















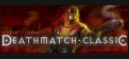






Anwendungsbeispiel: Function Point Verfahren

← → STORE LIBRARY		STEAMOS + LINUX		VIEW	
GAMES		★	☁	STATUS	METASCORE
					LAST PLAYED
	Amnesia: The Dark Descent			Not installed	85
	Antichamber			Not installed	82
	Aquaria			Not installed	82
	Audiosurf 2		☁	Not installed	76
	The Banner Saga		☁	Not installed	80
	Bastion		☁	Not installed	86
	BioShock Infinite		☁	Not installed	94
	Borderlands 2		☁	Not installed	89
	Braid		☁	Not installed	90
	Broforce		☁	Not installed	83
	Chaos on Deponia		☁	Not installed	78
	Counter-Strike			Not installed	88
	Counter-Strike: Global Offensive		☁	Not installed	83
	Counter-Strike: Source		☁	Not installed	88
	Darkest Dungeon		☁	Not installed	84
	Day of Defeat			Not installed	79
	Dead Island		☁	Not installed	80
	Deathmatch Classic			Not installed	
	Deponia		☁	Not installed	74

Anwendung der FP-Methode

Ausgangslage: Lastenheft

Aufbau und Funktion eines Lastenhefts – Beispiel 7(10)

1. Zielbestimmung

Das Programm DAMPF soll eine Internet-Vertriebsplattform für Computerspiele werden. Es soll möglich sein Computerspiele über die Plattform zu beziehen, welche von den Spieleherstellern der Spiele hochgeladen werden.

2. Produkteinsatz:

Das Produkt wird von einer großen Anzahl an unterschiedlichen Benutzern an unterschiedlichen Arbeitsplätzen mit verschiedenen Konfigurationen bedient werden. ...

3. Produktfunktionen:

- /LF10/ Hochladen, Änderung und Löschung von Computerspielen.
- /LF20/ Ersterfassung, Änderung und Löschung von Computerspielgenres.
- /LF30/ Ersterfassung, ... von Accounts für Benutzer.
- /LF40/ Ersterfassung, ... von Accounts für Spielehersteller.
- /LF50/ Anzeigen der auf der Plattform verfügbaren Computerspiele. ...

Lasten- und Pflichtenheft

Aufbau und Funktion eines Lastenhefts – Beispiel 8(10)

4. Produktdaten:

- /LD10/ Folgende Daten sind zu jedem Spiel zu speichern: Name, Genre, Preis, Hersteller, Beschreibung.
- /LD20/ Folgende Daten sind zu jedem Benutzer-Account zu speichern: Benutzername, Email, bevorzugte Kontaktmöglichkeit, bevorzugte Bezahlmöglichkeit, Name, Alter.
- /LD30/ ...

5. Produktleistungen:

- /LL10/ Bei der Ausgabe der Funktion LF50 werden zunächst nur die ersten n "Treffer" ausgegeben. Weitere Treffer werden nur auf Wunsch ausgegeben.
- /LL20/ Das System erzwingt die regelmäßige Erstellung von Datensicherungen für die Daten LD20.
- /LL30/ Maximal werden 5000 Spiele gespeichert.
- /LL40/ Die Bearbeitung eines Computerspiels (LF10) dauert nicht länger als 10 Sekunden.

Lasten- und Pflichtenheft

12

Aufbau und Funktion eines Lastenhefts – Beispiel 9(10)

6. Qualitätsanforderungen:

Produktqualität	sehr gut	gut	normal	irrelevant
Funktionalität		x		
Zuverlässigkeit	x			
Benutzbarkeit	x	x		
Effizienz			x	
Änderbarkeit			x	
Portierbarkeit				x

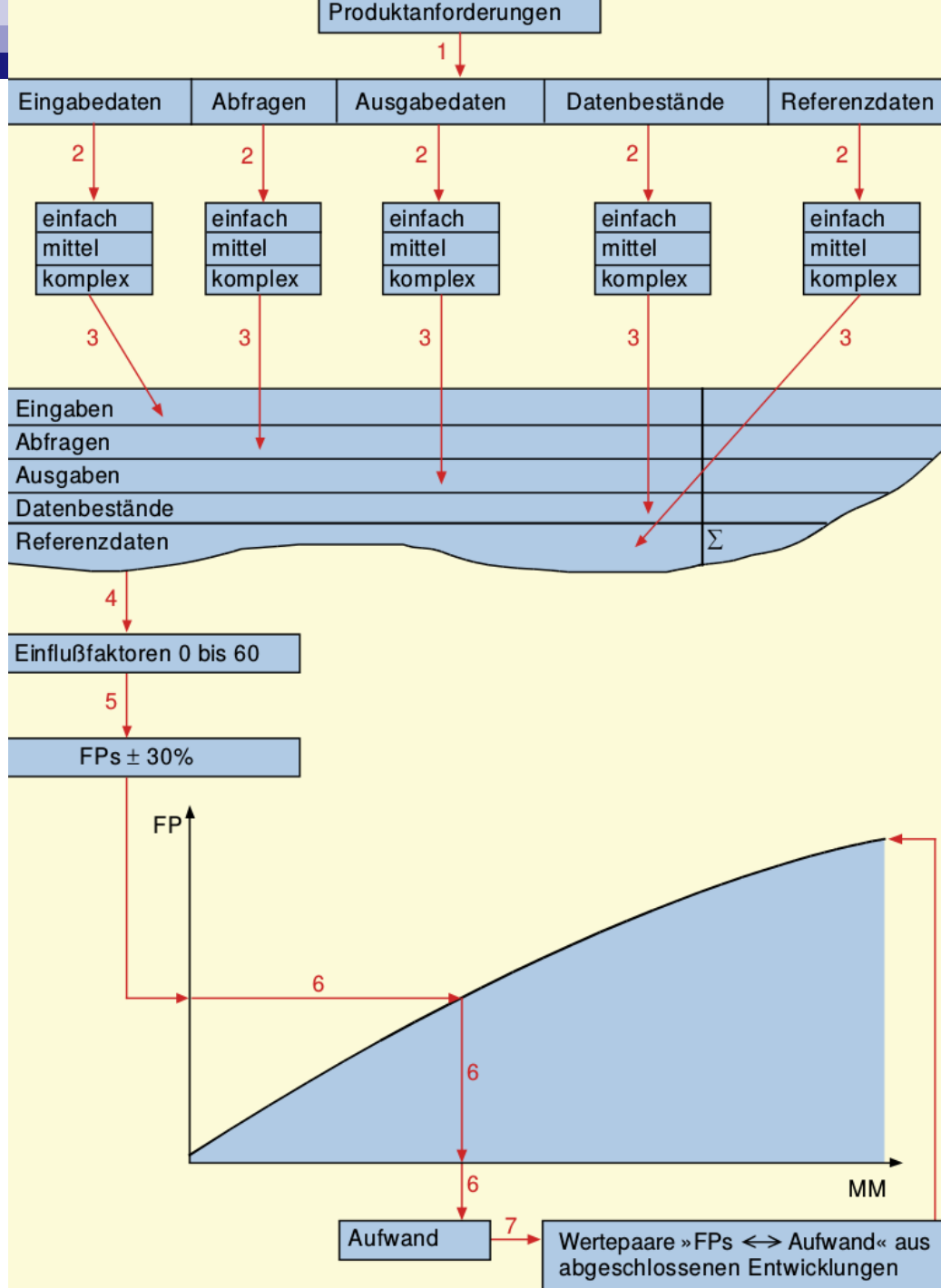
Die Benutzbarkeit der Funktionen mit dem die Benutzer arbeiten müssen sehr gut sein, da ein breites Spektrum an Nutzern angesprochen wird. Die Benutzbarkeit aller übrigen Funktionen muss gut sein, da sie von einem kleineren Klientel bedient wird (Spielehersteller und VAULT-Mitarbeiter).

7. Ergänzungen (wie z.B. Abgrenzungskriterien):

Buchhaltungsfunktionen (z.B. Erstellung von Abrechnungen zwischen VAULT und Spieleherstellern) gehören nicht zum Leistungsumfang.

Lasten- und Pflichtenheft

Kostenschätzung - Anwendung¹³



1. Schritt: Kategorisierung für jede Anforderung

2. Schritt: Klassifizierung jeder Anforderung

3. Schritt: Eintrag der jeweiligen Anzahl in das Berechnungsformular und Ermittlung der unadjusted FPs

4. Schritt: Bewertung der Einflussfaktoren

5. Schritt: Berechnung der adjusted FPs

6. Schritt: Ablesen des Aufwandes

7. Schritt: Aktualisierung der Wertepaare, Neuberechnung der Aufwandskurve

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

Aufbau und Funktion eines Lastenhefts – Beispiel 7(10)

1. Zielbestimmung

Das Programm DAMPF soll eine Internet-Vertriebsplattform für Computerspiele werden. Es soll möglich sein Computerspiele über die Plattform zu beziehen, welche von den Spieleherstellern der Spiele hochgeladen werden.

2. Produkteinsatz:

Das Produkt wird von einer großen Anzahl an unterschiedlichen Benutzern an unterschiedlichen Arbeitsplätzen mit verschiedenen Konfigurationen bedient werden. ...

3. Produktfunktionen:

/LF10/	Hochladen, Änderung und Löschung von Computerspielen.
/LF20/	Ersterfassung, Änderung und Löschung von Computerspielgenres.
/LF30/	Ersterfassung, ... von Accounts für Benutzer.
/LF40/	Ersterfassung, ... von Accounts für Spielehersteller.
/LF50/	Anzeigen der auf der Plattform verfügbaren Computerspiele. ...

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LF 10/ Hochladen, Änderung und Löschung von Computerspielen

- Es handelt sich um drei getrennte Eingaben (Hochladen, Änderung, Löschung)
 - Hochladen ist am umfangreichsten
 - Wahrscheinlich mehr als 10 Datenelemente zu erfassen
 - Auf die Datenbank muss schreibend zugegriffen werden
 - Die Ansprüche an die Bedienerführung sind hoch (z.B., feldweises Editieren, Konsistenzprüfungen)
 - ➔ komplexe Eingabe

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

□ Änderung

- Es wird lesend und schreibend auf Datenbank zugegriffen
- Die Ansprüche an die Bedienerführung sind normal
- Die Anzahl der zu ändernden Datenelemente kann von gering bis hoch reichen
- → mittlere Klassifizierung

□ Löschen

- Erfordert logische Überprüfung (z.B. wie werden Bibliotheken die mit dem zu löschenden Spiel verknüpft sind behandelt) und einen Datenbankzugriff
- → mittlere Klassifizierung

- Ergebnis: 1 komplexe, 2 mittlere Eingaben

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LF 20/ Ersterfassung, Änderung und
Löschung von Computerspielgenres

- Analog wie /LF 10/
- → 1 komplexe, 2 mittlere Eingaben

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LF 30/ Ersterfassung, Änderung und
Löschung von Accounts für Benutzer

- Analog wie /LF 10/
- → 1 komplexe, 2 mittlere Eingaben

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LF 40/ Ersterfassung, Änderung und
Löschung von Accounts für Spielehersteller

- Analog wie /LF 10/
- → 1 komplexe, 2 mittlere Eingaben

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LF 50/ Anzeigen der auf der Plattform verfügbaren Computerspiele

- Es handelt sich um eine Ausgabe
 - Kombination von ComputerspieleDaten mit Standardtexten (z.B. Titel: Playerunknowns Battlegrounds)
 - Aber: /LL 10/ sagt: „... die ersten „n“ Treffer ausgegeben. Weitere Treffer werden nur auf Wunsch ausgegeben.“
- ➔ 1 komplexe Ausgabe

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

Aufbau und Funktion eines Lastenhefts – Beispiel 8(10)

4. Produktdaten:

- /LD10/ Folgende Daten sind zu jedem Spiel zu speichern: Name, Genre, Preis, Hersteller, Beschreibung.
- /LD20/ Folgende Daten sind zu jedem Benutzer-Account zu speichern: Benutzername, Email, bevorzugte Kontaktmöglichkeit, bevorzugte Bezahlungsmöglichkeit, Name, Alter.
- /LD30/ ...

5. Produktleistungen:

- /LL10/ Bei der Ausgabe der Funktion LF50 werden zunächst nur die ersten n "Treffer" ausgegeben. Weitere Treffer werden nur auf Wunsch ausgegeben.
- /LL20/ Das System erzwingt die regelmäßige Erstellung von Datensicherungen für die Daten LD20.
- /LL30/ Maximal werden 5000 Spiele gespeichert.
- /LL40/ Die Bearbeitung eines Computerspiels (LF10) dauert nicht länger als 10 Sekunden.

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LD 10/ Folgende Daten sind zu jedem Spiel zu speichern: Name, Genre, Preis, Hersteller, Beschreibung.

- 1 zusammengesetzter Schlüssel (Name und Hersteller)
- 1 Fremdschlüssel (Hersteller)
- Anzahl unterschiedlicher Datenelemente < 20
- → 1 mittlerer Datenbestand

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LD 20/ Folgende Daten sind zu jedem Benutzer-Account zu speichern:
Benutzername, Email, bevorzugte
Kontaktmöglichkeit, bevorzugte
Bezahlungsmöglichkeit, Name, Alter.

- Analog zu /LD 10/
- ➔ 1 mittlerer Datenbestand

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

Aufbau und Funktion eines Lastenhefts – Beispiel 8(10)

4. Produktdaten:

- /LD10/ Folgende Daten sind zu jedem Spiel zu speichern: Name, Genre, Preis, Hersteller, Beschreibung.
- /LD20/ Folgende Daten sind zu jedem Benutzer-Account zu speichern: Benutzername, Email, bevorzugte Kontaktmöglichkeit, bevorzugte Bezahlungsmöglichkeit, Name, Alter.
- /LD30/ ...

5. Produktleistungen:

- /LL10/ Bei der Ausgabe der Funktion LF50 werden zunächst nur die ersten n "Treffer" ausgegeben. Weitere Treffer werden nur auf Wunsch ausgegeben.
- /LL20/ Das System erzwingt die regelmäßige Erstellung von Datensicherungen für die Daten LD20.
- /LL30/ Maximal werden 5000 Spiele gespeichert.
- /LL40/ Die Bearbeitung eines Computerspiels (LF10) dauert nicht länger als 10 Sekunden.

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LL 10/ Bei der Ausgabe der Funktion LF50 werden zunächst nur die ersten n "Treffer" ausgegeben. Weitere Treffer werden nur auf Wunsch ausgegeben.

- Schon in /LF 50/ mit eingepflegt

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LL 20/ Das System erzwingt die regelmäßige Erstellung von Datensicherungen für die Daten LD20.

- /LD 20/ ist ein mittlerer Datenbestand
- ➔ mittlere Ausgabe

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LL 30/ Maximal 5000 Spiele werden gespeichert.

- Schon in /LD 10/ mit bedacht

Anwendung der FP-Methode

Beispiel: Kategorisierung und Klassifizierung

/LL 40/ Die Bearbeitung eines Computerspiels /LF 10/ dauert nicht länger als 10 Sekunden.

- Schon in /LF 10/ mit bedacht

Anwendung der FP-Methode

Beispiel: Berechnung unadjusted FP

Typ	Komplexität			
	Einfach	Mittel	Komplex	Summe
Eingabedaten	$0 \cdot 3 = 0$	$8 \cdot 4 = 32$	$4 \cdot 6 = 24$	56
Ausgabedaten	$0 \cdot 4 = 0$	$1 \cdot 5 = 5$	$1 \cdot 7 = 7$	12
Abfragen	$0 \cdot 3 = 0$	$0 \cdot 4 = 0$	$0 \cdot 6 = 0$	0
Datenbestände (Anwenderdaten)	$0 \cdot 7 = 0$	$2 \cdot 10 = 20$	$0 \cdot 15 = 0$	20
Referenzdaten	$0 \cdot 5 = 0$	$0 \cdot 7 = 0$	$0 \cdot 10 = 0$	0
	Unadjusted Function Point (UFP)			88
	Value Adjustment Factor (VAF)			
	Adjusted Function Points (UFP * VAF)			

Anwendung der FP-Methode

Beispiel: Berechnung TDI für VAF

Faktor	Wert
Datenkommunikation	3
Verteilte Funktionen	3
Leistungsanforderungen	3
Belastung der Hardware	4
Verlangte Transaktionsrate	4
Online Dateieingabe	2
Effiziente Benutzerschnittstelle	5
Online-Datenänderungen	3

Faktor	Wert
Komplexe Verarbeitungen	1
Wiederverwendbarkeit	2
Einfache Installation	2
Einfach Benutzbarkeit	4
Installation an mehreren Orten	4
Änder- und Erweiterbarkeit	4
Summe der Faktoren (TDI)	44

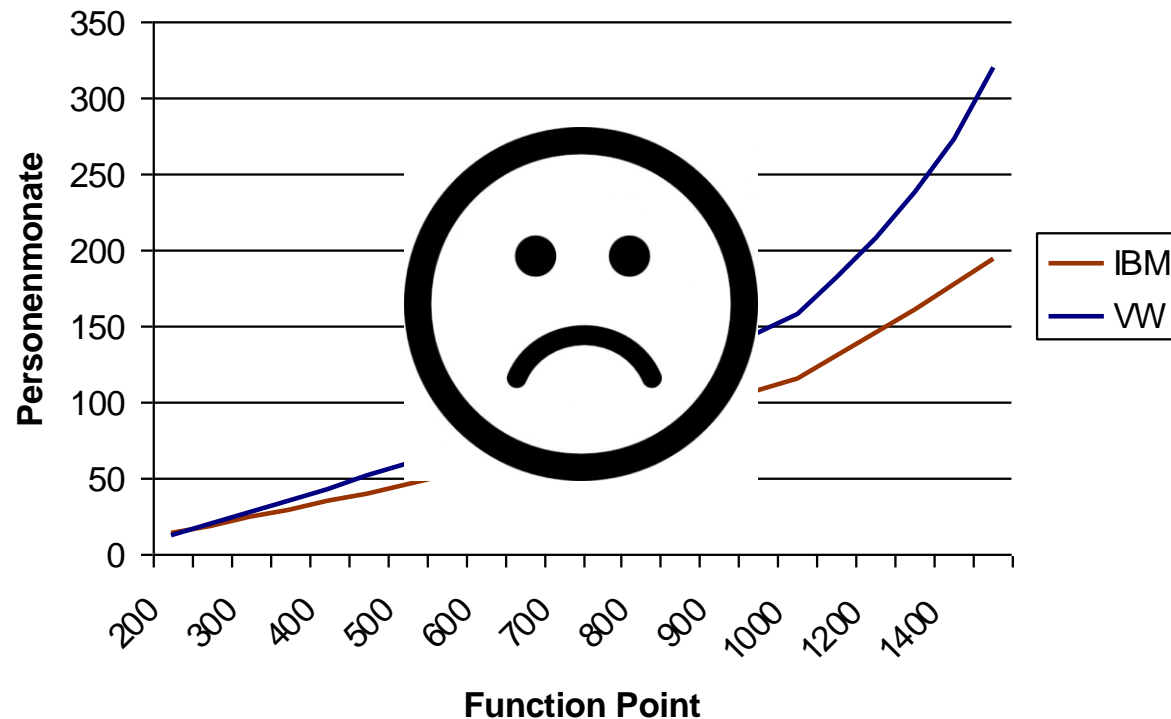
Anwendung der FP-Methode

Beispiel: Berechnung VAF und AFP

Typ	Komplexität			
	Einfach	Mittel	Komplex	Summe
Eingabedaten	$0 \cdot 3 = 0$	$8 \cdot 4 = 32$	$4 \cdot 6 = 24$	56
Ausgabedaten	$0 \cdot 4 = 0$	$1 \cdot 5 = 5$	$1 \cdot 7 = 7$	12
Abfragen	$0 \cdot 3 = 0$	$0 \cdot 4 = 0$	$0 \cdot 6 = 0$	0
Datenbestände (Anwenderdaten)	$0 \cdot 7 = 0$	$2 \cdot 10 = 20$	$0 \cdot 15 = 0$	20
Referenzdaten	$0 \cdot 5 = 0$	$0 \cdot 7 = 0$	$0 \cdot 10 = 0$	0
	Unadjusted Function Point (UFP)			88
	Value Adjustment Factor (VAF)			1,09
	Adjusted Function Points (UFP * VAF)			95,92

Anwendung der FP-Methode

Beispiel: Ablesen des Aufwandes



Anwendung der FP-Methode

Beispiel: Ablesen des Aufwandes

- Werte an Tabelle abgelesen
 - 100 FP ungefähr 5,6 PM (IBM-Tabelle)

Anwendung der FP-Methode

Beispiel: Aktualisierung der Wertepaare

- Nach Projekterledigung war klar, dass insgesamt 4,2PM benötigt wurden
- ➔ Zufügen von dem Wertepaar (95,92; 4,2)

Anwendung der FP-Methode

Beispiel: Zu schreibender Quellcode

- C++ Zeilen = $95,92 * 53 = \text{rund } 5100$
- C++ Zeilen pro Monat = rund 1200
- C++ Zeilen pro Tag = rund 57
(21 Arbeitstage)

Algorithmische Schätzverfahren – Function-Point-Verfahren

- Jones (1996) gibt Faustregeln zur Aufwandsberechnung mit Function-Points an und eine Abschätzung der Anzahl der Code-Zeilen pro Function Point:
 - Durchlaufzeit [in Monaten] = $FP^{0.4}$
 - Anzahl Mitarbeiter = $FP / 150$
 - Aufwand = Durchlaufzeit * Anzahl Mitarbeiter = $FP^{0.4} * FP / 150$

Sprache	Mittlere Anzahl Codezeilen	Sprache	Mittlere Anzahl Codezeilen
Assembler	320	C++	53
C	128	Ada 95	49
FORTRAN	107	Smalltalk	21
COBOL	197	SQL	12
Pascal	91		

Literatur

- Vorlesungsfolien von Helmut Balzert
 - https://mafiadoc.com/download/1-die-planungsphase-die-function-point-methode_59f5b0471723dd1b3ab56077.html