Zusammenfassung Kalkulation&Wirtschaftlichkeitsanalyse.docx

Probleme der **Ausgangbedingungen**:
Unklare Zielsetzung, unzureichende Mittel, falsche Projektbesetzung, Unkenntnis Ist-Situation, mangelnde Anwenderbeteiligung, fehlendes technisches Know-how, unzulängliche Projektkontrolle
Alternativen (Standardpakete, Cloud-Services, Open-Source-Lösungen) vor eigenen Anwendungen.
Deadlock zwischen obere Führungsriege(Kosten) & untere/Sachbearbeiter(Inhalt)
Möglicher Ausweg Lastenheft, aber unbeliebt da Kosten für Software Ist-Zustand
Urproblem Widerspruch von Planungskonzept & Sichtfeld.
Lösung dafür Agile Softwareentwicklung. (Iterative Realisierung von Sprints von max. 1 Monat)

Teufelsquadrat von Sneed, Mitte (Produktivität), Ecken(Qualität, Funktionalität, Kosten, Zeit)
Produktivität fix(Wachstum zu langsam für Projekt), außen nur 2 Planbar, Rest variabel.
Kosten+Zeit fix = Rückwärtsplanung, Quali+Funk = Vorwärtsplanung
Mit Grenzen professionell abfinden.

Produktivität mit Produktivitätsstudie messen(Größe & Komplexität & Qualität der Systeme)
Bestmöglich Source-Code-Analyse (jährlich wiederholen)
(Zeilen, Anweisungen, Function-/Data-/Object/Story-Points) / Qualitätsfaktor

$$Produktivität = \frac{Quantität \times Qualität}{Aufwand}$$

Produktivität sinkt zur Projektgröße (Bei hoher Wiederverwendungsrate umgekehrter Effekt)
Produktivitätskurve mit links Function-Points, unten Personenmonate. Graph oft flacher am Ende.
Hohe Produktivitätsunterschiede da(Personen, Branchen, Projektgrößen).

Projektproduktivität = Codemengengröße / Summe Personentage(gebucht)
Mehrere Projektproduktivität = Produktivitätskurve (für neue Projekte)

Projektrisiken: Technologie-/,Applikations-/,Personal-/,Performanz-/,Qualitäts-/,betriebliche Risiken
Risikostufe = Risikogewicht * Risikowahrscheinlichkeit; Risikoaussetzung = Max-Verlust * Risikostufe
Risikoaussetzungsfaktor = Risikoaussetzung/Gesamtkosten des Projekts
Nun Gegenmaßnahmen auflisten und Testwerkzeuge & -personal dafür einplanen
Risikominderungsfaktor = Durchschnitt der Erfolgswahrscheinlichkeiten von Gegenmaßnahmen
Risikofaktor = Risikowahrscheinlichkeit * Risikoaussetzung * (1 – Risikominderungsfaktor) + 1

Nutzwert eines Projekts: Nutzen = Umsatz – Betriebskosten oder Betriebskosten – Kostenersparnis
Altsysteme halten oft höherer Nutzwert als Neuentwicklung/Systemablösung
Neuerungen am besten langsam einführen

Value-Driven Software Engineering = welche IT-Projekte guten **Return on Investment (ROI)**
ROI = (Nutzen – Kosten) / (Kosten + Risiken oder Kosten * Risikofaktor) = min 1 für Genehmigung
Projekte nach **Wertbeitrag selektieren**/steuern über jede einzelne Funktion hinterfragen.

Rahmenbedingungen Aufstellung der **Kostenpläne**
Messbarkeit Produktnutzen, Kalkulierbarkeit Projektkosten, Erkennbarkeit Projektrisiken, Vorhersehbarkeit Produktfolgekosten
Nutzenfaktoren: effizentere Geschäftsprozesse, besser motivierte Mitarbeiter, befreite unternehmenskritische Ressourcen, gestiegene Wettbewerbsfähigkeit, innovative Geschäftsprozesse, erweitertes Unternehmensumfeld, zusätzliche Einnahmequellen

Wege zu IT-Produkten: Standardprodukte, Open-Source, Produktsanierung, Neuentwicklung
Erkennbarkeit Projektrisiken: Risiken müssen erkennbar sein, darum ist SAP gefragt, da viele erfahrene Berater existieren
Folgekosten: ROI = (Nutzen - (Einmalige Kosten + Folgekosten)) / (Einmalige Kosten + Risikodaten)
Portfolio-Analyse: Vergleich 2er Projekte, im Nutzenquadrat an den kreuzenden Achsen plaziert
Projektwirtschaftlichkeit: Projekte sollte sich in 5 Jahren amortisieren (Software-Lebensdauer)