

## Übungen zu Projektmanagement (Lösungsvorschlag)

Die angegebenen Lösungen sind nicht als „Musterlösung“, sondern nur als Diskussionsvorlage zu verstehen.

### Aufgabe 4-1

### Produktivitätsmetriken

- a) Nennen Sie mindestens vier wichtige Produktivitätsfaktoren sowie Eigenschaften des hergestellten Codes, die in der Produktivitäts- und Aufwandsmetrik LoC nicht korrekt berücksichtigt werden.

**Lösung:**

- Menge der durch den Code hergestellten Funktionalität
- Qualität des Codes
- Performance des Codes
- Wartbarkeit des Codes
- Zuverlässigkeit des Codes
- Wiederverwendbarkeit des Codes
- Programmiersprache, in die der Code geschrieben wurde
- Andere Entwicklungsphasen, die zur Erzeugung des Codes notwendig sind (Ein SW-Architekt oder ein Tester z.B. hat eine LoC-Produktivität von 0)
- Techniken zur automatischen Generierung (MDD Techniken insbesondere)

- b) Welche dieser Faktoren und Eigenschaften werden von der Funktionspunkt- und Objektpunkt-Metrik korrekt berücksichtigt?

**Lösung:** Die Menge der durch den Code hergestellten Funktionalität.

- c) Suchen Sie äquivalente Metriken wie LoC und Funktionspunkte für die Bewertung der Komplexität von Wissenschaftlichen Arbeiten (z.B. Bachelorarbeiten, Masterarbeiten, Wissenschaftliche Artikel) und der Produktivität von Wissenschaftlern.

**Lösung:** LoC: Länge des Textes (Seiten, Zeilen), UFP: Schwierigkeitsgrad der Theorie, Komplexität der Programmierung, etc. Korrekturfaktoren können eigene Interesse an und Erfahrungen in dem Themengebiet, Qualität der Betreuung usw. umfassen.

- d) Welche Produktivität hat ein Projektmitarbeiter, der ein Programm refaktoriert oder optimiert und dabei das Programm kürzt? In LoC? Und in Funktionspunkten?

**Lösung:** LoC: negative Produktivität, FP: Null Produktivität.

- e) Sollten Produktivitätsmetriken genutzt werden, um Projektmitarbeiter zu bewerten bzw. zu loben?

**Lösung:** Nein, da viele äußerst wichtige Faktoren des Codes durch diese Metriken nicht berücksichtigt werden. D.h. es gibt bisher keine objektive Metrik, nach der sich die Produktivität eines Programmierers bewerten lässt.

### Aufgabe 4-2

### COCOMO-Modelle

Welche Ausprägungen des COCOMO-Modells gibt es, und unter welchen Bedingungen sind sie einsetzbar? Auf welchen Metriken zur Systemkomplexität basieren diese Modelle?

**Lösung:** **COCOMO I:** Ursprüngliches COCOMO-Modell, ist wegen seiner Einfachheit nur mehr für Lehrzwecke empfohlen. Verwendet KDSI (k delivered source instructions), Berechnung aus Funktionspunkten.

**COCOMO II Application Composition Model / Early Prototype Model:** Dient zur Schätzung von Prototypen. Verwendbar, falls das System aus wiederverwendbaren Komponenten, Skripten und Datenbankprogrammierung erstellt wird.

Grundlage: Objektpunkte / Applikationspunkte

Type	Simple	Moderate	Complex
Screens	1	2	3
Reports	2	5	8
DB Support Modules	10		

“This model simplistically assumes that there is no additional effort involved in reuse” [1, S. 628], wodurch die Schätzung sehr ungenau wird. Beispiele einfacher, dennoch schwer wiederverwendbarer Software-Modelle in Form von UML-Zustandsmaschinen finden sich unter anderem in [2].

**COCOMO II Early Design Model:** Vereinfachtes Modell für die initiale Schätzung nachdem Requirements festgelegt wurden.

**COCOMO II Reuse Model:** (*Nicht prüfungsrelevant in dieser Vorlesung*) Wird verwendet, um zusammen mit dem Post-Architecture Model den Aufwand der Wiederverwendung und Codegenerierung zu schätzen. Grundlage: Wiederverwendete oder erzeugte LoC

**COCOMO II Post-Architecture Model:** Detaillierteste Aufwandsschätzung nach dem Architektur-entwurf. Grundlage: LoC.

#### Aufgabe 4-3

#### Projektaufwandsschätzung

Ein Software-Manager ist verantwortlich für die Entwicklung eines Systems, das ein medizinisches Radiotherapie-Gerät zur Behandlung von an Krebs erkrankten Patienten steuern soll. Dieses System wird als eingebettete Software betrieben, und muss auf einem Spezialprozessor mit einer festen Speichermenge von 8 MB laufen. Die Maschine kommuniziert mit einer externen Patientendatenbank, um Patientendaten zu laden und Details zur Behandlung (Datum, Strahlungsmenge, etc.) zu speichern.

Zur Aufwandsschätzung für dieses System wurde die COCOMO-Methode benutzt, die eine Schätzung von 26 Personenmonaten ergab. Die Multiplikatoren für Kostentreiber wurden dabei alle auf 1 gesetzt.

- a) Erklären Sie, warum diese Schätzung angepasst werden sollte um projektspezifische, personalspezifische, produktspezifische und organisationsspezifische Faktoren zu berücksichtigen.

##### Lösung:

- In diesem Projekt existieren viele spezifische Faktoren mit wahrscheinlich großer Auswirkung auf den Aufwand.
- Einige (!) dieser Faktoren sind offensichtlich bekannt.
- Validität der Schätzung würde sich durch Berücksichtigung dieser Faktoren verbessern.
- Ethische Verantwortung gegenüber Kunde (keine absichtlich falschen Daten liefern) und auch den Patienten gegenüber (falsche Schätzung führt zu mangelnder Zeit, führt zu schlechter Qualität, führt zu Gefährdung von Menschenleben).

- b) Schlagen Sie vier Faktoren vor, die die ursprüngliche Schätzung stark beeinflussen würden, und schlagen Sie Werte für diese Faktoren vor. Begründen Sie Ihre Auswahl.

##### Lösung:

- Verlangte Zuverlässigkeit, Extra High: 1, 40
- Größe der Datenbank, High: 1, 16
- Komplexität, High: 1, 15
- Speicher-Einschränkungen, High: 1, 56

- c) Berechnen Sie aus der gegebenen Schätzung eine neue, welche die vier Faktoren berücksichtigt.

**Lösung:** Faktoren:  $1, 40 \cdot 1, 16 \cdot 1, 15 \cdot 1, 56 = 2, 91$

Personenmonate:  $26 \cdot 2, 91 = 75, 66$

## Aufgabe 4-4

## Funktionspunktanalyse

In dieser Aufgabe sollen Sie berechnen, wie viele Funktionspunkte eine Prozessunterstützungssoftware für einen kleinen Verlag umfasst. Die Software soll die Erfassung und Koordinierung von Herstellungs- und Vertriebsprozessen vereinfachen. Das System besteht aus folgenden sechs Komponenten: Stammdatenverwaltung, Rechnungsverwaltung, Auftragsverwaltung, Lagerverwaltung, Betriebsmittelverwaltung und Projektverwaltung.

- a) Überlegen Sie, welche Eingaben, Ausgaben, Interaktionen, Datenbestände und Referenzdaten das System besitzt.

### Lösung:

- Stammdatenverwaltung: Stammdaten erstellen (SE), updaten (SU), abfragen (SA)
- Rechnungsverwaltung: Rechnung erstellen, updaten, abfragen (RE, RU, RA)
- Auftragsverwaltung: Auftrag erstellen, updaten, abfragen (AE, AU, AA)
- Lagerverwaltung: Lagerbestand abfragen, update (LA, LU)
- Betriebsmittelverwaltung: Auslastung abfragen, update (BA, BU)
- Projektverwaltung: Projekt erstellen, abfragen, update (PE, PA, PU)
- Zusätzlich Datenbestand: Stammdaten, Rechnungen und Aufträge, Lager, Betriebsmittel und Projekte

- b) Ermitteln Sie die Komplexität der möglichen Eingaben, Ausgaben, Interaktionen, Datenbestände und Referenzdaten, und berechnen Sie daraus die Anzahl an zu implementierenden Funktionspunkte.

**Lösung:** FTR: File Type Referenced (Data Structures / DB tables referenced)

*External Inputs (EI)*

FTR	Data Elements		
	1 – 4	5 – 15	> 15
< 2	Low	Low	Average
2	Low	Average	High
> 2	Average	High	High

*External Outputs (EO) and External Inquiries (EQ)*

FTR	Data Elements		
	1 – 5	6 – 19	> 19
< 2	Low	Low	Average
2 – 3	Low	Average	High
> 3	Average	High	High

*Function points weighting*

SE	Eingabe	Einfach	3
SU	Eingabe	Einfach	3
SA	Ausgabe	2× Einfach	2 · 4
RE	Eingabe	Komplex	6
RU	Eingabe	Einfach	3
RA	Ausgabe	2× Komplex	2 · 7
AE	Eingabe	Mittel	4
AU	Eingabe	Mittel	4
AA	Ausgabe	2× Mittel	2 · 5
LU	Eingabe	Einfach	3
LA	Ausgabe	Einfach	4
BU	Eingabe	Einfach	3
BA	Ausgabe	Einfach	4
PE	Eingabe	Komplex	6
PU	Eingabe	Komplex	6
PA	Ausgabe	2× Komplex	2 · 7
	Datenbestand	5× Einfach +1× Mittel	45
Summe			140

## Literatur

- [1] Ian Sommerville. *Software Engineering*. Addison-Wesley, 8<sup>th</sup> edition, 2004.

- [2] Gefei Zhang, Matthias Hölzl, and Alexander Knapp. Enhancing UML State Machines with Aspects. In Gregor Engels, Bill Opdyke, Douglas C. Schmidt, and Frank Weil, editors, *Proc. 10<sup>th</sup> Int. Conf. Model Driven Engineering Languages and Systems (MoDELS'07)*, volume 4735 of *Lect. Notes. Comp. Sci.*, pages 529–543. Springer-Verlag, 2007.