FiSMA 1.1

A Functional Size Measurement Method with continuous scale:

Basic principles and practical examples
FiSMA Functional Size Measurement Method Version 1.1 (referred to as FiSMA 1.1) is a general, parameterised functional size measurement method for all types of software.
Introduction

FiSMA 1.1 is a mature and rigorous Functional Size Measurement (FSM) method. Since its first introduction as “Laturi method” over 15 years ago, FiSMA 1.1 has evolved based on scientific research and user company responses. When practitioners familiar with other popular FSM methods first encounter FiSMA 1.1 they may notice that this method measures more concrete manifestations of Functional User Requirements (FUR) and takes a different approach than any of the first generation FSM methods. The reader is reminded that FiSMA 1.1 measurement data have been independently validated by no less than five internationally respected researchers and size measurement results have been found to correlate extremely well with productivity, estimate accuracy etc.

This presentation is intended to:

- Introduce the basic principles and all measurement procedures and formulas of FiSMA 1.1.
- Provide real life examples of each Base Functional Component (BFC) type in FiSMA 1.1.
- Include sample FFP counts for the example BFC:s.
- Increase knowledge transfer and understanding of FiSMA 1.1.

The reader can find more details about the use and the related research of FiSMA 1.1 method in another document: FiSMA_PAS_Explanatory report, 17.1.2007.
Main features of FiSMA 1.1

- Based purely on Functional User Requirements
- Service oriented
- Requires identification of all different services provided by the piece of software
Purpose and Scope

- Functional size is an essential measure for all software comparisons.
- Main uses are estimating and productivity analysis.
- Proven to be useful in project planning, tracking, control, and contracting.
FiSMA 1.1 terms and definitions

- **User** is any person or thing that communicates or interacts with the software at any time
- **End-user** is any person that communicates or interacts with the software at any time
- **Functional User Requirements (FUR’s)** are specified as functions or functional services
- **Base Functional Component (BFC)** is an elementary unit of the FUR, i.e. a functional service
- **BFC classes** (7) and **BFC types** (28)
- + all ISO/IEC 14143-1 definitions
How people see functional services of software
Seven FiSMA 1.1 BFC classes

- Interactive end-user navigation and query services (q)
- Interactive end-user input services (i)
- Non-interactive end-user output services (o)
- Interface services to other application (t)
- Interface services from other applications (f)
- Data storage services (d)
- Algorithmic and manipulation services (a)
28 FiSMA 1.1 BFC types

Functional services
of FiSMA FSM

- Interface services from other applications
  - Messages in
  - Batch records in
  - Signals in

- Interface services to other applications
  - Messages out
  - Batch records out
  - Signals out

- Algorithmic and data manipulation services
  - Security routines
  - Calculations routines
  - Formatting routines
  - Delete routines
  - Other routines

- Data storage services
  - Entities/classes
  - Other records

- Interactive end-user navigation and query services
- Interactive end-user input services
- Non-interactive end-user output services

- Icons
- Log-in, log-out
- Menus
- Selection lists
- Data inquiries
- Generation dial
- Browsing lists
- 3-functional
- 2-functional
- 1-functional
- Forms
- Reports
- E-mails or sms
- Monitor screens
Counting rule for navigation and query services (q1-q7)

- **Input:** Number of data elements displayed (n) AND number of reading references needed to provide them (r).
- **Process:** \(0.2 + \frac{n}{7} + \frac{r}{2}\)
- **Output:** Functional size of the navigation or query BFC
Icon (q1)

- an object that initiates a service or the piece of software

FUR: "I want to find my new application and get it started."

Examples of different icons:

Adobe Reader 6.0.Ink
A BFC with 2 data elements and 1 reading reference ("system parameters") => 1,0 FFP

Skype.Ink
A BFC with 2 data elements and 1 reading reference ("system parameters") => 1,0 FFP

A BFC with 2 data elements and 1 reading reference ("system parameters") => 1,0 FFP
Log-in and log-out dialogs (q2)

• parts of interactive end-user navigation and query services that control users' access and prevent illegal use

FUR: "I want to control and authenticate the access to my application."
Menus (q3)

• common parts of interactive end-user navigation and query services that are made for selecting the next operation

FUR: “I want to be able to select which task will be activated or to which part of the system to navigate next.”
Selection lists (q4)

- parts of interactive end-user navigation and query services that show a list of acceptable parameter values to the end-user

FUR: “I want to find all available values of the parameter.”

A BFC with 3 data elements and 1 reading reference (=system parameters) => 1,1 FFP

A BFC with 2 data elements and 1 reading reference (=system parameters) => 1,0 FFP

A BFC with 2 data elements and 1 reading reference (=system parameters) => 1,0 FFP
Data inquiries (q5)

- parts of interactive end-user navigation and query services that show the content of data store(s) to the end-user

FUR: "I want to see detailed information about the object of my interest."

A BFC with 40 data elements and 4 reading references => 7,9FFP
Generation dialogs (q6)

- parts of interactive end-user navigation and query services that help the end-user to initiate the production of a report or a manipulation routine

FUR: "I want to have control to initiate a particular sub-process."

A BFC with 21 data elements and 2 reading references => 4,2 FFP

A BFC with 13 data elements and 2 reading references => 3,1 FFP
Browsing lists (q7)

- are parts of interactive end-user navigation and query services. They show a list of similar data elements, typically the most important details to help filter the entities for further operations.

FUR: "I want to see a high-level list of the data on which I can further filter results or check the details."

A BFC with 18 data elements and 3 reading reference => 4,3 FFP
Summary of navigation and query services (q1-q7)

• Total size of the presented 14 sample navigation and query services from 7 previous slides:

\[ S_q = 3 \times 1,0 + 2,3 + 1,6 + 1,4 + 1,3 + 1,1 + 2 \times 1,0 + 7,9 + 4,2 + 3,1 + 4,3 = 32,2 \text{ FFP} \]
Counting rule for interactive input services (i1-i3)

- **Input**: Functionality multiplier ($m$), number of data elements displayed ($n$) AND number of writing references ($w$) AND number of other reading references needed to provide them ($r$). The value of ($m$) is 1, 2 or 3, depending on how many of functions create, update and delete the service incorporates.

- **Process**: $m \times (0.2 + n/5 + w/1.5 + r/2)$

- **Output**: Functional size of the interactive input BFC
1-functional input dialogs (i1)

- support only one of the three maintenance types: create, update, or delete

FUR: "I want to be able to create (or update or delete) data to be stored in the application."

A BFC with 10 data elements, 1 writing reference and 1 additional reading reference (=system parameters) => 3.4 FFP

A BFC with 8 data elements, 1 writing reference and 1 additional reading reference (=system parameters) => 3.0 FFP
2-functional input dialogs (i2)

- support two of the three maintenance types create, update or delete

FUR: “I want to create and update data stored in the application”

A BFC with 19 data elements, 1 writing reference and 1 additional reading reference (=system parameters) => 10.3 FFP
3-functional input dialogs (i3)

- support all three maintenance types create, update or delete

FUR: "I want to create and update and delete data stored in the application"

A BFC with 26 data elements, 1 writing reference and 3 additional reading references => 22.7 FFP
Summary of interactive input services (i1-i3)

- Total size of the 4 presented sample input services from 3 previous slides:

\[ S_i = 3.4 + 3.0 + 10.3 + 22.7 = 39.4 \text{ FFP} \]
Counting rule for non-interactive output services (o1-o4)

- **Input:** Number of data elements on the output \((n)\) AND number of reading references needed to provide them \((r)\).
- **Process:** \(1 + \frac{n}{5} + \frac{r}{2}\)
- **Output:** Functional size of the non-interactive output BFC
Output forms (o1)

- are services resulting in printed or displayed documents, which always present the same layout (e.g. a receipt)

FUR: "I want to be able to see data from the application in a standard report"
Reports (o2)

- are services resulting in printed or displayed documents, whose layout may vary within the specified framework according to the presented data (e.g. product list or sales report).

FUR: "I want to see the data from the application displayed in a standard report for which the data drives the format."
E-mails and text messages (o3)

- are services resulting in electronically transmitted output documents, which have a standardised structure. The structure often contains title fields, data fields and optional attachments.

A BFC with 5 data elements, and 2 reading references => 3.0 FFP

FUR: ”I want to send this information directly from my application to the users mobile phone or e-mail box”

A BFC with 6 data elements, and 2 reading references => 3.2 FFP
Monitor screen output (o4)

- service involves continuously displayed documents, which are updated regularly in consequence of data changes (e.g. measurement display of a process).

FUR: "I want to see up to the minute values of particular data in the application"
Summary of non-interactive output services (o1-o4)

- Total size of the 7 presented sample output services from 4 previous slides:

$$S_o = 6.8 + 8.7 + 10.9 + 5.5 + 3.0 + 3.2 + 6.1 = 43.2 \text{ FFP}$$
Counting rule for out going interface services (t1-t3)

- **Input:** Number of data elements on the interface record (n) AND number of reading references needed to provide them (r).
- **Process:** 0.5 + n/7 + r/2
- **Output:** Functional size of the out going interface BFC
Messages to other applications (t1)

• are services where data groups are sent on-line, usually in real-time, to any other application.

FUR: "I want my application to send data real-time to other applications"
Batch records to other applications (t2)

A BFC with 5 data elements, and 2 reading references => 2,2 FFP

FUR: "I want my application to send data to other applications ... e.g. every night"

- are services where data groups are written to a temporary file for transfer to another application.
Signals to devices or other applications (t3)

- are services where data strings or single pieces of information are sent to any other application or a device (e.g., a LED).

FUR: “I want my application to change the status of a device”

A BFC with 1 data elements, and 1 reading reference (=system parameters) => 1,1 FFP

A BFC with 2 data elements, and 1 reading reference (=system parameters) => 1,3 FFP
Summary of outgoing interface services (t1-t3)

- Total size of the 4 presented sample outgoing interface services from 3 previous slides:

\[ S_t = 3.5 + 2.2 + 1.1 + 1.3 = 8.1 \text{ FFP} \]
Counting rule for in coming interface services (f1-f3)

• **Input:** Number of data elements on the interface record (n) AND number of writing references (w) AND number of other reading references needed to provide the service (r).

• **Process:** $0,2 + \frac{n}{5} + \frac{w}{1,5} + \frac{r}{2}$

• **Output:** Functional size of the in coming interface BFC
Messages from other applications (f1)

- are services where data groups are received on-line, usually in real-time from any other application.

FUR: "I want my application to receive real-time data from other applications"

A BFC with 14 data elements, and 1 writing reference and 1 reading reference (=system parameters) => 4,2 FFP
Batch records from other applications (f2)

- are services where data groups are received off-line from any other application.

FUR: ”I want my application to receive data from other applications every night”

A BFC with 5 data elements, 1 writing reference and 2 additional reading references ⇒ 2.9 FFP
Signals from devices or other applications (f3)

- are services where strings or single pieces of information are received from any other application or device (e.g., a sensor).

FUR: "I want my application to receive signals from for example a card reader"
Summary of in coming interface services (f1-f3)

- Total size of the 3 presented sample in coming interface services from 3 previous slides:

\[ S_f = 4,2 + 2,9 + 2,0 = 9,1 \text{ FFP} \]
Counting rule for data storage services (d1-d2)

- **Input**: Number of data elements i.e. attributes of an entity (n).
- **Process**: $1.5 + \frac{n}{4}$
- **Output**: Functional size of the data storage BFC
Entities or classes (d1)

- are data storage services resulting in persistent logical data representing fundamental things of relevance to the user, and about which persistent information is stored.

FUR: "I want my application to store data about ..."
Other record types (d2)

- are the other type of data storage services and result in the persistent logical data besides that which is counted as entities or classes.

FUR: "I want my application to store data about ... game statistics"
Summary of data storage services (d1-d2)

- Total size of the 5 presented sample data storage services from 2 previous slides:

$$S_f = 2.1 + 1.9 + 2.5 + 2.3 + 2.7 = 11.5 \text{ FFP}$$
Counting rule for algorithmic and manipulation services (a1-a6)

- **Input**: Number of data elements i.e. different variables (n) AND maximum number of operations needed (r).
- **Process**: $0.1 + n/5 + r/3$
- **Output**: Functional size of the manipulation BFC
Security routines (a1)

- are manipulating services providing security features such as encryption, decryption, advanced authorization, etc.

FUR: "I want my application to check the authentication of a user using a certain type of algorithm ..."

A BFC with 5 data elements, and 4 operations => 2.4 FFP
Calculation routines (a2)

- are manipulating services providing arithmetic or logical counting services.

Count the **Functional-Size** using **Number-of-data-elements** on the interface record (n) AND **Number-of-writing-references** (w) AND **Number-of-reading-references** (r) using the formula: \( A + \frac{n}{B} + \frac{w}{C} + \frac{r}{D} \) where \( A, B, C \) and \( D \) are constraints given by the FiSMA 1.1 method.

A BFC with 8 data elements, and 7 operations => 4.0 FFP

FUR: "I want my application to calculate the size of a piece of software using a certain type of algorithm ..."
Simulation routines (a3)

- are manipulating services providing simulative calculating services.

A BFC with 12 data elements, and 14 operations => 7,2 FFP

Loan Amortisation Schedule

<table>
<thead>
<tr>
<th>Principal</th>
<th>calculate Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Payments</td>
<td>calculate Number</td>
</tr>
<tr>
<td>Interest Rate (%)</td>
<td>calculate Interest</td>
</tr>
<tr>
<td>Payment</td>
<td>calculate Payment</td>
</tr>
</tbody>
</table>

reset  Payment Schedule

FUR: “I want my application to count the missing value if I know any other three parameters of ... e.g. loan simulation algorithm ...”
Formatting routines (a4)

- are manipulating services providing special format conversion services (i.e. beyond typical, simple editing). An example of a formatting routine could be changing table rows into graphics.

FUR: ”I want my application to draw a colourful surface from the values of the table ... using a certain type of algorithm ...”

A BFC with 11 data elements, and 8 operations => 5,0 FFP
Database cleaning routines (a5)

- are manipulating services supporting data storage maintenance, such as removing unnecessary records and combining or cumulating data elements based on user-defined rules.

”If start-date of record is older than age-limit, then add a-number to history-a-number, add b-number to history-b-number and delete record.”

FUR: ”I want my application to maintain the data contents automatically following the next rules ...”

A BFC with 6 data elements, and 4 operations => 2.6 FFP
Other manipulation routines (a6)

- include all independent user-defined data manipulation services, which are not counted as any other algorithmic and manipulation BFC type functions.

FUR: "I want my application to do something very domain specific manipulation using certain type of algorithm ..."

A BFC with 11 data elements, and 1 reading reference (= system parameters) => 3.4 FFP
Summary of algorithmic and manipulation services (a1-a6)

- Total size of the 6 presented sample algorithmic services from 6 previous slides:

\[ S_a = 2.4 + 4.0 + 7.2 + 5.0 + 2.6 + 3.4 = 24.2 \text{ FFP} \]
FiSMA 1.1 measurement process

1. How many each type of BFC's do you have in your piece of software?
2. What are they? Identify them all.
3. What are they like? Give the numbers of details for each BFC.
Size of a piece of software

- \( S = S_q + S_i + S_o + S_f + S_t + S_d + S_a \)
- Measurement unit = 1 FFP (or fp if the method is obvious or reported separately)
- Reporting the measurement results at detailed level is highly recommended
- The functional size of a multi-component software is the sum of the functional sizes of the components
Summary of FiSMA 1.1 size measurement

• Total size of all presented sample functional services from all previous slides:

\[ S = 32.2 + 39.4 + 43.2 + 8.1 + 9.1 + 11.5 + 24.6 = 168.1 \text{ FFP} \]