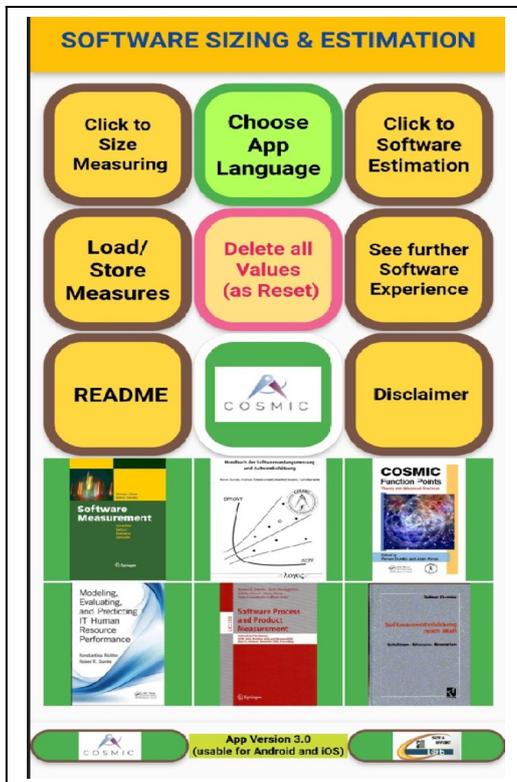


The SoftwareExpert App

by Reiner R. Dumke



The **SoftwareExpert** app is not a comprehensive and large project management system. It helps to get a **quick and concrete overview of the IT project situation** and/or the essential software system features and allow to save them for further comparisons. **SoftwareExpert** helps to accurately measure the software scope. This app serves mainly for the application of the **COSMIC Function Point method** (as **International Standard ISO/IEC 19761**) in a simplified form and can be used via the button "Software Sizing". Up to 20 so-called functional processes and up to 15 associated data groups (as identification of the COSMIC metrics (data movement)) can be defined/acquired. The determination of the **COSMIC Function Points (CFP)** with the four sub metrics as **Entries, Exits, Reads and Writes** then takes place via the COSMIC Sizing Button whereby each functional process receives its CFP and the total CFP (**Total CFP**) is then displayed. This app also enables the application of a shortened COSMIC method as **Early & Quick Method** and the local extension approach of the process CFPs (e. g. by considering the "inner" functional scope) as **Extend Method**.

As a special benefit, this app includes the ability (in the **Estimate button**) of using the estimation formulas and relations known from the international experience for the effort, the productivity, the project duration and much more due to a size/volume measurement (mainly as CFP). Some of these estimates are displayed under the **Project Dashboard** button as a general overview and a **Project Metrics** overview. If other size measurements are given (e. g. as **LOC, IFPUG Function Points, Story Points** etc.), a conversion can be done under **Convert** before and the estimation formulas can be used on this basis.

The informative part of this app is supplemented by an overview of some of the very classic experiences of software engineering (as **Software experiences**). The measurement data can be provided with an identification and stored within the app (and reloaded later).

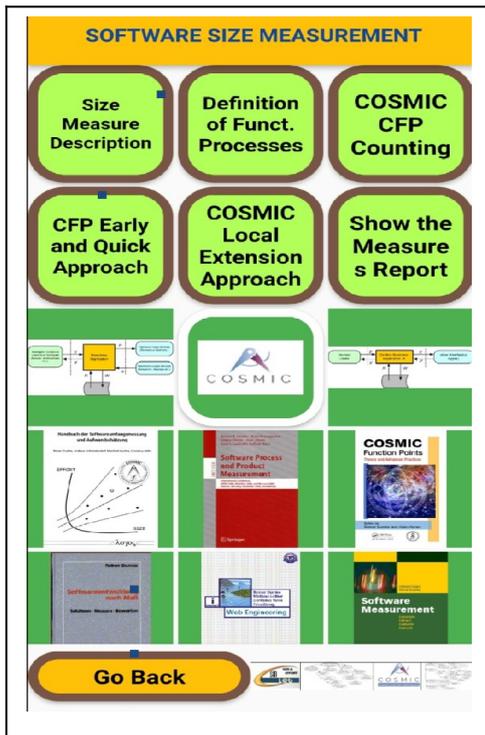
On the other hand, these app pages contain links to the COSMIC community, to SML@b, to our metrics bibliography on our GI website as well as to the risks of Peter Neumann and a SWEBOK classification for software engineering in general.

This app is useful for short and fast IT project controlling in the agile development and as educational support for computer science students and professionals based on the current knowledge in empirical software engineering.

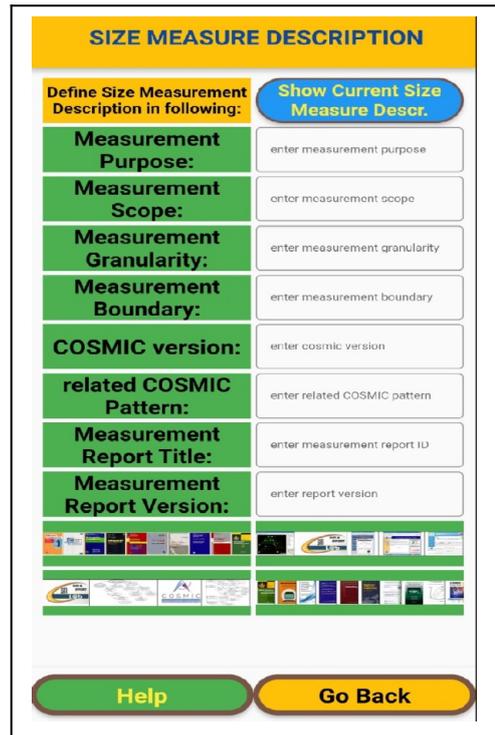
(sorry, **SoftwareExpert** app has a symbolic price)

Software Sizing:

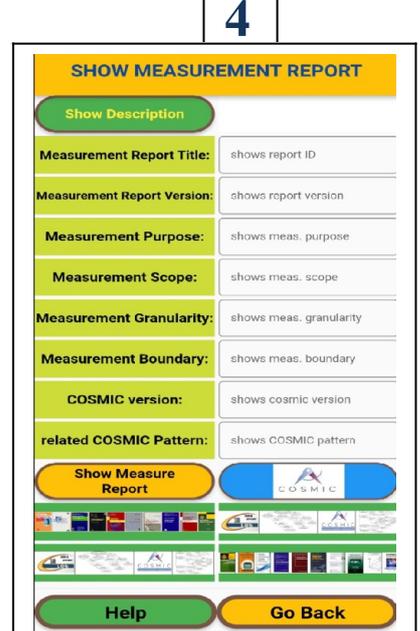
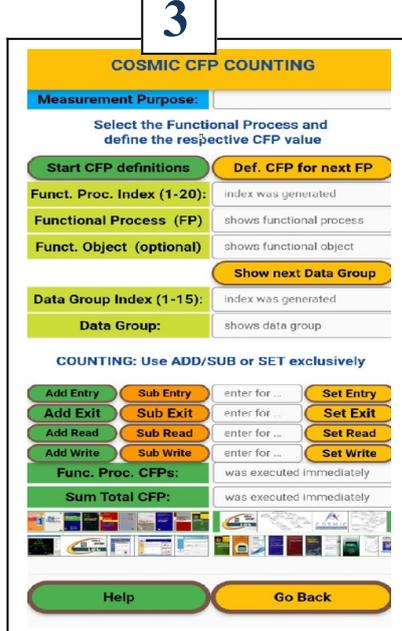
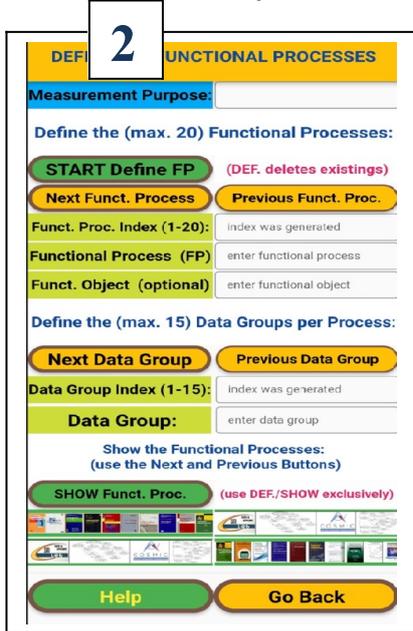
Software sizing starts with the definition of the Size measurement characteristics in the **Size Measure Description (1)**. These inputs are optional but meaningful in the IT context.



1



After these inputs, the **Functional Processes** (limited by 20) are defined as basis of size measurement (2). Each functional process can contain several (limited by 15) data groups in order to identify the data movements.



The **COSMIC CFP Counting** supports the size measurement based on the *Entry, Exit, Read and Write metrics* (3). The **Set buttons** help in the case of large metrics values.

Every size measurement page has a **Help** button. Especially, the **Help** for **Counting** includes an overview about the **COSMIC patterns** in order to support the COSMIC software model identification.

COSMIC LOCAL EXTENSION

Measurement Purpose:	<input type="text"/>
Start of the FP-based CFP extension	Define CFP extension for next FP
Funct. Proc. Index (1-20):	<input type="text" value="Index was generated"/>
Functional Process (FP)	<input type="text" value="shows functional process"/>
func. Proc. CFPs	<input type="text" value="current CFP value"/>
Local Extension per Functional Process:	
Define CFP Value Extension	<input type="text" value="enter FP - CFP extension"/>
Add CFP Extension	<input type="text" value="extended FP related CFP"/>
Subtract from Total CFP	
Sum Total CFP:	<input type="text" value="executed total CFP"/>
	
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> Help Go Back </div>	

A COSMIC size measurement can be *stored* or *loaded* using the **Load/Store button**.

LOAD/STORE MEASUREMENTS

ID List of Stored Size Measurements

see MeasureID list

Load Size Measures from Database	<input type="text" value="enter MeasureID for loading"/>
Delete ID Size Measures in Database	<input type="text" value="enter MeasureID for delete"/>
Delete All Measures in Database(really?)	<input type="text" value="enter 'yes' for all delete"/>
Stored Measurements into Database	<input type="text" value="enter MeasureID for saving"/>






Go Main Menu

SELECT THE APP LANGUAGE

English

Deutsch

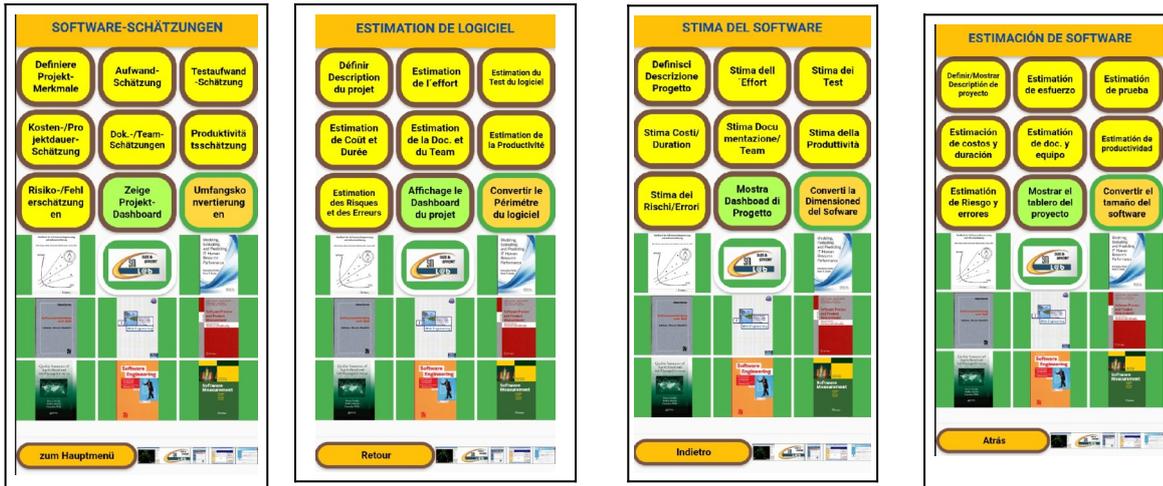
Français

Español

Italiano

Go Main Menu

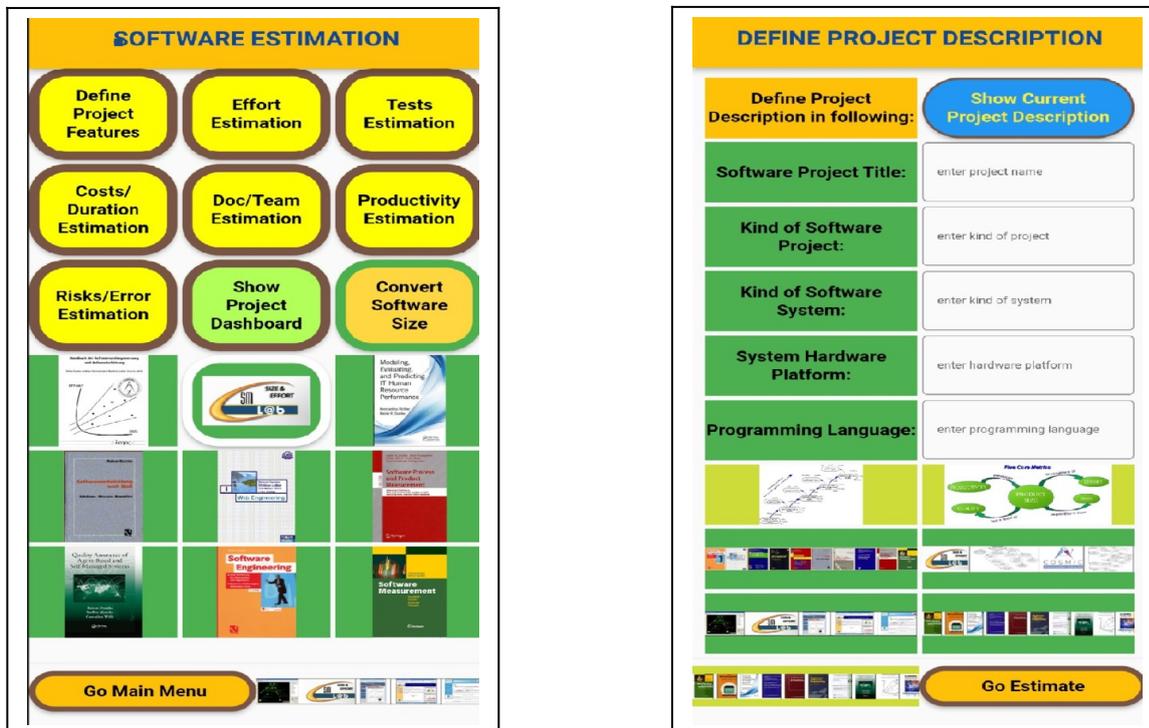
The **SoftwareExpert** app is usable in five languages: **English, German, French, Italy and Spain**



depending on the language setting of your Smartphone/Tablet.

Software Estimation:

The different aspects of software process estimation are essential features in the *SoftwareExpert* app. The Software estimation area can be used independently of COSMIC software sizing. It starts with a **Project description** as basic for the identification of the software process estimations. These inputs are also optional but meaningful for later measurement comparisons.



Then you can continue with the estimations using the **total CFP value** from the software sizing before.

CONVERTING SOFTWARE SIZE

Software Project Title:

See/Use current Size:

Convert the Size Value from one Unit to another one:

Convert from Size Unit: << CFP unit >>

Convert to Size Unit: << CFP unit >>

(use for COSMIC FP: CFP, IFPUG Function Points: IFPUG, Lines of Code: LOC, NESMA FP: NESMA, Story Points: SP, Use Case Points: UCP, Feature Points: FP)

Define the from Size:

Converted Software Size:

In following would be used:

use It!

LOC Relationships between different Programming Languages by Jones:

See/Define current Size LOC:

See current Prog. Language:

Define current Prog. Language: < JAVA language >

Define other Prog. Language: < JAVA language >

Execute other LOC:

Go Estimate

But, you can start with **your own size value**. You can choose between *Lines of Code (LOC)*, *IFPUG Function Points*, *NESMA Function Points*, *Feature Points*, *Use Case Points* and *Story Points* and can transform this value to the *general app unit* as **COSMIC Function Points (CFP)** using the **Convert** features.

Then you can perform the different estimates like effort, test, cost, duration, documentation, team size, productivity, risk and errors.

Note, all estimations are based on the given/transformed/defined/assumed software size. You can choose the existing value (by measurement or converting before) or you define your own size value (in CFP!).

In every part of project aspect estimation, you can select (one or more) between several estimates by **Experts** that you can find in the literature of the empirical software engineering. Press the **Average** button for storing the value in the **Project Dashboard**. The correction of any chosen values can be done by pressing the chosen button again.

In principle you can define your own estimation using your *own scaling factors*. The following estimation can be used with the **SoftwareExpert** app.

Development and test estimations

SOFTWARE EFFORT ESTIMATION

Software Project Title:

Effort Estimation is based on CFP Size:

See/Use current Size:

Define the new Size:

use It!

The Effort Estimation Unit is Personal Month [PM]:

Dev. Effort by ABRAN:

Dev. Effort by SNEED:

Effort by BUNDSCHU:

Dev. Effort by SBS3c:

Development Effort on Average:

Maintenance Effort by HERINGEN:

Migration Effort by SNEED:

Define your own Factor:

Execute your own Effort:

Go Estimate

SOFTWARE TEST ESTIMATION

Software Project Title:

The Test Estimation is based on CFP Size:

See/Use current Size:

Define the new Size:

use It!

The Effort Estimation Unit is Personal Month [PM]:

Test Effort by Sneed

Regression Test (Heringen)

The Number of Test Cases by CAPERS JONES:

Unit Test Cases

Integration

System Test Cases

Perform. Test Cases

Usability Test Cases

Security Test Cases

Acceptance

Chosen Number of Test Cases:

Go Estimate

Costs, duration, documentation and team estimations

COSTS & DURATION ESTIMATION

Software Project Title:

See/Use current Size:

Define the new Size:

use it!

Costs Estimation is based on CFP Size:

Define the Costs per PM in EURO: **use it!**

Cost Estimation (Jones): **[EURO]**

Project Duration Estimation is based on CFP Size:

Duration by BOURQUE: **Proj. Duration:**

Duration by SNEED: **Project Duration on Average:**

Duration by ISBSG (Java):

Duration for Realtime:

Define your own Factor:

Execute your own Duration:

Go Estimate

DOCUMENTATION & TEAM ESTIMATION

Software Project Title:

Doc/Team Estimation is based on CFP Size:

See/Use current Size:

Define the new Size:

use it!

Project Doc. Size:

Program Doc. Size:

User Manual Size:

Chosen Doc. Size by JONES:

Doc. Effort by Lehner:

Development Team Size in Number of Members:

Team Size by JONES:

Team Size by BUNDSCHUH:

Team Size on Average:

Go Estimate

Productivity, risks and error estimations

SOFTWARE PRODUCTIVITY ESTIMATION

Software Project Title:

Productivity Estimation is based on CFP Size:

See/Use current Size:

Define the new Size:

use it!

The Productivity Unit is CFP per Personal Month (PM):

Productivity by ISBSG:

Productivity by PUTNAM:

Productivity on Average:

Productivity by ISBSG-CBSE:

Define your own Factor:

Execute own Productivity:

PRODUCTIVITY COMPARISON by JONES:

Average Case (Agile Development): 7.26 [CFP/PM]

PRODUCTIVITY COMPARISON by PUTNAM/MYERS:

Business System Development:

Worst Case (Waterfall Modell): 2.71 [CFP/PM]

Best Case (Team Software Process): 10.89 [CFP/PM]

Avionics Development: 4.9 [CFP/PM]

Embedded System Development:

Go Estimate

RISKS & ERROR ESTIMATION

Software Project Title:

See/Use current Size:

Define the new Size:

use it!

Risks Estimation is based on CFP Size:

Risk Factor by JONES:

Error Estimation is based on CFP Size:

Error Est. by HALSTEAD:

Error Est. by SNEED:

Error Est. by JONES:

Error Estimation on Average:

Programming Language related Error Estimation by JONES
(for C, C++, Cobol, Java or Assembler)

Choose Prog. Language and press this Button:

Estimated Errors:

Go Estimate

Finally, these estimation values are stored in the so-called **Project Dashboard**. You can consider and accept this dashboard or you can make any corrections in the different estimation parts and define the Project dashboard again. Furthermore, some **Project Metrics** are delivered in a temporary manner and can help to characterize your IT project controlling with further interesting aspects. Both, the size measurement and/or estimation can be stored using the **Load/Store Button**.

SHOW PROJECT DASHBOARD

Software Project Title:

PROJECT SIZES

Product Size in CFP:

Product Size in LOC:

Team Size in members:

Documentation Size in Pages:

PROJECT EFFORT

Development Effort in PM:

Testing Effort in PM:

Documentation Effort in PM:

PROJECT EFFICIENCY

Project Costs in Euro:

Project Duration in Month:

Productivity in CFP/PM:

Productivity in LOC/PM:

PROJECT QUALITY

Chosen Number of Test Cases:

Risk Factor in percent:

Estimated Number of Errors:

Show further Project Metrics
Go Estimate

SHOW PROJECT METRICS

Software Project Title:

Product Size in CFP:

Product Size in LOC:

FURTHER EXECUTED PROJECT METRICS:

Percentage of Documentation (chosen vs. complete):

Percentage of chosen Test Cases (chosen vs. all test aspects):

Errors per team member:

Percentage of Test Effort vs. Development Effort:

Percentage of Documentation Effort vs. Development Effort:

Costs per COSMIC Function Points (in Euro):

Costs per Project Months (in Euro):

Estimated Maintenance Effort (based on the Dev. Effort) in [PM]:

Total Formal Verification Effort (by Klein et al.) in [PM]:

Go Back

Software Experiences:

The part of software experiences summarizes the well known empirical knowledge in software engineering for typical software engineering areas.

The Web links to our Metrics bibliography, the Neumann risks overview and the software engineering description in the SWEBOK could be helpful for understanding and motivating of IT project aspects and problems.

SOFTWARE EXPERIENCES

Software Products Experience

Software Processes Experience

Software Teams Experience

Software Platforms Experience

Experience of AOSE Software

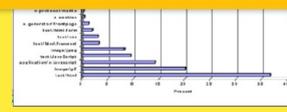
Experience of OOSE Software

Experience of CBSE Software

Experience of SOSE Software

Go Main Menu

SOFTWARE PLATFORMS EXPERIENCE



Consider the USABILITY frequently:
Cumulative Number of Hits by Hour of Day



THE RISKS DIGEST

MEASUREMENT BIBLIOGRAPHY

SWEBOK

Go Back

This **SoftwareExpert** app is based on our COSMIC books:

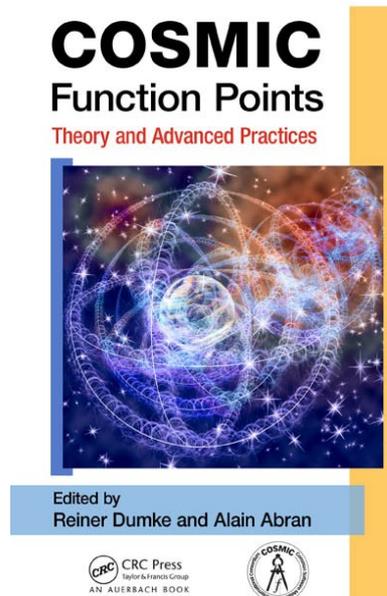
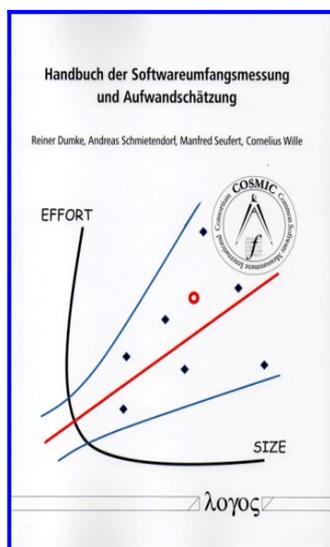
Dumke, R., Abran, A.:

COSMIC Function Points – Theory and Advanced Practices
CRC Publ., Boca Raton, 2011 (334 p.) ISBN 978-1-4398-4486-1

and

Dumke, R., Schmietendorf, A., Seufert, M., Wille, C.:

Handbuch der Softwareumfangsmessung und Aufwandschätzung
Logos Verlag, Berlin, 2014 (570 p.), ISBN 978-3-8325-3784-5



Further/our resources you can find in our measurement bibliography at

<https://fg-metriken.gi.de/publikationen/bibliographie>

The **SoftwareExpert** app can be helpful in computer science education in order to better understand the complexities and relationships in software project management in practices.