

## 1 Einführung in das Projekt

Die Kleinstweich Deutschland GmbH (KWD) plant den Ersatz des klassischen WINDOWS - Taschenrechners durch eine modularisierte Lösung, die branchenabhängig für jeden Kunden individuell konfiguriert wird.

### 1.1 Veranlassung

Unsere Kunden wünschen individuelle, teils sehr spezielle Lösungen anstelle unseres klassischen, im Betriebssystemumfeld integrierten Taschenrechners. Außerdem soll das lästige handschriftliche Zwischenspeichern von Rechenergebnissen entfallen.

### 1.2 Zielsetzung

Erstellung einer modularisierten Softwarelösung mit weitgehend einheitlichem, den Benutzer führenden Bedienkonzept, die uns beliebige Kombinationen von maximal je drei branchenbezogenen Branchenmodulen (Form-\*.dlls einschließlich der zugehörigen Berechnungs-Funktionen) zu erstellen ermöglicht und möglichst wenig redundanten Code enthält, um die Performance hoch und den Ressourcenbedarf gering zu halten. Die Form-Module sind als unabhängig nutzbare Laufzeitbibliotheken einschließlich des funktionalen Codes zu realisieren.

### 1.3 Projektumfeld

AG: Kleinstweich Deutschland GmbH, Sitz Dresden; Softwarehersteller

Aufgrund des Closed-Shop-Betriebes unseres Unternehmens ist die Nutzung unserer betrieblichen technischen und organisatorischen Infrastruktur für die Projektbearbeitung gefordert. Als Entwicklungsbasis sind möglichst Softwareprodukte unserer Tochterunternehmung Microsoft AG zu verwenden. Weitere Kosten entstehen somit nicht.

Die bei KWD gültigen Code-Konventionen sollen eingehalten werden. Der Taschenrechner muss unter WINDOWS-Betriebssystemen als selbstständiges Programm funktionieren, mit individuell, vom AG zu konfigurierender Anbindung von Branchenmodulen, die als Laufzeitbibliotheken beigelegt werden. Zukünftig soll die Taschenrechner-Software mit individuellen Preisen je nach Modulausstattung ausgeliefert werden, um den Preis für den Kunden beeinflussbar und transparent zu gestalten.

### 1.4 Wesentliche Aufgaben

Unsere Qualitätsansprüche fordern die Durchführung des Projektes gemäß der DIN ISO 21500. Wir erwarten die detaillierte Lösungsbeschreibung als Pflichtenheft gemäß VDI/ VDE 3694 und ein agiles Vorgehen, das uns Mitsprache- und Kontrollmöglichkeiten bietet. Letztendlich ist eine ausgereifte Software auszuliefern, die keine Nachbesserungen und Updates erforderlich macht. Sämtliche Fenstermodule der Software sind als "WINDOWS-Form" - gebundene Laufzeitbibliotheken zu erstellen, deren Schnittstellen einheitlich zu gestalten und geeignet zu dokumentieren sind.

### 1.5 Eckdaten

Realisierungszeitraum:	02/2021 - 07/2021	Projektkostenrahmen:	30.000 € (Personalkosten)
Ansprechpartner seitens KWD:	Herr Dammüller		
Email-Adresse des AG:	dammuelle@bszetdd.lernsax.de		

## 2 Beschreibung der Ausgangssituation (Istzustand)

Mit WINDOWS-Betriebssystemen wird ein Taschenrechnermodul ausgeliefert, dessen Bedienung derer marktüblicher, als elektronische Einzelgeräte verfügbarer Taschenrechner gleicht.

Außer den Grundrechenarten verfügt das bisherige System über klassische wissenschaftlich-technische Funktionalitäten, besitzt keine Protokollfunktion und setzt fundiertes Wissen über den benötigten Formelzusammenhang voraus.

## 3 Aufgabenstellung (Sollzustand)

Das geplante neue Modul zwingt den Benutzer zu einer von der Software vorgegebenen, sequentiellen Eingabe der Funktionalparameter, die zur jeweils ausgewählten Funktion gehören. Jeder KWD-Kunde kann aus einem Katalog von zusätzlichen Funktionsmodulen maximal drei auswählen, die individuell in seine Taschenrechner-Applikation durch KWD fest als Laufzeitbibliothek eingebunden werden. Eine dynamische Anbindung der Funktionalmodule ist zur Verhinderung der Einbindung von Raubkopien der Kleinstweich-Module oder von Fremdmodulen ausgeschlossen. Basisausstattung sind die Grundrechenarten gemäß Anforderungen unter 3.2.

### 3.1 Kurzbeschreibung der Aufgabenstellung

Entwicklung und Realisierung einer ausführbaren Programmdatei mit individueller Anbindung von Laufzeitbibliotheken bei Projektablauf gemäß DIN ISO 21500 und mittels objektorientierter Analyse- und Entwicklungstechnologien. Als Programmiersprache ist Microsoft C# mit Windows - Forms und Komponenten aus dem Microsoft .NET Framework zu verwenden. Die Software muss vollständig mit der PC-Maus bedienbar sein, und die Bedienoberflächen sollen in jeder Bedienphase Form-einheitlich hinsichtlich der Schriftgröße und der Schriftart sowie der Hintergrundfarbe vom Kunden anpassbar sein.

### 3.2 Gliederung und Beschreibung der Aufgabenstellung

Anmerkung: Die Spezifikationen \*) sind für **alle** Branchenmodule einheitlich gemäß Beispiel Prozentrechnung zu realisieren.

Teilmodul	Output	Input/ Funktionen	Bemerkungen
Startfenster	Liste der zur Laufzeit durchgeführten gültigen Berechnungen mit Eingabewerten und Ergebnissen	Auswahl der Funktionsbereiche (Start der jeweiligen Branchenmodule) sowie Beendigung des Programms und automatische Löschung der Ergebnisliste	Grundrechenartenmodul standardmäßig eingebunden
Prozentrechnung	<b>Übergabe des Aufgabentextes samt Ergebnissen an das Startfenster *)</b>	<b>Auswahl einer Funktion aus den Branchenfunktionen *)</b> , hier: a) %dazu, b) %weg, c) %davon, d) %Satz, e) Bruttopreis aus Nettopreis, f) Nettopreis aus Bruttopreis	während Parametereingabe ist die Ausführung von Nebenrechnungen zu ermöglichen *)
Kreditberechnung	Kreditbetrag, Zinssatz, Ratenhöhe, Laufzeit in Monaten, ggf. Schlussrate, Zinsen gesamt	a) Kredit mit einmaliger Rückzahlung, b) Ratenkredit, Vorgabe der Laufzeit, c) Ratenkredit, Vorgabe der Ratenhöhe	Berechnungsvorschrift gemäß Kundenvorgabe
Geometrie	Ausgabe von Umfang und Flächeninhalt	a) Dreieck, b) Kreis, c) Parallelogramm	Berechnungsparameter mit dem AG abstimmen
mathematische Funktionen	vgl. Funktionen  c) gemeiner Bruch	a) Fakultät, Quadratwurzel, Potenzfunktion (Eingabewerte € Q) b) Primzahlen zwischen Grenzwerten, c) Umwandlung Dezimalbruch	
Schule	Zeugnisnotenempfehlung, Anzahl und Notendurchschnitt	Sequentielle Eingabe von Noten	
Informationstechnik	a) Grafikspeicher- und Videodateigröße in techniküblicher Dimension von Binär- und Dezimalpräfix b) geg. Zahl im Binär-, Ternär-, Oktal- und Dezimalsystem, c) Datenmengen in b bzw. B	a) Farbtiefe und Breite x Höhe (Pixel), Bilder s <sup>-1</sup> b) Zahlensystemumrechnungen; c) Umrechnung von Datenmengen zwischen Binär- und Dezimalpräfixen	
Grundrechner	Rechenaufgaben und Ergebnisse in Ergebnisliste; wahlweise Übernahme des Ergebnisses als Parameter für andere Module	Ziffern, Dezimalzeichen, Vorzeichen, Klammern (mehrere Ebenen), gesamte Berechnung als Zeichenkette/ Grundrechenarten mit mehreren Operanden und Operatoren einschließlich Klammerrechnung und Einhaltung grundlegender Rechenregeln	Grundrechnermodul ggf. als Nebenrechner verwenden; alle Nebenrechnungen erscheinen in der Ergebnisliste
Eingabemodul	branchenrichtige Bezeichnung des einzugebenden Parameters anzeigen, Rückgabe des gültigen Zahlenwertes für den angeforderten Parameter, ggf. Fehlermeldung	Ziffern, Dezimalzeichen, Vorzeichen als Eingabeparameter für die ausgewählte Funktion; kontextbezogene Verhinderung von Fehleingaben	Gültigkeitsprüfung der Eingabe, auch bezogen auf den Definitionsbereich der ausgewählten Funktion

Anmerkung: Die Spezifikationen \*) sind für **alle** Branchenmodule einheitlich gemäß Beispiel Prozentrechnung zu realisieren.

### 3.3 Ablaufbeschreibung

Die Bedienung ist wie folgt zu realisieren:

Nach Auswahl des Branchenmoduls im Startfenster und nachfolgender Auswahl der gewünschten Funktion im jeweiligen Branchenmodul gibt der Benutzer über das auch für andere Softwareprojekte universell verwendbare Eingabemodul (vgl. Bild rechts), nacheinander und durch die Software zwangsgeführt, alle für die jeweilige Funktion notwendigen Parameter über die Bildschirmtastatur ein. Dabei ist es notwendig, dass mit Hilfe des Grundrechenmoduls Nebenrechnungen ausführbar sind, deren Ergebnis wahlweise als Funktionsparameter übernommen wird. Nach jeder Berechnung kehrt das Programm in das Startfenster zurück, das eine Auflistung der zur Laufzeit des Rechnermoduls erledigten Berechnungen einschließlich der Nebenrechnungen anzeigt.

Zu jeder Zeit soll es dem Benutzer möglich sein, die anwendungseinheitliche Darstellung aller Fensterelemente dem Sehvermögen und Geschmack des Benutzers anzupassen.

**Eingabemodul**

---

Kreditbetrag:

1

2

3

4

5

...

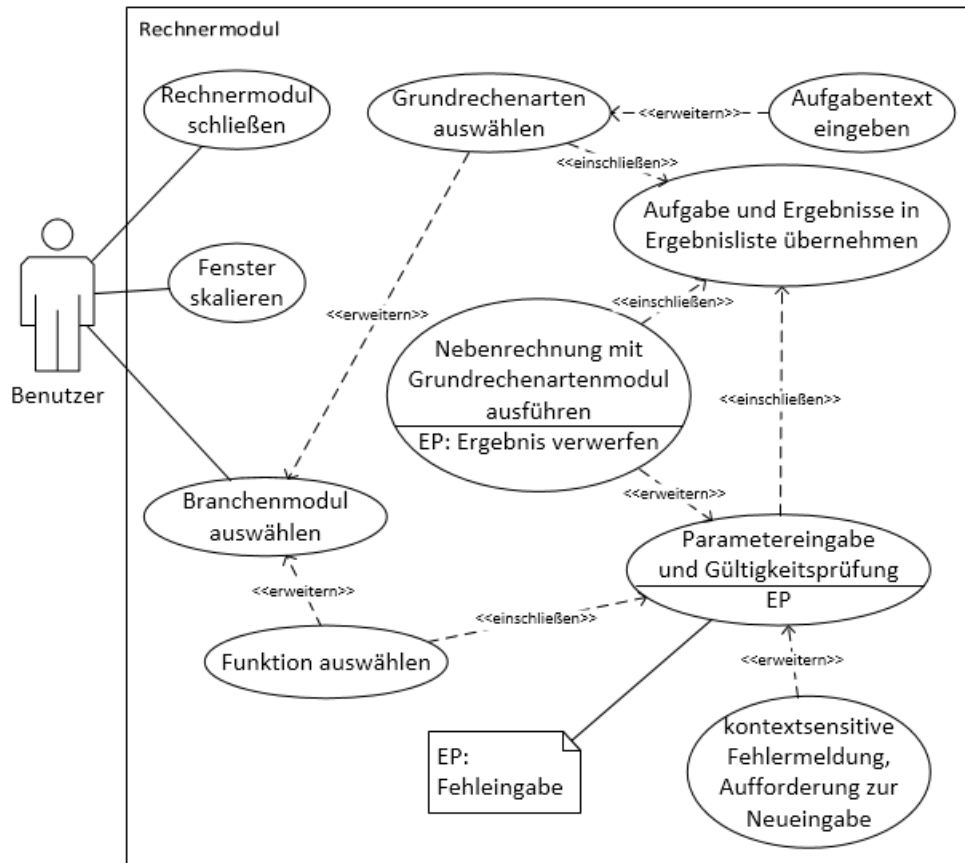
Die Anzeige der absolvierten Berechnungen muss sowohl die Rechenoperation, ihre Parameter und die Ergebnisse anzeigen, z. B.:

$$"2 + 4 \times 2,6 - 2 \times (0,5 - 2) = 15,4"$$

oder

"Ratenkredit: 12000 €, Zinsen 10 %, Laufzeit 12 Monate  $\Rightarrow$  Rate 1050,17 €, Zinsen gesamt 602,04 €".

Fehleingaben und Rechnungsabbrüche sollen nicht protokolliert werden. Die Bedienung ist im folgenden Anwendungsfalldiagramm veranschaulicht:



### 3.4 Datendarstellung (Sollzustand)

Alle Berechnungen werden einschließlich aller oben geforderten Ergebniswerte, eingegebener Parameter und Nebenrechnungen protokolliert. Grundsätzlich sind alle Ergebnisse und Eingabeparameter auf sechs signifikante Ziffernstellen mathematisch richtig zu runden. Das trifft nicht auf finanzmathematische Berechnungen zu; dort muss regelmäßig mit mindestens sechs Nachkommastellen gerechnet und das Ergebnis währungsrichtig gerundet werden.

Die Reihenfolge der Berechnungen im Protokoll soll der Reihenfolge der Bearbeitung entsprechen.

### 3.5 Zukunftsaspekte

Die Ausbaubarkeit und Nachhaltigkeit der Lösung ist durch ein modulares Fensterkonzept, die Realisierung der Fenstermodul-Implementationen als \*.dll - Dateien einschließlich ihrer individuellen Branchen-Funktionen und der geeigneten Offenlegung aller Modul-Schnittstellen gewährleistet. Der AG behält sich die Präzisierung der Aufgabe, auch die Ergänzung der Software durch extern bereitgestellte WINDOWS-Form-Module gemäß Schnittstellenbeschreibung laut PH, vor.

## 4 Schnittstellen

### 4.1 Schnittstellenübersicht

Externe Schnittstellen existieren zwischen Anwendung und Betriebssystem, Mensch und PC/Betriebssystem (Grafische Benutzerschnittstelle GUI) und zwischen .NET Framework und Anwendung. Die Anwendung realisiert intern Schnittstellen zu proprietären, eigenen Laufzeitbibliotheken.

### 4.2 Anwendung - Betriebssystem

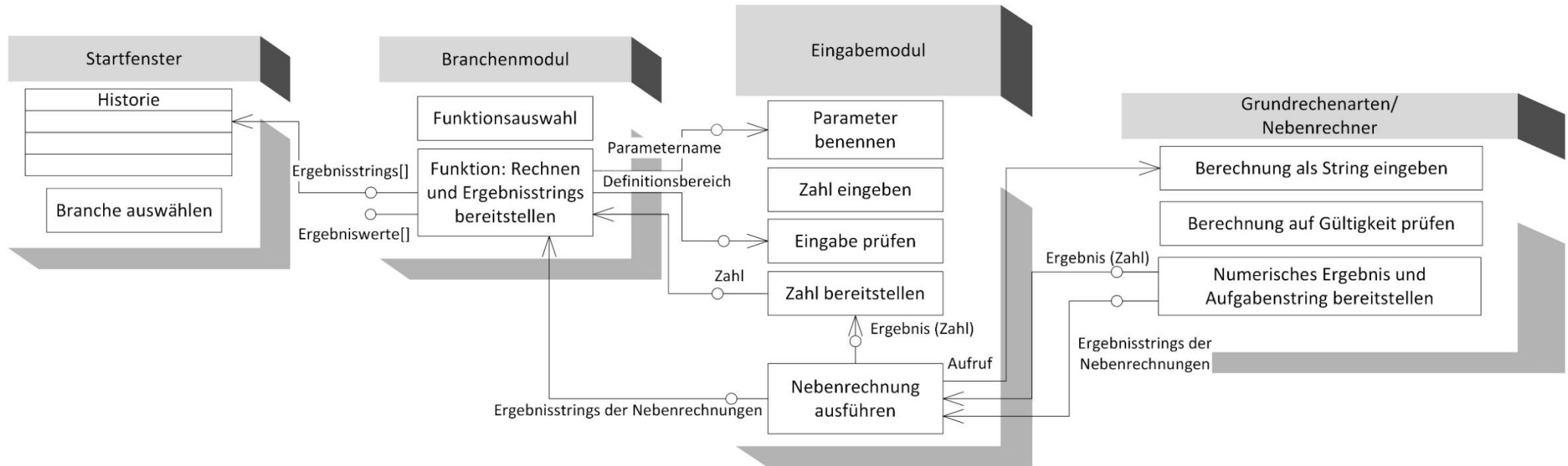
Die Anwendung ist als ausführbare .EXE - Datei zzgl. der Laufzeitbibliotheken unter WINDOWS zu realisieren.

### 4.3 Mensch - PC/ Betriebssystem

Windows-Forms - Dialogkomponenten auf Basis eines .NET - Frameworks; zwangsgeführte Dialoge gemäß polnischer Notation, soweit zweckmäßig; Zahleneingabe über die Grafische Bedienoberfläche möglich.

#### 4.4 Anwendungsprogramm - Laufzeitmodule

Die Teilmodule gemäß 3.2 sind als visuell vererbte Forms als Dynamic Linked Libraries (.dlls) mit integrierter Funktionalität zu realisieren. Es darf immer nur ein Fenster aktiv bedienbar sein, und die Modulaufrufe sind ausschließlich modal zu gestalten. Die Verteilung der Funktionen und die geforderten Schnittstellen zeigt folgendes Diagramm; die realisierten Schnittstellentypen sind im Rahmen eines Design-Klassendiagramms für die weitergehende Verwendung verbindlich zu spezifizieren.



#### 5 Anforderungen an die Systemtechnik

Die Systemtechnik ist ressourcensparend und zahlenmäßig konkret vom AN zu spezifizieren. Die Schnittstellen der Laufzeitbibliotheken sind verbindlich im Pflichtenheft per Design-Klassendiagramm zu deklarieren.

#### 6 Anforderungen für die Inbetriebnahme und den Einsatz

##### 6.1 Dokumentation

Die Entwicklungskosten der Einzelmodule sind dem AG zwecks Preisbildung im Projektstrukturplan zur Verfügung zu stellen.

##### 6.2 Instandhaltung und Softwarepflege

Mit der Bereitstellung der Module als visuell vererbte Forms ist der Entwicklungsprozess abgeschlossen.

#### 7 Anforderungen an die Qualität

##### 7.1 Software-Qualität

Die Anwendung ist mit C# zu implementieren. Die Blackbox-Test - Parameter für die Einzelmodule sind dem AG als Bestandteil des Pflichtenheftes vorzulegen. Universell nutzbare Funktionsbibliotheken für Rechenmethoden sind zu vermeiden; die Berechnungsmethoden sind an die jeweilige partielle Formklasse zu binden. Das GUI muss designeinheitlich gestaltet und als Prototyp vom AG abgenommen und bestätigt werden.

Die Funktionalmethoden sollen aus Performancegründen proprietär implementiert werden; die Verwendung der Funktionsbibliotheken der Math-, DataTable- bzw. weiterer Bibliotheksklassen ist ohne Zustimmung des AG nicht zulässig. Der systemtechnische Ressourcenbedarf ist ansonsten permanent so gering wie möglich zu halten. Zentralisierte abstrakte- oder Schnittstellen-Klassen sollen nur Eigenschaften und Methoden enthalten, die **alle** Branchenmodule nutzen.

##### 7.2 Hardware-Qualität

Die Anwendung muss auf marktüblichen Office-PC-Systemen funktionieren.

## 8 Anforderungen an die Projektabwicklung

Das Projekt ist gemäß DIN ISO 21500 unter Nutzung agiler Entwicklungsmethoden durchzuführen. Sämtliche elektronischen Projektdaten sind i. S. des Closed-Shop-Betriebes möglichst mit Microsoft-Software zu erstellen und in den dem AN bereitgestellten Dateiablagen der entsprechenden LernSax-Projektgruppe abzulegen. Der AN ist verpflichtet, Produkt und Know-How gegen unberechtigten Zugriff zu sichern. Verstöße gegen diese Regel ziehen einen Minuspunkt für den Projektleiter nach sich. Die Risiken aus der eventuellen Nutzung von Cloud-Diensten oder eines Versionskontrollsystems trägt ausschließlich das Team des AN zu gleichen Teilen.

### 8.1 Projektorganisation

Es ist ein Projektleiter sowie ein davon unabhängiger Product-Owner zu benennen. Die Projektplanung und -kontrolle kann mit Microsoft Project oder ProjectLibre erfolgen. Dafür ist ein geeigneter Projektkalender zu erstellen, in dem ein abgerechneter Arbeitstag eine schulische Projektwoche mit je sechs Zeitstunden Projekt-Arbeitszeit repräsentiert. Der AG akzeptiert höchstens 100 € Personalkosten je Mitarbeiterstunde. Dem AG ist jederzeit Einsicht in die Projektplandatei und auf den Erfüllungsstand zu gewähren. Ein Vorgangs- und ein mitarbeiterbezogener Blockwochen - Personalkostenplan ist zu erstellen. Vom AG geforderte Dokumente sind blockwochenweise vorzugsweise papiergebunden sowie transparent und gut strukturiert in den LernSax-Dateiablagen der Projektgruppe bereitzustellen. Jedes elektronische Dokument enthält im Dateinamen den BSZET-Klassen- und den Teamnamen sowie den Hinweis auf den Inhalt.

### 8.2 Projektdurchführung

Meilensteine zum GUI - Prototyping, für eine inkrementelle Realisierungsstufe und den Projektabschluss einschließlich der jeweils anfallenden Projektkosten und der Termine sind im Pflichtenheft zu fixieren. Die Realisierung der Funktionalmodule ist entsprechend der vom Kunden zu bestätigenden Priorisierung zu realisieren.

Es ist ein Projekttagbuch mit schultäglicher Einteilung in der Gliederung:

**Projektwoche + Tag • Aktivitäten/Arbeitspakete - Bearbeiter - Zeitaufwand • Entscheidungen/Probleme** zu führen.

Dresden, am 08. Februar 2021

Auftraggeber:



#### Was erwarten die Lehrer von jeder Projektgruppe?

Zahlen in () = Abgabe-Blockwoche, jeweils freitags;  
Zahlen in [] = Ziel-Punktzahl für die jeweilige Aufgabe

- Projektantrag mit rückseitiger Begründung (1)[10]
- Projektstrukturplan in Bauplanansicht mit Verantwortlichkeiten für die Implementierung und voraussichtlichen Kosten (2)[15]
- Risikoanalyse mit mind. zehn projektbezogenen Risiken aus vier Tätigkeitsbereichen (2)[20]
- Projekttagbuch (Kategorien: s. o.) (taggenau aktuell!) [min. 2 x 5]
- jeder Mitarbeiter ist an Implementation beteiligt [20...max. 30]
- Projektplan einschließlich Personaleinsatz fürs Gesamtprojekt, gern als Microsoft-Project- oder ProjectLibre-Datei (3)[20]
- komplettes analyt. Klassendiagramm (4)[20], passend zum bestätigten GUI-Prototypen der proprietären Module und
- optional davon getrenntes Design-Klassendiagramm (4)[+x]
- Prototyping des GUI anlässlich Meilensteinberatung mit dem Kunden (3)[Prototyp: mind. 15, Beratung: 12, Protokoll: 8]
- Testliste für alle Programmfunktionen, mit prüfbaren Beispieldaten (4)[10] (Product-Owner als Stabsstelle für die ganze Klasse)
- In jeder Projektwoche ab Mai 2021 muss dem Kunden ein Mehrwert übergeben werden können (6)[ggf. 12 + Protokoll: 8]
- Projekt- und Releaseabnahme (7)[15 + Protokoll 10 + ggf. Powerpoint-Präsentation]
- Projektleiter [10](5...7), Product Owner; Letzterer verantwortet die Produkteigenschaften gemäß Pflichtenheft zur Abnahme, führt die Release-Abnahmen durch und verantwortet die Richtigkeit der Black-Box-Tests(s. o.).
- geheftete Projektdokumentation(mittels Aktendulli) und offengelegte Software mit proprietärem Quellcode (8)[Doku: 5]
- Nachleistungen am Quellcode gemäß Festlegungen aus der Abnahme, abschließende Quellcodebewertung (6...8)[20...30]

#### Zur Benotung:

1. Jeder Schüler muss während der Planungs- und während der Realisierungsphase jeweils Aufgaben mit einer Gesamt-Sollpunktzahl von mind. 30 bearbeiten, um zwei Noten mit je 15 Sollpunkten bilden zu können.
2. Erreicht er Anspruch 1. nicht, werden Quantitätsnoten aus den erreichten Punkten, bezogen auf 30, gebildet.
3. Bei Punktteilen von mehr als 150% pro Schüler, gemessen am Sollpunkte-Klassendurchschnitt, wird eine der erreichten Noten wegen Mängeln der Selbst- und Sozialkompetenz um einen Grad heruntergestuft.
4. Verschwinden Dateien, die nicht in den LernSax-Team-Dateiablagen abgespeichert wurden, wird dem Projektleiter jeweils ein Punkt abgezogen.
5. Während der Planungsphase kann je Gruppe und Blockwoche eine abzugebende Leistung u. U. um eine Projektwoche verschoben werden (Woche 1 - 4). Voraussetzung dafür ist eine **vorherige** Abstimmung mit dem Kunden einschließlich Begründung. Das Gespräch ist zu protokollieren.