**Aufgabe:**

Entwickelt eine Ereignisgesteuerte Prozesskette (EPK) für den gesamten Lebenszyklus der Entwicklung einer Anwendungssoftware. Ihr solltet dabei die folgenden Phasen berücksichtigen:

1. Anforderungsanalyse
2. Entwurf
3. Implementierung
4. Testen
5. Deployment
6. Wartung

Beachtet die verschiedenen Funktionen, Ereignisse und Kontrollflüsse, die innerhalb jeder Phase auftreten können. Identifiziert auch Entscheidungspunkte, an denen alternative Wege eingeschlagen werden könnten, und verwendet die entsprechenden Operatoren (OR, AND, XOR) um diese zu kennzeichnen.

Hier ist eine detailliertere mögliche Lösung für die Aufgabe, strukturiert nach den Hauptelementen der Ereignisgesteuerten Prozesskette:

**1. Anforderungsanalyse**

**- Funktionen:**

- "Anforderungen sammeln"

- "Stakeholder-Interviews durchführen"

- "Technische Machbarkeit prüfen"

**- Ereignisse:**

- "Anforderungskatalog erstellt"

- "Stakeholder-Zustimmung erhalten"

- "Machbarkeitsbericht abgeschlossen"

**- Operatoren:**

- XOR nach "Stakeholder-Zustimmung erhalten", um entweder Änderungen zu bearbeiten oder fortzufahren

**- Kontrollfluss:**

- Vom Ereignis "Anforderungskatalog erstellt" zum Operator XOR und dann entweder zurück zur Funktion "Anforderungen anpassen" oder vorwärts zur Funktion "Entwurf vorbereiten"

**2. Entwurf**

**- Funktionen:**

- "Systemarchitektur entwerfen"

- "Datenmodell entwickeln"

- "Benutzeroberflächen-Prototypen erstellen"

**- Ereignisse:**

- "Designspezifikationen fertiggestellt"

- "Datenmodell genehmigt"

- "UI-Prototypen genehmigt"

**- Operatoren:**

- AND zwischen "Designspezifikationen fertiggestellt" und "Datenmodell genehmigt", um Parallelität zu signalisieren

**- Kontrollfluss:**

- Von "Machbarkeitsbericht abgeschlossen" zu "Systemarchitektur entwerfen"

**3. Implementierung**

**- Funktionen:**

- "Code entwickeln"

- "Datenbank einrichten"

- "Integration von APIs"

**- Ereignisse:**

- "Code-Implementierung abgeschlossen"

- "Datenbank betriebsbereit"

- "API-Integration erfolgreich"

**- Operatoren:**

- AND nach "API-Integration erfolgreich", um die Bereitschaft zum Testen zu signalisieren

**- Kontrollfluss:**

- Vom Ereignis "UI-Prototypen genehmigt" zur Funktion "Code entwickeln"

**4. Testen**

**- Funktionen:**

- "Unit-Tests durchführen"

- "Integrationstests durchführen"

- "Systemtests durchführen"

**- Ereignisse:**

- "Unit-Tests bestanden"

- "Integrationstests erfolgreich"

- "Systemtests abgeschlossen"

**- Operatoren:**

- XOR nach "Systemtests abgeschlossen", um zu entscheiden, ob Fehlerbehebung notwendig ist

**- Kontrollfluss:**

- Von "Code-Implementierung abgeschlossen" zu "Unit-Tests durchführen"

**5. Deployment**

**- Funktionen:**

- "Deployment-Plan erstellen"

- "Anwendung deployen"

**- Ereignisse:**

- "Deployment-Plan genehmigt"

- "Anwendung live geschaltet"

**- Operatoren:**

- Keine spezifischen Operatoren benötigt

**- Kontrollfluss:**

- Vom Ereignis "Systemtests abgeschlossen" zur Funktion "Deployment-Plan erstellen"

**6. Wartung**

**- Funktionen:**

- "Nutzerfeedback sammeln"

- "Updates planen und implementieren"

**- Ereignisse:**

- "Wartungsanforderungen identifiziert"

- "Update erfolgreich ausgerollt"

**- Operatoren:**

- XOR nach "Nutzerfeedback sammeln", um entweder sofortige Updates oder geplante Wartungszyklen zu entscheiden

**- Kontrollfluss:**

- Von "Anwendung live geschaltet" zu "Nutzerfeedback sammeln"

Jede dieser Phasen könnte in der EPK noch weiter detailliert werden, abhängig von der Komplexität des Projekts und der Tiefe.