# Brrr brrr patapim #brainrot #tralalerotralala #humor #brrbrrrpatapim | TikTokNumerisch Lösen

# **BISEKTION**

# **Markus Kattner**

# **3BWHII**

# **2024/2025**

# **SWP**

DECKBLATT

**1. Vorgehenmodell = Kanban  
Warum Kanban?**

* Maximale Flexibilität bei sich ändernden Anforderungen
* Einfaches Task-Tracking über 8 Wochen
* Visuelles Management (Notion)
* Ideal für Einzelprojekte ohne aufwendige Planung

**2. Programmierparadigma: Objektorientiert**

**Warum OOP?**

* Gute Strukturierung und Wiederverwendbarkeit
* Erleichtert spätere Erweiterung oder Änderungen
* Besonders bei datenbasierten Systemen sinnvoll

**3. Zeitmanagement + Aufwandsschätzung**

Gesamtdauer: **ca. 8 Stunden** über **8 Wochen**, also im Schnitt

| Anforderungen & Planung | 30min |
| --- | --- |
| PSP erstellen | 20min |
| Architektur & Datenstruktur | 1h |
| Implementierung Kernlogik | 2h |
| Tests & Fehlerbehebung | 1.30min |
| Dokumentation und PM | 2h |
| Puffer | 40min |

**4 .PSP/WBS**

**1. Projektplanung**

* **1.1 Anforderungen erfassen**
  + Analyse der Projektziele
  + Grobspezifikation der Funktionen
* **1.2 PSP & Zeitplanung**
  + Erstellung des Projektstrukturplans
  + Aufwandsschätzung
  + Zeitrahmen mit Meilensteinen
* **1.3 Ressourcen- & Risikomanagement**
  + Identifikation benötigter Ressourcen (HW/SW/Personal)
  + Risikobetrachtung (intern/extern)
  + Maßnahmenplanung zur Risikovermeidung

**2. Implementierung**

* **2.1 Architektur festlegen**
  + Auswahl des Programmierparadigmas
  + Strukturierung der Softwaremodule
* **2.2 Datenstrukturen und Logik**
  + Entwurf und Umsetzung zentraler Algorithmen
  + Einbindung benötigter Bibliotheken
* **2.3 Modultests**
  + Testfälle definieren
  + Durchführung der Tests (Fehleranalyse und Bugfixing)

**3. Abschluss**

* **3.1 Dokumentation**
  + Quellcodedokumentation
  + Anwenderdokumentation (falls nötig)
* **3.2 Lizenzprüfung / Veröffentlichung**
  + Prüfung genutzter Bibliotheken auf Lizenzbedingungen
  + Abklärung Urheberrecht / PSF / Open-Source
* **3.3 Abschlussbericht / Übergabe**
  + Zusammenfassung des Projekts
  + Abgabe der finalen Dateien und Doku

**5. GANTT:**

Woche: 1 2 3 4 5 6 7 8

--------------------------------------------------------------

Planung █████

Architekt ███

Code █████████ █ █

PM ██ ██ █ █

Tests ███ ███

Doku ███████████████████████

Lizenz ██

**6. Ressourcenmanagement**

* **Personal:** ICH (evtl. Review durch Mitschüler)
* **Hardware:** PC/Laptop mit Python
* **Software:**
  + IDE (z. B. VS Code, Sublime Text)
  + GitHub (für version control)
* **Dokumentation:** Word bissl Notion

**7. Risikomanagement**

| **Innere Risiken** | **Maßnahme** |
| --- | --- |
| Zeitmangel durch andere Fächer | Wöchentliche Micro-Ziele setzen |
| Demotivation / Überarbeitung | Realistische Pausen + Puffer einbauen |
| Fehler im Code / Bugs | Frühzeitig testen + Git-Versionierung |
| **Äußere Risiken** | **Maßnahme** |
| PC defekt / Datenverlust | Backups via Git / Cloud |
| Krankheit | Pufferzeit einbauen, notfalls verschieben |

**8. Lizenzprüfung (Veröffentlichung)**

* Nutze nur Open-Source-Module mit erlaubten Lizenzen (MIT, Apache, BSD = ✅)
* Keine GPL mischen, wenn du nicht Open Source veröffentlichen willst
* PSF License (Python Standard) erlaubt kommerzielle Nutzung = ✅
* Keine urheberrechtlich geschützten Inhalte (Logos, fremde Daten etc.)

**AUFGABEN CODEN**

**Aufgabe 1: Wurzelgleichung**

√(n) = x

→ x² - n = 0

**Aufgabe 5 Bisektion:**

c = (a+b)/2

Quadrieren: n = x²

Nullstellenform: f(x) = x² - n = 0

Neues Intervall wählen basierend auf f(c)

**Aufgabe 6 Newton:**

f(x) = x² - n (Wie vorher bei Bisektion)

Lernressource: https://studyflix.de/mathematik/newton-verfahren-1780

**Aufgabe 9: Kettenlinie**

Ausgangsformel:

y(x) = a·cosh((x-x₀)/a) - a + y₀

Randbedingungen:

x₀ = w/2 (Symmetrie)

y(0) = 0 (Mastbasis)

Durchhang s = y(0) - y(w/2)

Umformung:

0 = a·cosh(w/2a) - a + y₀

s = -y₀

→ a·cosh(w/2a) - a - s = 0

Code-Form:

x\*cosh(w/(2x)) - x - s = 0 (s und w in m)

### **Aufgabe 8: Polynomtest (P₄(x) = -x⁴ + 3x³ + x² + 2x)**

**Intervall**: [3, 4]

**Toleranz ε = 10⁻²**: 7 Iterationen → Lösung ≈ 3.4531

**Toleranz ε = 10⁻⁸**: 27 Iterationen → Lösung ≈ 3.4567