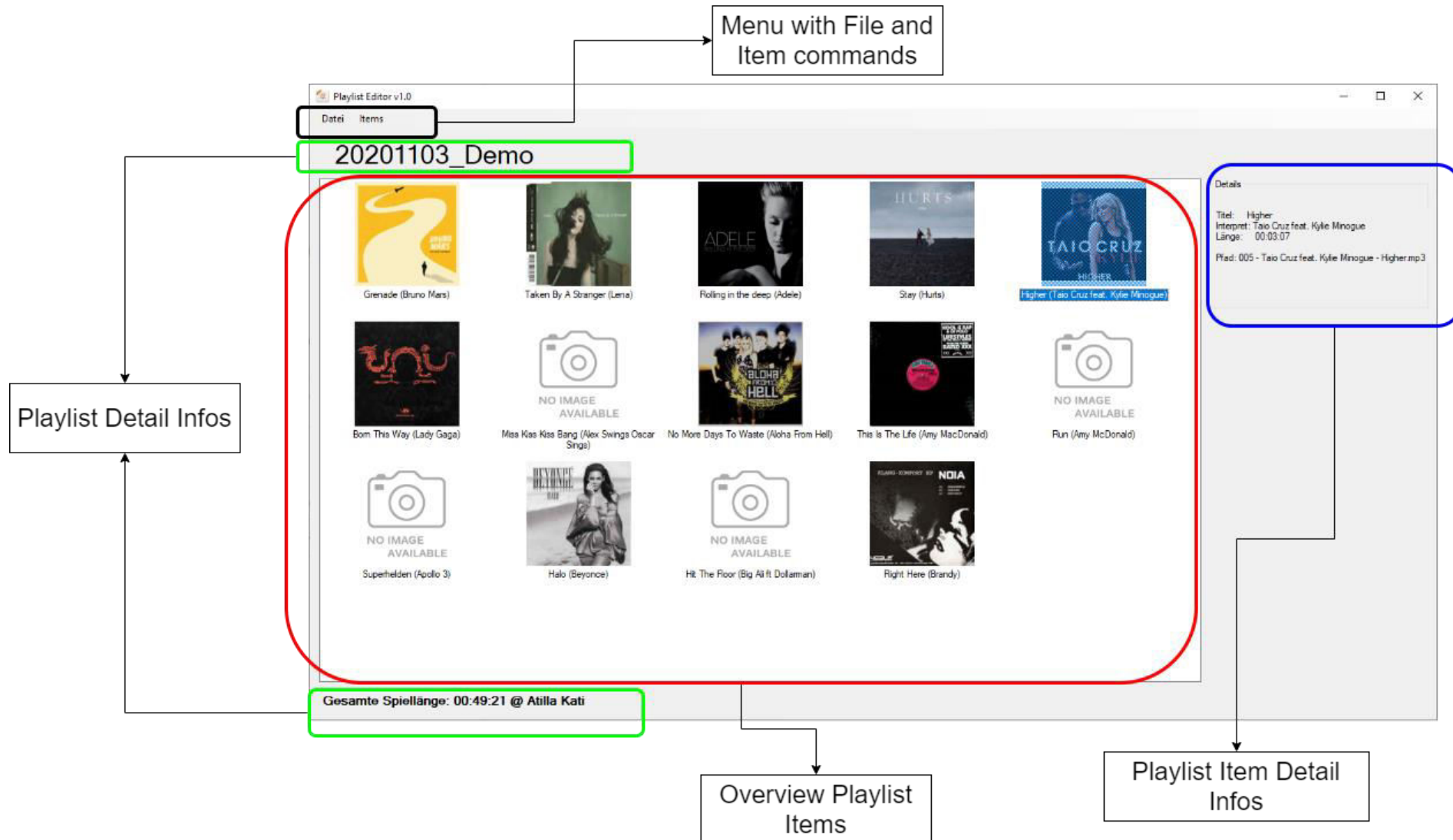
- Was sind Design-Patterns?
 - Basieren auf Entwurfsprinzipien
 - Blaupausen
 - Lösungen zu bestimmten Problemen
 - Wartbarkeit & Wiederverwendung
 - Gemeinsame Sprache

Erzeugungsmuster	Strukturmuster	Verhaltensmuster
Factory Singleton Builder Prototype ...	Adapter Bridge Decorator Facade Flyweight Proxy Composite ...	Template Command Observer Strategy Iterator State Visitor ...

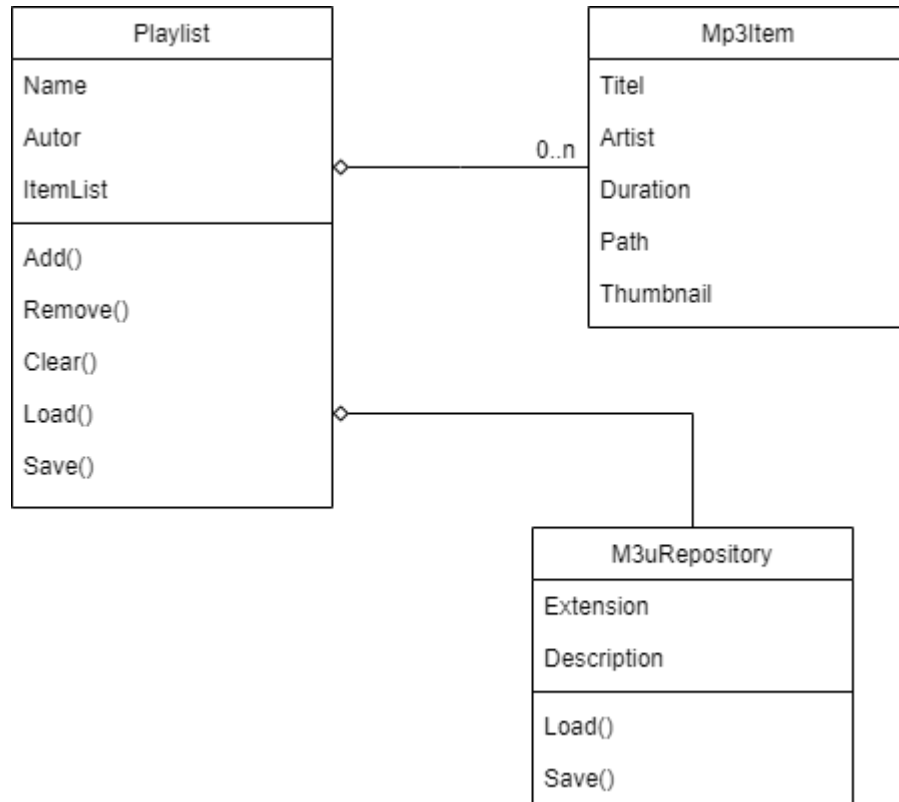
Das Beispiel



Das Beispiel - Requirements

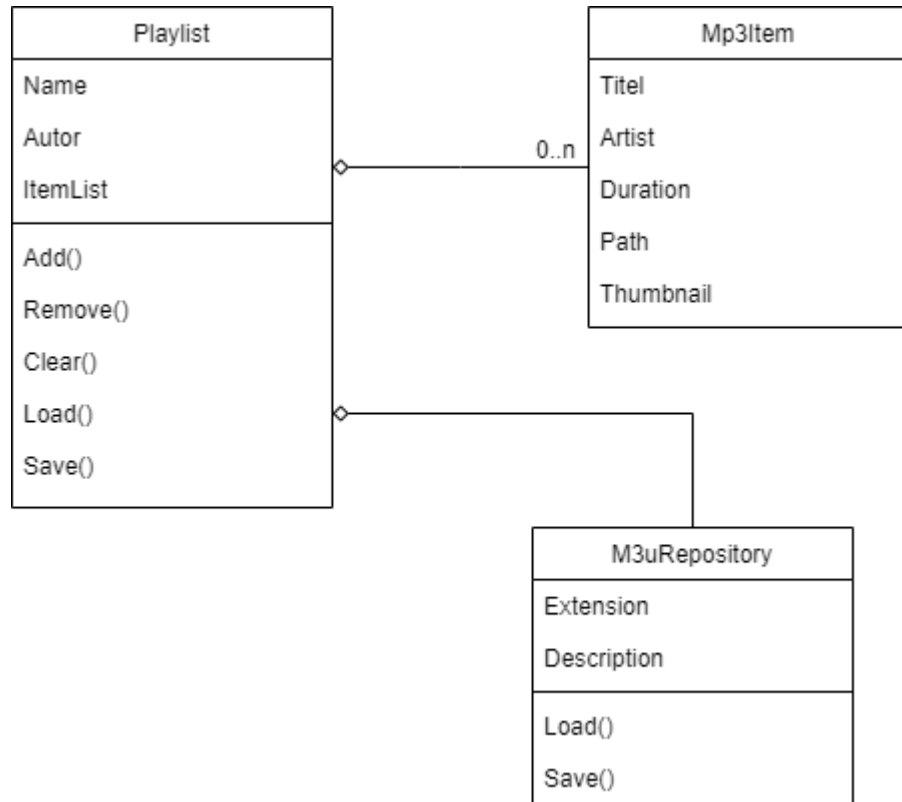


Das Beispiel - Requirements



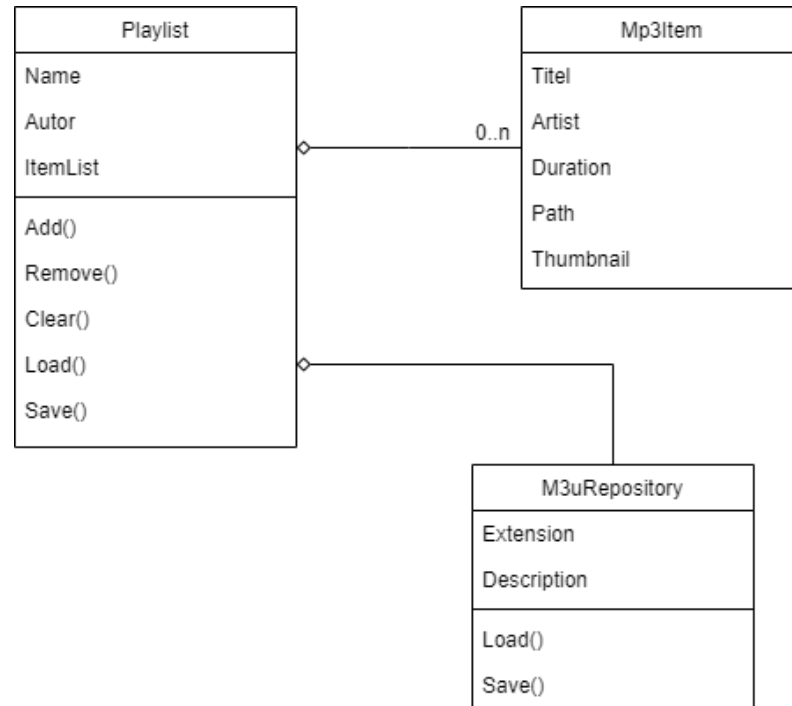
- In einer Playlist sollen auch andere Medien wie Bilder oder Videos verwaltet werden können. Die Typ-Liste soll einfach erweiterbar sein.
- Das Laden und Speichern verschiedener Playlistformate soll möglich sein. Die Typ-Liste soll einfach erweiterbar sein.
- Für PlaylistItems sollen immer Thumbnails angezeigt werden
- Für Details über mp3-Files soll eine Mediendatenbank kontaktiert werden (imdb, discogs)

Das Beispiel - Requirements

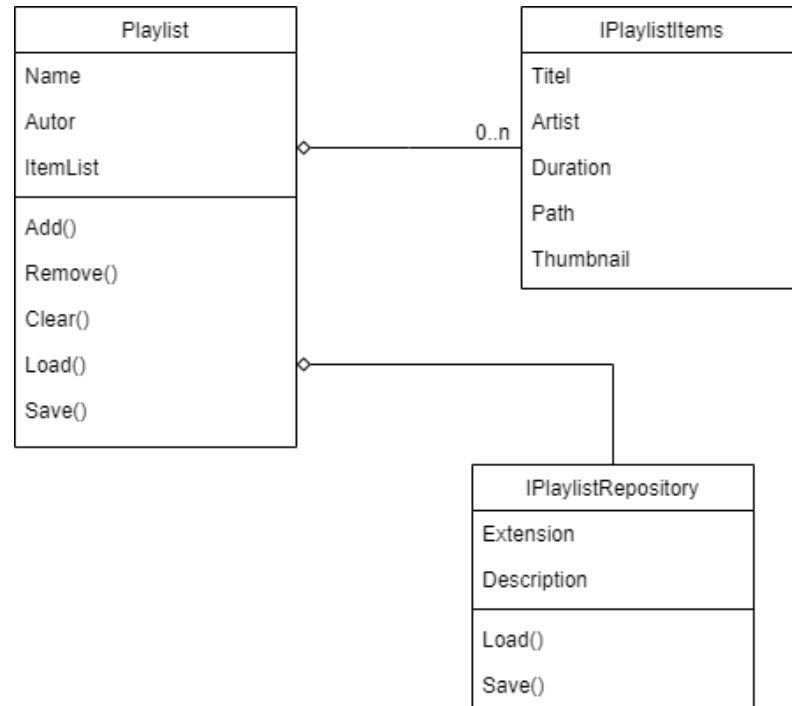


- Bei neuen Item- und Playlist-Typen soll so gut wie nichts bestehendes (Code) geändert werden müssen
- Komponenten sollen einzeln getestet werden können
- Abhängigkeiten (Erzeugung- und Nutzungsabhängigkeiten) sollen reduziert werden

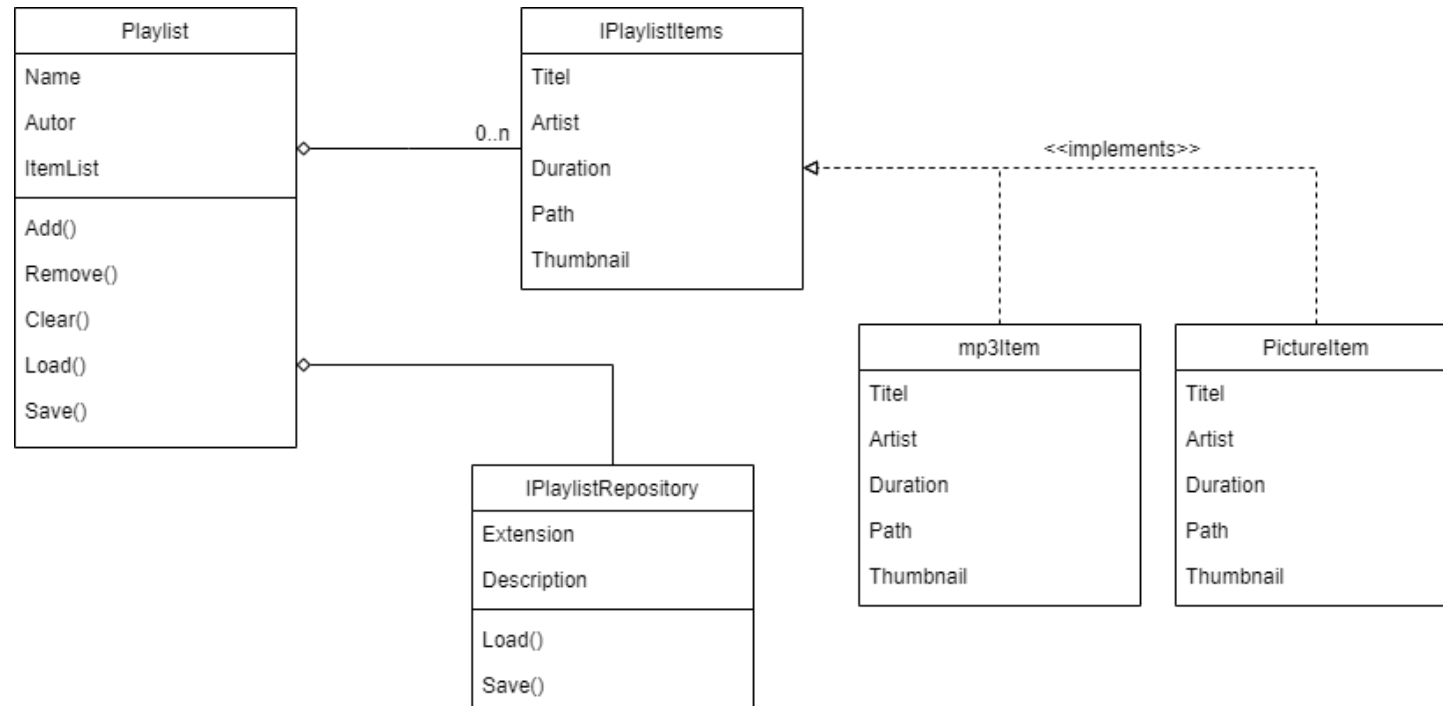
Das Beispiel - Lösung



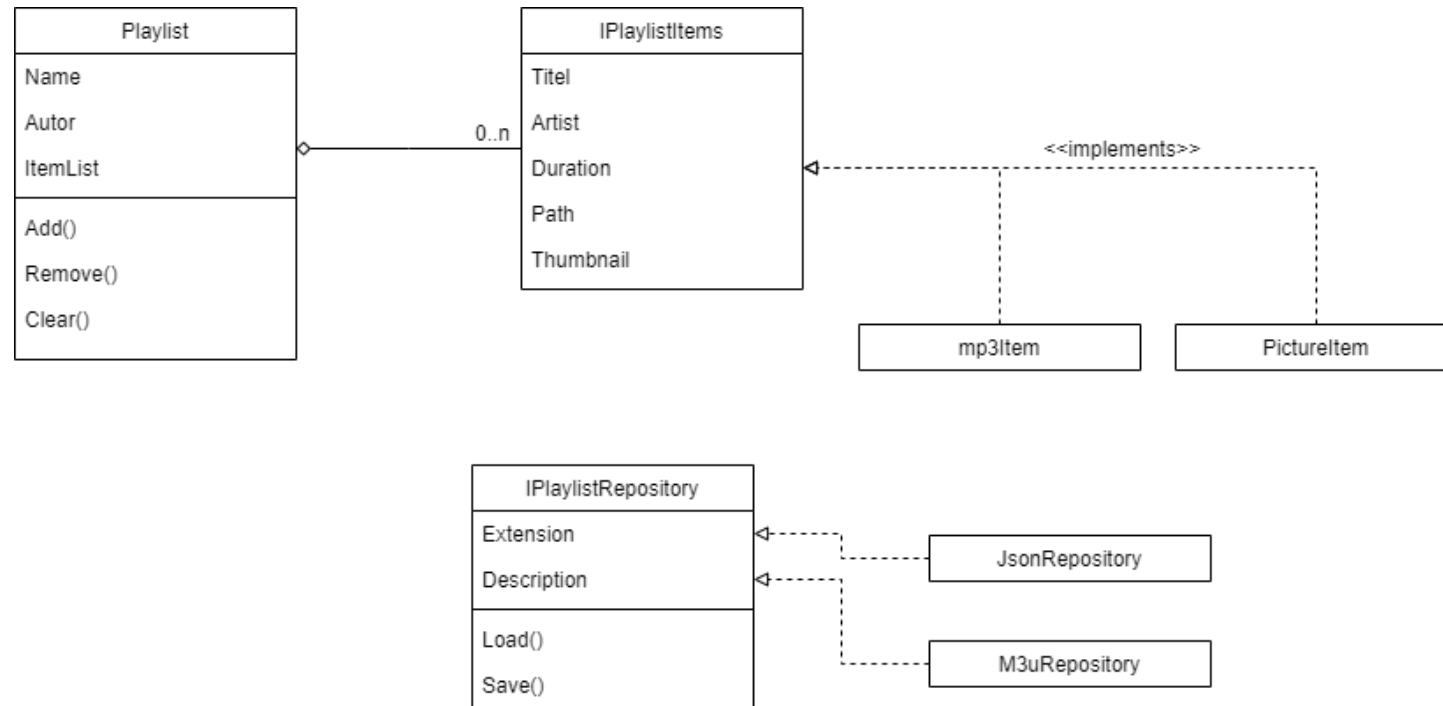
Das Beispiel - Lösung



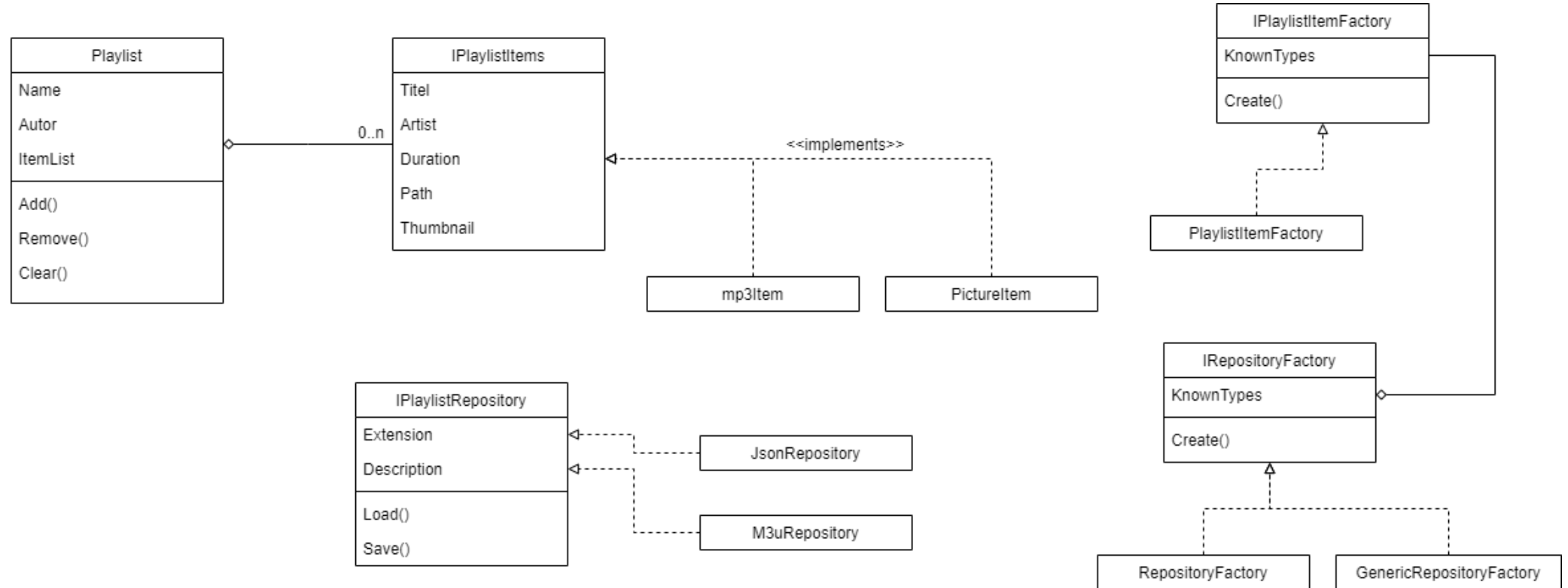
Das Beispiel - Lösung



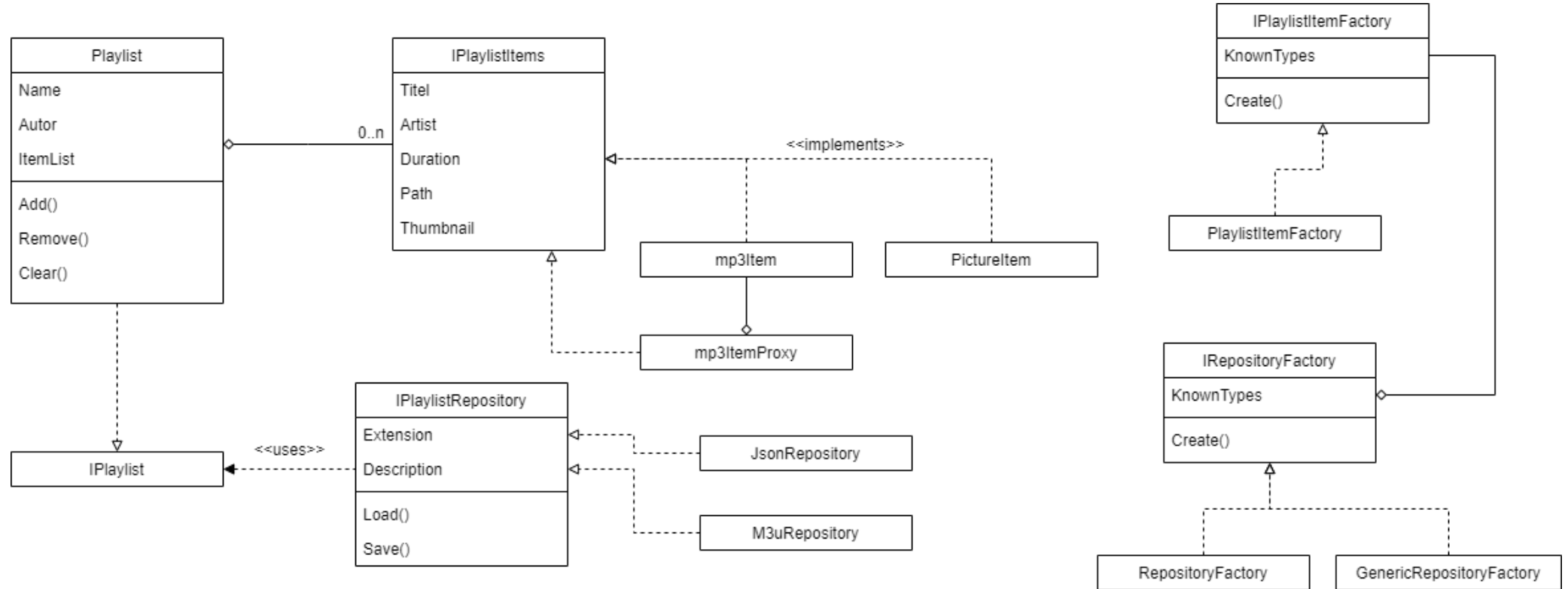
Das Beispiel - Lösung



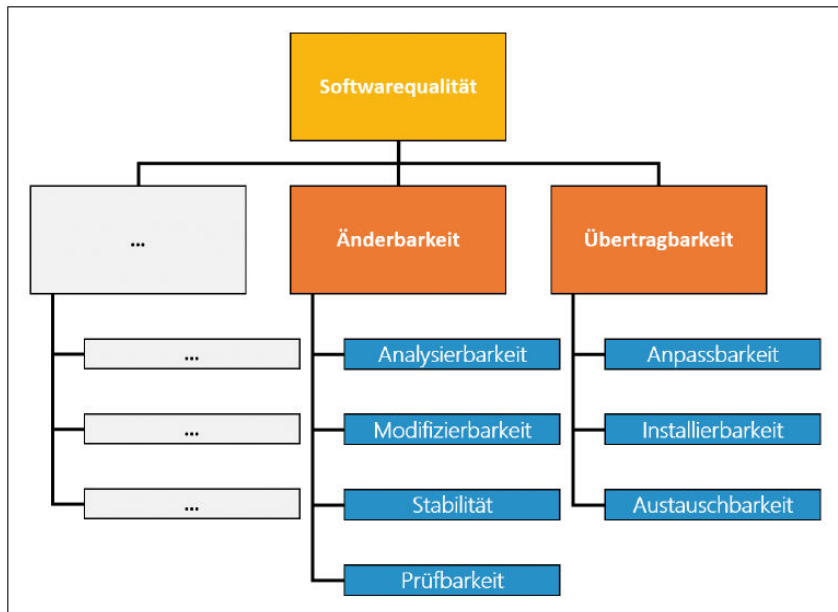
Das Beispiel - Lösung



Das Beispiel - Lösung



Fazit



- Vereinfachung durch
 - Single Responsibility Prinzip
 - Program to abstractions, not implementations
 - Composition over inheritance
 - Inversion of Control
- Auflösung von Abhängigkeiten
 - Strategy Pattern
 - Dependency Injection
 - Factory Method

Danke!

