Inhaltsverzeichnis

1. Einführung 3

1.1 Bestimmung 3

1.2 Einschränkungen 3

2. Berechnung 3

2.1 Use-Case Login 3

2.2 Diagramme 3

3. Abschätzung: Use-Case “Eingeben” 3

3.1 Use-Case Beschreibungen 3

Function Points

# Einführung

Function Points sind eine Größe, die die Komplexität von Software repräsentiert. Sie sind unabhängig von der Programmiersprache und fassen viele Faktoren, wie z.B. Eingänge und Ausgänge in einer Zahl

zusammen. Mit ihrer Hilfe können aus bereits implementierten Projekten bzw. Use Cases Abschätzungen

von Kenngrößen wie Aufwand und Lines of Code (LOC) für noch folgende Projekte bzw. Use Cases

erstellt werden.

## Bestimmung

Zur Bestimmung werden die Anzahl der Internal Logical Files (ILF), External Interface Files (EIF) External Inputs (EI), External Output (EO) und External Inquiries (EI) gewichtet miteinander verrechnet. Die Gewichtung wird über die Komplexität festgelegt

## Einschränkungen

Function Points wurden für Programme mit Datenbankanbindung konzipiert. Programme mit komplexer Kommunikation zwischen mehreren Teilnehmern, aber ohne Datenbank erhalten nur wenige Funktion Points, obwohl sie trotzdem sehr komplex sind. Deswegen haben wir bei unseren Use Cases die Komplexität der Files nicht nach der herkömmlichen Methode berechnet, sondern nach der Anzahl der Teilnehmer. Dabei wurden Files die nur den Server betreffen mit Low eingestuft und welche die Server und einen Client betreffen mit Average. Bei Files die den Server und mehrere Clients betreffen wurde die Komplexität High gewählt. Mit dieser Methode erzielten wir erheblich bessere Ergebnisse.

# Berechnung

## Use-Case Login

Function Points berechnen…

## Diagramme

Implementierungsaufwand...

# Abschätzung: Use-Case “Eingeben”

Dieser Abschnitt beschreibt die spezifischen Anforderungen an die Software.

## Use-Case Beschreibungen

Die zu implementierenden Use-Cases warden im Folgenden beschrieben.