**Problem & Ziel**

* **Problem:** Viele Beteiligte ⇒ Missverständnisse, unklare Änderungen, Fehler.
* **Ziel:** **Einheitliches Systemverständnis** und klare Abstimmung über Strukturen & Abläufe.

**Was ist UML?**

* **Definition: Unified Modeling Language – Standard** zur **Visualisierung, Spezifikation, Konstruktion, Dokumentation** von Softwaresystemen.
* **Zweck:** Ein Set aus Diagrammtypen, das **Struktur**, **Verhalten** und **Interaktion** eines Systems abbildet.

**Warum modellieren?**

* **Abstraktion:** Komplexes greifbar machen (Komponenten & Beziehungen).
* **Planung & Qualität:** Risiken früh sehen, Lösungen sauber entwerfen.
* **Kommunikation:** Gemeinsame Sprache im Team, weniger Missverständnisse.

**Kerneigenschaften der UML**

1. **Visualisieren:** Grafische Notation macht Architektur & Logik **anschaulich**.
2. **Spezifizieren:** Elemente & Beziehungen **präzise** festlegen.
3. **Konstruieren:** Modelle helfen **Richtung Code** (z. B. Schnittstellen, Klassen).
4. **Dokumentieren:** **Nachhaltige** Basis für Wartung & Weiterentwicklung.

**Anwendungsbereich**

* **Vor der Implementierung:** Anforderungen & Abhängigkeiten **klar** entwerfen.
* **Während/Nach der Umsetzung:** Architektur & Prozesse **abgleichen** und **pflegen**.

**Was kann UML modellieren? (3 Sichten)**

* **Struktur:** Bausteine & Beziehungen (z. B. **Klassen**).
* **Verhalten:** Abläufe & Zustände (z. B. **Aktivitäten**, **Zustandslogik**).
* **Interaktionen:** Wer spricht mit wem, in welcher Reihenfolge (z. B. **Sequenzen**).

**Diagrammgruppen – Überblick mit typischen Beispielen**

* **Strukturdiagramme** (Wie ist es aufgebaut?)
  + **Klassendiagramm:** Domänenobjekte, Attribute, Beziehungen.
  + *Merke:* Gut für **Begriffe & Datenmodell**.
* **Verhaltensdiagramme** (Wie funktioniert es intern?)
  + **Aktivitätsdiagramm:** Prozessfluss mit Entscheidungen/Parallelität.
  + **Zustandsdiagramm:** Lebenszyklus eines Objekts (Zustände/Übergänge).
  + **Use-Case-Diagramm:** Funktionen **aus Nutzersicht** (Systemgrenze, Akteure).
  + *Merke:* Gut für **Abläufe, Regeln, Nutzerziele**.
* **Interaktionsdiagramme** (Wie kooperieren Teile zeitlich?)
  + **Sequenzdiagramm:** Zeitlich geordnete Nachrichten zwischen Objekten/Services.
  + *Merke:* Gut für **API-Aufrufe & Service-Ketten**.