

Thriving on Uncertainty

-

A Method for Functional Sizing Based on Early Requirements

Robyn Lawrie

robyn.lawrie@charismatek.com

CHARISMATEK Software Metrics

175 Dorcas Street

South Melbourne, Australia,

www.charismatek.com



Functional Sizing Early in the Life-Cycle

Why Do We Need It?

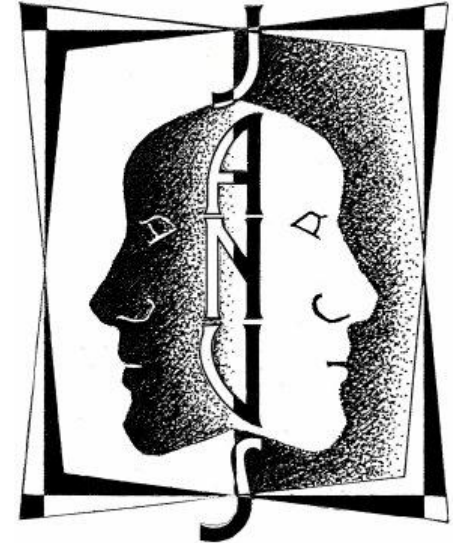
- Early life-cycle functional sizing is performed for a number of **important business** reasons:
 - As a basis for **estimation** of project effort, cost and schedule
 - To assess project **risk** associated with the uncertainty of project requirements
 - To establish the basis for a **cost – benefit** statement which can be monitored throughout the project
 - To **clarify** and **communicate** understood project functional requirements.
 - To **negotiate** and **agree** the functional baseline so that scope change can be monitored throughout the project
 - To explore different delivery strategies – **what-if?**



Functional Sizing Early in the Life-Cycle

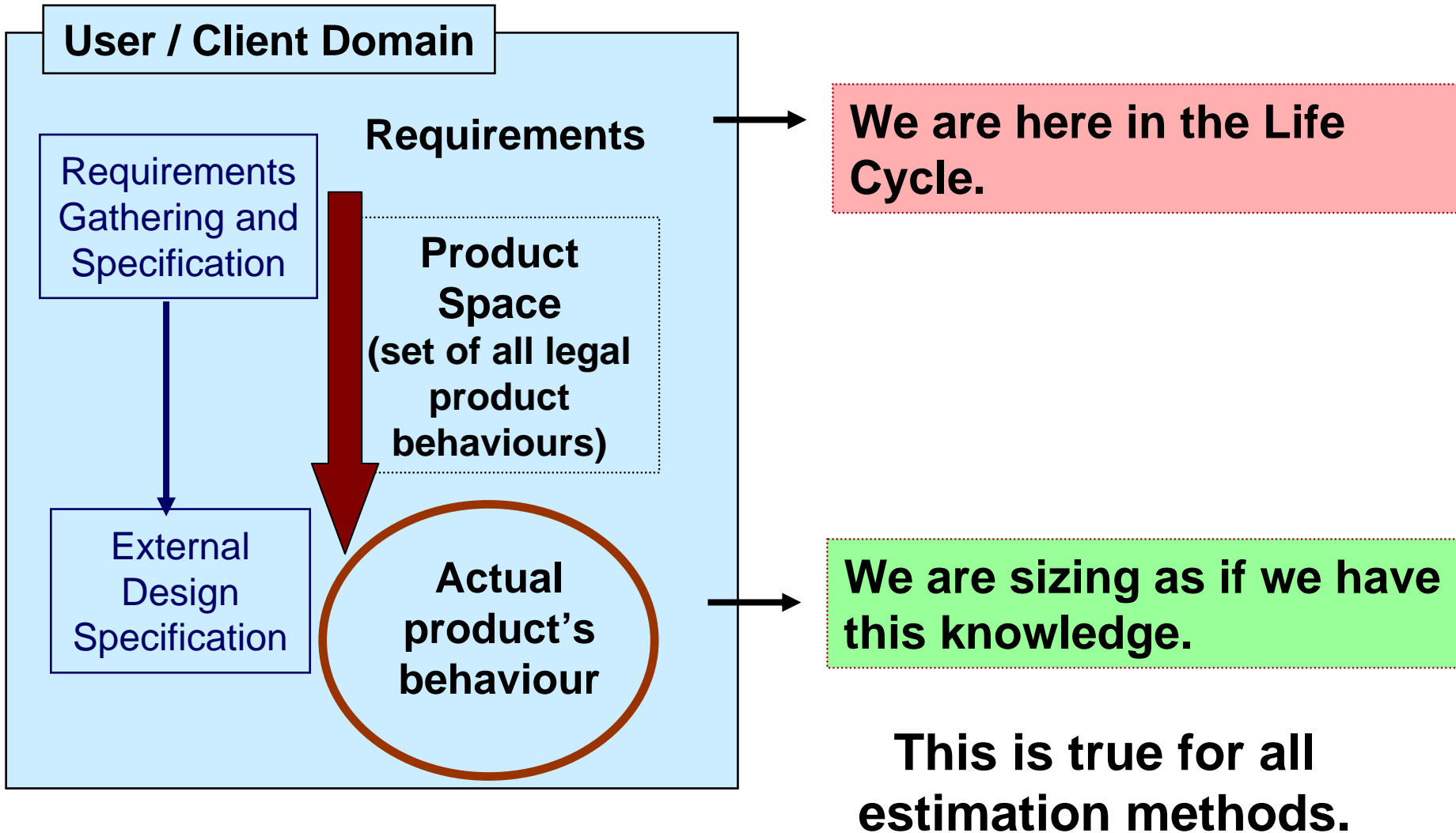
Why Is it Different?

- Most sizing activity across the world is done **retrospectively**, to underpin outsourcing contracts or to benchmark productivity or quality. At this stage the **logical software solution** is fully **known**,
- Even experienced counters may have **little experience** in sizing from **early life-cycle high level requirements**, **before the logical software solution** can be **known**.
- Early life-cycle functional sizing must deal with a **range of problems and issues** arising out of the **uncertainty** of the project knowledge and requirements specifications, typically available early in the project.



Functional Sizing Early in the Life-Cycle

Sizing with Partial Knowledge





Functional Sizing Early in the Life-Cycle

The Nature of the Requirements

- **Definition:**
 - A **high level** expression of **client needs** which will be satisfied by the software.
 - Often assembled by the client **prior** to any developer involvement.
 - May be a structured document but often a **collation** of outcomes from workshops, focus groups, etc.
 - Belong to the **client's knowledge domain** and are written using **client terminology**.

	4.3 Application Processes		
	4.3.1 Lodgement (applies to all Application Transfer types except for Transfer of Land)		
610	New Application		
611	> The system must have a Payment Received flag	Highly Desirable	
612	> The system must generate a unique Application number	Critical	
613	> The system must allow the user to select an existing Applicant or a new Applicant	Critical	
614	> Existing Customer (Buyer/Seller)		
615	* The system must display all Applicant Water Register entries	Critical	
616	* The system must allow a user to select a customer based on a customer number	Critical	
617	* The system must allow a user to select a customer based on a customer name	Critical	
618	* The system must populate all Applicant fields with Applicant data	Critical	



Functional Sizing Early in the Life-Cycle

Accepting The Challenge of Uncertainty

Typical Characteristics of Early Life-Cycle Requirements:

- A mixture of:
 - ‘**Functional**’ requirements – **of interest** when sizing
 - ‘**Non-Functional**’ requirements – **not of interest** when sizing.
- A tendency to ‘**unevenness**’ - ranging from:
 - **Very precise** (prescribing the solution) to
 - **Very vague** (inviting a solution).
- Often quite **unstructured** and **disorganised**.
- To size **successfully** early in the life cycle, you have to accept the **challenge of uncertainty**.



Functional Sizing Early in the Life-Cycle

The Purpose of the Sizing

- To deliver a Functional Sizing which is:
 - **Open**
 - **Fully defensible** (Cross-referenced to supporting information source)
 - With **Indicators of Uncertainty / Confidence** of Sizing

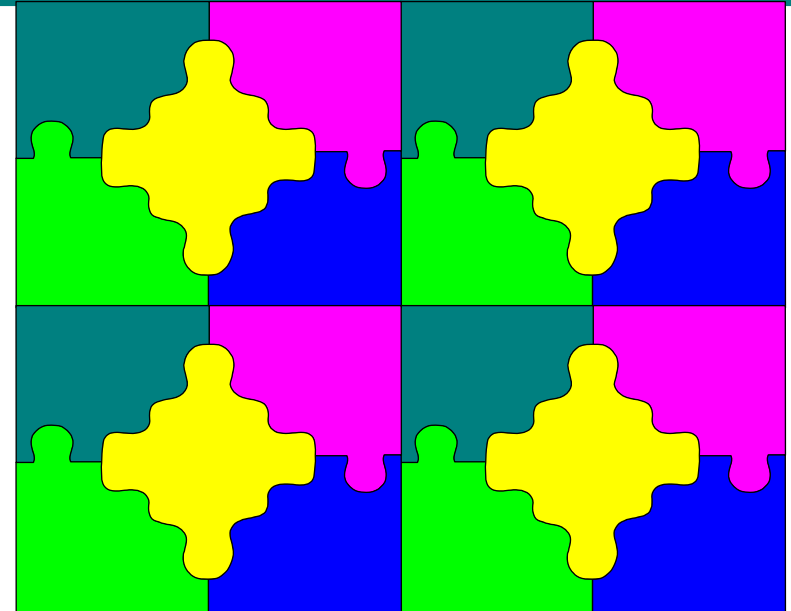
- Business decisions based on this sizing must take into account the “**uncertainty**” so that there can be **overt and conscious** acceptance of the associated **risk**.
- The aim is to understand what is **known** and what **isn't known, what is likely and what is possible**.



Functional Sizing Early in the Life-Cycle

Setting Realistic Expectations

- A software application is, in essence, a defined **set of functions**.
- If we **fail** to identify any of the software's functions, we **fail** to fully and properly determine the software's size.



- In functional sizing terms, in order for a software project to be **properly** sized, **every** unique Logical Transaction and Data Group which contributes its functionality to the software must be **identified and assessed**.
- **Realistically**, at this point in the life cycle, we are attempting to identify the **approximate number** of functions, classified into **Function Type** with **default Complexity Ratings**.



A Method for Functional Sizing from High Level Requirements



1. Getting Started

Prepare for the Count

- Read the requirements **right through** before you start any sizing activity to get a general **understanding** of the project and application.
- Put yourself in the position of the Systems Analyst. You may need to “read between the lines”.
- Identify and markup* where the **functional requirements**, including the **data definitions**, are to be found within the broader requirements.
- Establish stakeholders and their priorities.
- Record the count **purpose** and **scope**
- Establish the application boundaries
 - there may be more than one count



1. Getting Started

Prepare for the Count

- **Mark** any Requirement which will result in Functionality

Functional



"Need to be able to edit and save drafts of content."
'Awaiting approval state in the database must alert the approver using email."
"A download capability for some content types."
"A mechanism is required that allows a content item to be automatically deleted on or after its expiry date."
"The following Entities must be created as part of the Water Register:...."

Non-Functional



"The fonts should be large enough and sentences of a reasonably short length."
"Logos, button, lines should not exceed 2K. Larger graphics and pictures should average 10K, and not exceed the absolute limit of 20K."
"System upgrades must be seamless to the end-user."



1. Getting Started

Prepare for the Count

- Some “functional” requirements will prescribe the solution while others will describe the issue to be addressed.

Requirement	Prescriptive
“Need to be able to edit and save drafts of letter templates.”	No
“Need to be able to undo all critical core functions”	No
“Provide a process to set and manage user access to functions.”	No
“... reporting includes, but is not limited to , the following reports....	No
Report Expired Licences	Yes
Report Student Enrolments in Equine Studies 1”	Yes
“ A mechanism is required that allows a content item to be automatically deleted on or after its expiry date. ”	Yes



1. Getting Started

Prepare for the Count

- **Classify each “Functional” Requirement → Confidence in Sizing**

Type of Requirement	Sizing Confidence Indicator for Transaction / Data Functions
Requirement prescribes Logical Solution for Need	Prescribed –There is a reasonably high degree of confidence that this functionality is in the actual logical solution.
Requirement states Need and invites Solution	Indicative - There is a requirement and the solution is not precisely known at this time. Functionality is included as an indicative solution
Requirement is not explicit but is implied by other Requirements	Implied -There is no stated requirement for this functionality but its existence is implied to support other requirements.



2. Size Prescribed Functionality Primary Business Data

- Identify the “**primary business**” Logical Data Groups or Entities
 - These Data Groups form the Application’s Data Footprint.
 - This is what the application is about.
 - The Data Footprint typically changes very little during the lifetime of the Application.
 - The Data Footprint is key in identifying the boundary of the Application.
 - Each Data Group in this category should be able to be named.
- For each Data group, identify the approximate number of sub-groups (RETs) within each Data Group. Rate the Complexity as Low, Average or High.
- The list of “primary business” Data Groups should be able to be close to complete.



2. Size Prescribed Functionality Non-Primary Business Data

- Identify any “**non-primary business**” Logical Data Groups or Entities specifically identified. These Data Groups provide a supporting role in the Application.
 - **Reference Logical Data**
 - These Data Groups enable the activities against the primary business data or against other Reference Data Groups.
 - **Application and Operation Support Logical Data**
 - These Data Groups support the operation of the application.
- The easiest way to size these is to simply assign **Low Complexity**. Industry Data shows this is a fair approach.
- Expect that the list of Reference and Support Data Groups is incomplete at this stage.



598

The system must hold and enable maintenance of a matrix of Trading Zone Rules

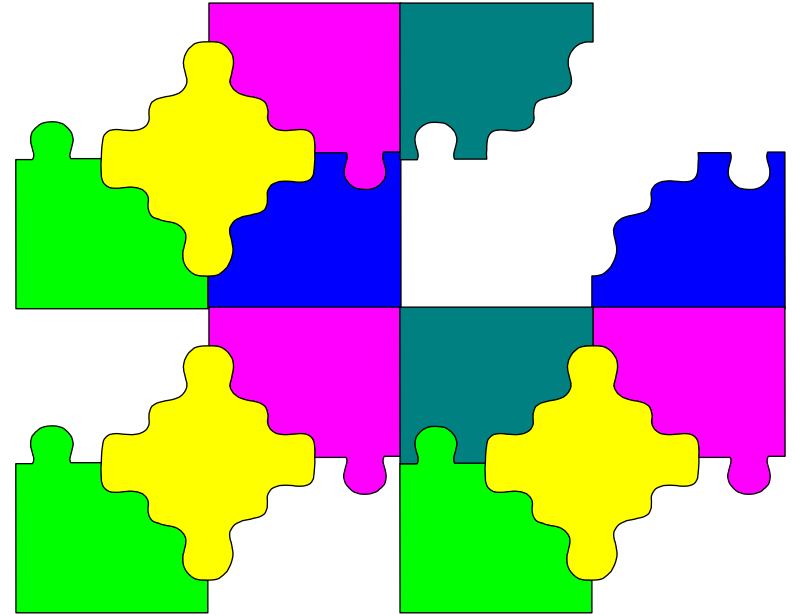
2. Size Prescribed Functionality Logical Transactions

- Record the Logical Transactions where the requirements **prescribe the solution.**
- Map each of these against the Logical Data Groups with which it is mainly associated. Do this so that you can check completeness of coverage later.
- Assign Function Type and give each Logical Transaction ‘**average**’ complexity for the Function Type.



2. Size Prescribed Functionality

- The **prescribed** functionality has now been sized.
- The project size is **at least** what we have counted so far. The Count-To-Date reflects what we know.



- But now comes the **first challenge** ...
“**Filling in the gaps**” for the remaining requirements.



3. Size Indicative Functionality

Some Suggested Methods

Indicative Functionality

There is a requirement but the Logical Solution is not precisely known at this time.

Functionality is included as an **indicative** or **possible** solution

- **Decompose** common usage high-level verbs into “typical breakdown” – e.g. Maintain, Manage
- **Borrow** functionality sets (blueprints) from other applications.
 - If your business requires you to provide fixed price quotations, consider building a library of “blueprints’ for common functionality:
 - Security, Accounting e.g. Payments, Client Management, Standard Letters, Document Management, Website
- Look at functionality sets in **existing ‘legacy’ application.**
- Look at **Interfaces to Other Applications**
- Get **Indicative Number-of-Instances** from the Client.





3. Size Indicative Functionality Decomposition of High-Level Verbs

Simple **Decomposition** of Maintain

- For the Logical Data Group data group to be maintained, create a set of Logical Transactions for a basic life-cycle.
 - For each Logical Data Group:
 - Assume 1 Create, 1 Delete, 1 Browse List (at least)
 - For each sub-group (RET):
 - Assume 1 Modify, 1 Inquiry
- Use your **common sense** and vary this if you think **it is not appropriate in the context**. Remember, Function Point Analysts are hired for their brains! This is a suggestion, not a rule.

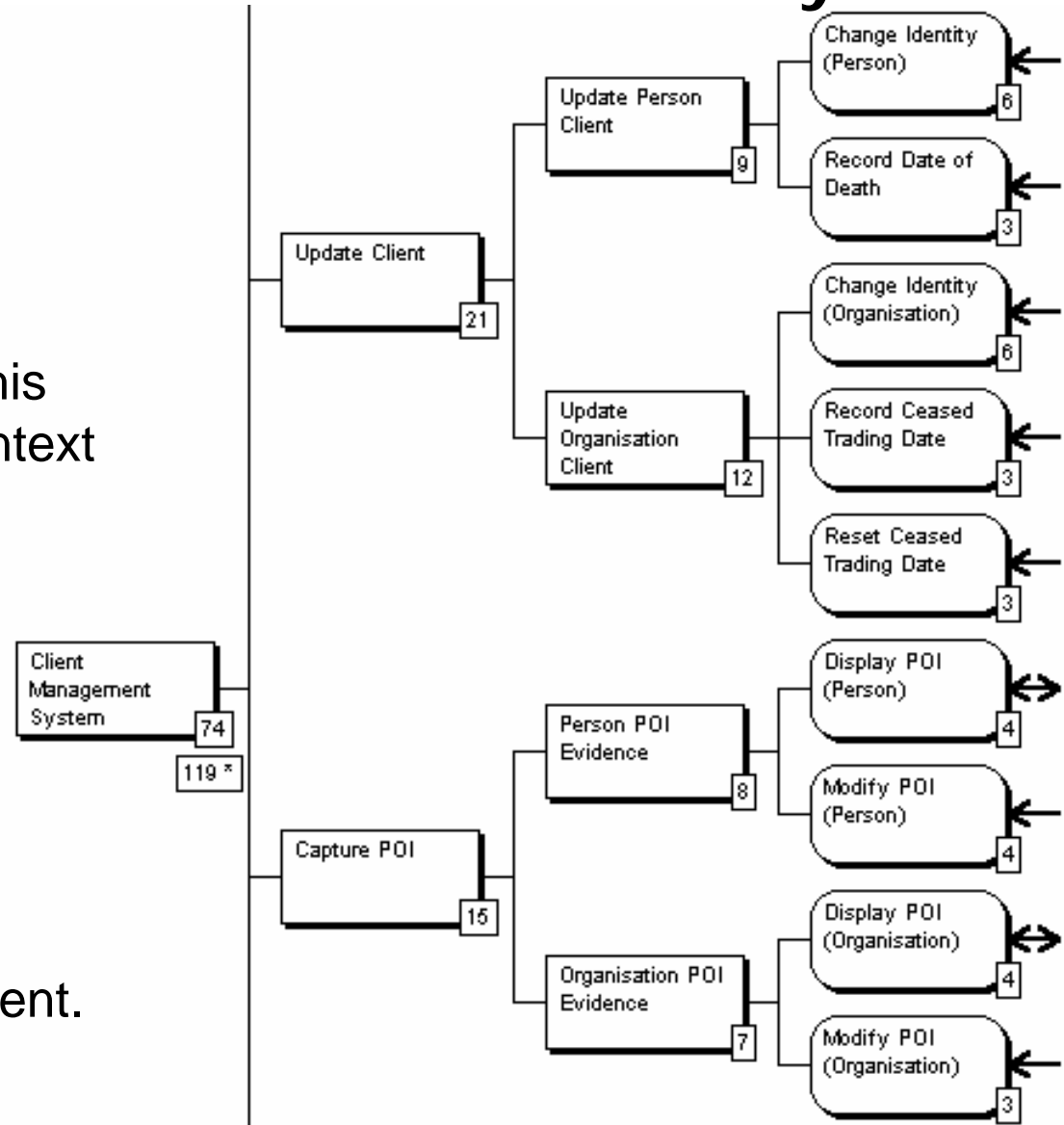


3. Size Indicative Functionality

Example of "Borrowed" Functionality

Requirement: To be able to maintain Client Details. Clients can be People or Organisations.

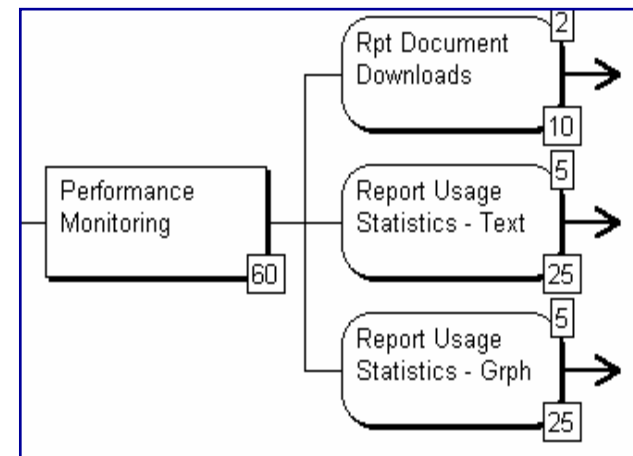
- What sort of solution does this Requirement seek in the context of this application?
- The Logical Solution was **copied** from another application with perceived similar requirements.
- Result is 74 Function Points **allowed** for client management.



3. Size Indicative Functionality

Example of "Borrowed" Functionality

- Requirement: 'To be able to monitor web-site usage'
- What sort of solution does this Requirement seek in the context of this application?
- The Logical Solution was **copied** from another application with perceived similar requirements.
- Result is 60 Function Points **allowed** for Performance Monitoring.
- Note the use of the Multiplier to show indicative number of instances.

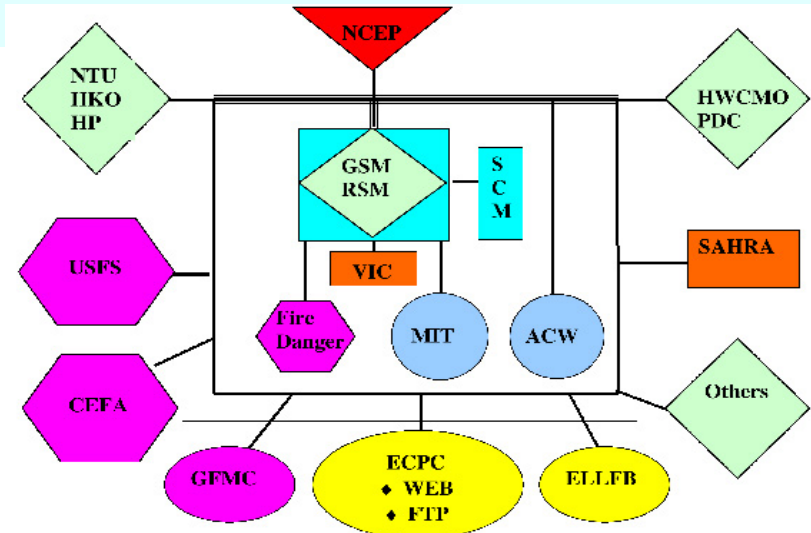


The assumptions may be wrong but they are completely **transparent** and can be **adjusted** if **better information** becomes available

3. Size Indicative Functionality Beware - External Interfaces

- What are the **interfaces** to other Applications?
- Example: The requirement to support interfaces to other systems had been described in a single line item in the requirements specifications.
- A quick look at the existing system highlighted that the interfaces to the bank's subsidiaries, corresponding to 80 data flows, 50 control reports, and approximately 700 Function Points.

- Don't ignore the legacy systems.





3. Size Indicative Functionality Indicative Number of Instances

“... reporting includes, but is not limited to, the following reports....”

- Try to get this **open-ended** requirement capped or a cap to an overall reporting requirement.
 - At least, get an indicative number of reports for this type of requirement e.g. 10, 20, 50. (“capture the vision”)
 - If replacing an existing system, use the number of reports in the existing system as an aid.
-
- A legacy system encapsulates many years’ accumulation of knowledge – **don’t just ignore or discard it!**





3. Size Indicative Functionality Indicative Data Groups

- There will be other Logical Data Groups mentioned. **Name** as many as you can but also get an idea of approximately how many.
- These will (generally) be in the category of “**non-primary business**”
 - **Reference Logical Data**
 - These Data Groups enable the activities against the primary business data or against other Reference Data Groups.

598

When recording an order, the system must allow a user to select a customer based on a customer name.

- **Application and Operation Support Logical Data**

- These Data Groups support the operation of the application.

Once again, don't ignore the legacy systems as a source of how many.



4. Size Implied Functionality

Implied

There is no stated requirement for this functionality but its existence is implied to support other requirements.

Look for

- For the Logical Data Groups, there is often 'prescribed solution' requirements about entering data but **no clear requirements** for modifying or deleting or just looking at it.
 - For each Logical Data Group, examine the Logical Transactions already recorded against it and extend this list to make a basic life-cycle.
- Audit and Control Reports –
 - E.g. Check every interface and data load for Control Reports.
 - Include some allowance for Audit Reports
- Reporting in general





Using General Observations on Growth to Advantage

- Some **observations** of changes between early specification and actual delivery:
 - The overall ‘average’ **Complexity** of Logical Transactions **increases**.
 - One Logical Transaction ‘**splits**’ into many. In particular beware of “**generic**” Logical Transactions or Transactions with a lot of **Reuse** code.. They may hide different sets of business rules.
 - The **number** of Reference Data Groups increases.
 - **Interfaces** between applications are more numerous than planned for.
 - **Data take-on** and **conversion** is not sufficiently catered for.
- Early in the life cycle, **err on the size of generosity** with both the number of instances and complexity defaults.

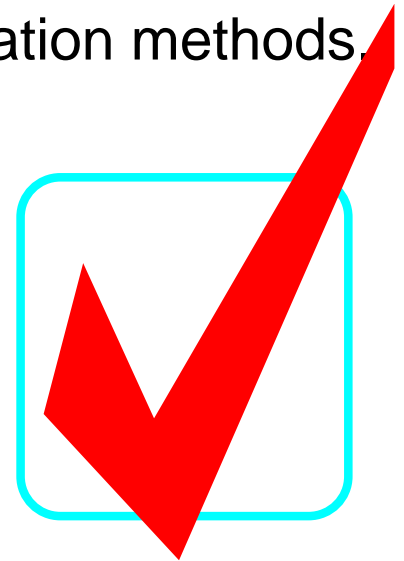


5. QA the Count

Have we Sized the Right Project?

- Quality Assuring the Functional Size must go beyond showing consistency with the rules of the chosen Functional Size Measurement Method.
- To be a truly useful input to early life cycle activities such as estimation, cost-benefit analysis, etc, the project's assessed functional size must align with the functionality to be delivered by the project.
- It must also align with the vision driving other estimation methods.

- **Scope:** The sizing **includes** all functionality to be delivered as the Water Register with the **exclusion** of any functionality for the data cleansing and initial loading of data from the existing Water Authority systems.



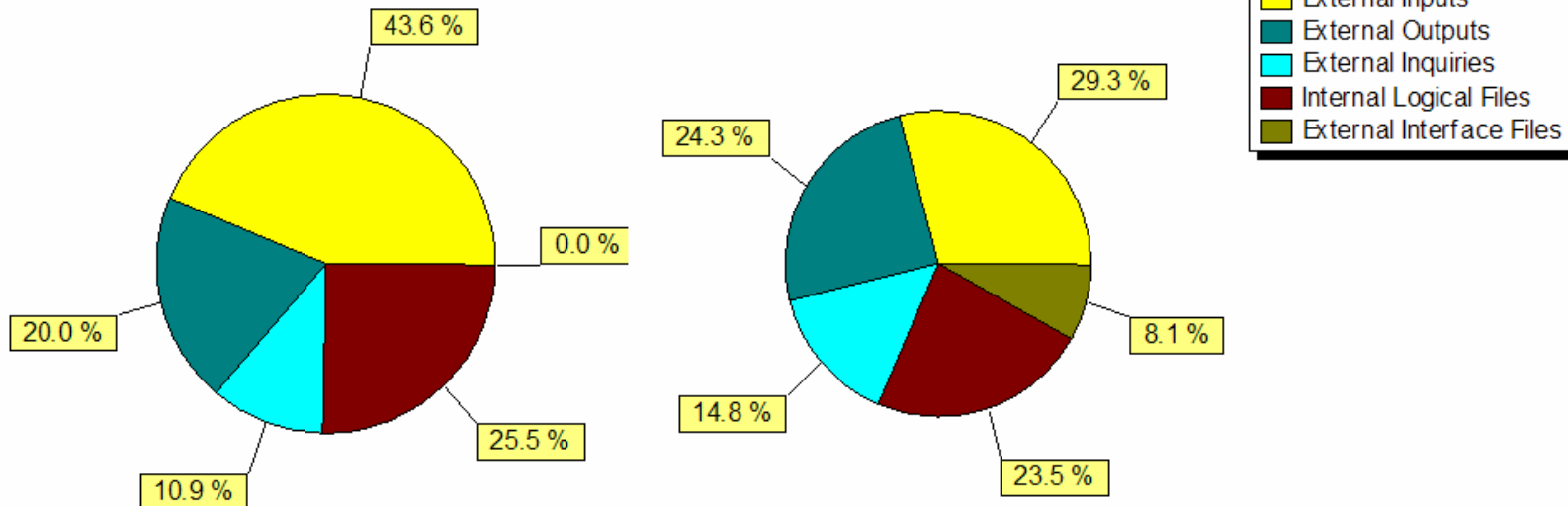
5. QA the Count

What! - Use Metrics!

- Industry and organisational benchmark data can quickly and easily highlight situations where Function Point Counts need checking.

Function Type	Contribution (%)		Function Ratios (<i>Transactions : Files</i>)	
	Project/Application	Industry Average ¹	Project/Application	Industry Average ¹
1 External Inputs	43.6	29.3	4.0 : 1	3.2 : 1
2 External Outputs	20.0	24.3	1.2 : 1	1.7 : 1
3 External Inquiries	10.9	14.8	1.0 : 1	2.6 : 1
4 Internal Logical Files	25.5	23.5	NA	NA
5 External Interface Files	0.0	8.1	NA	NA

Relative Contribution of Function Types



5. QA the Count

Fully Document the Count

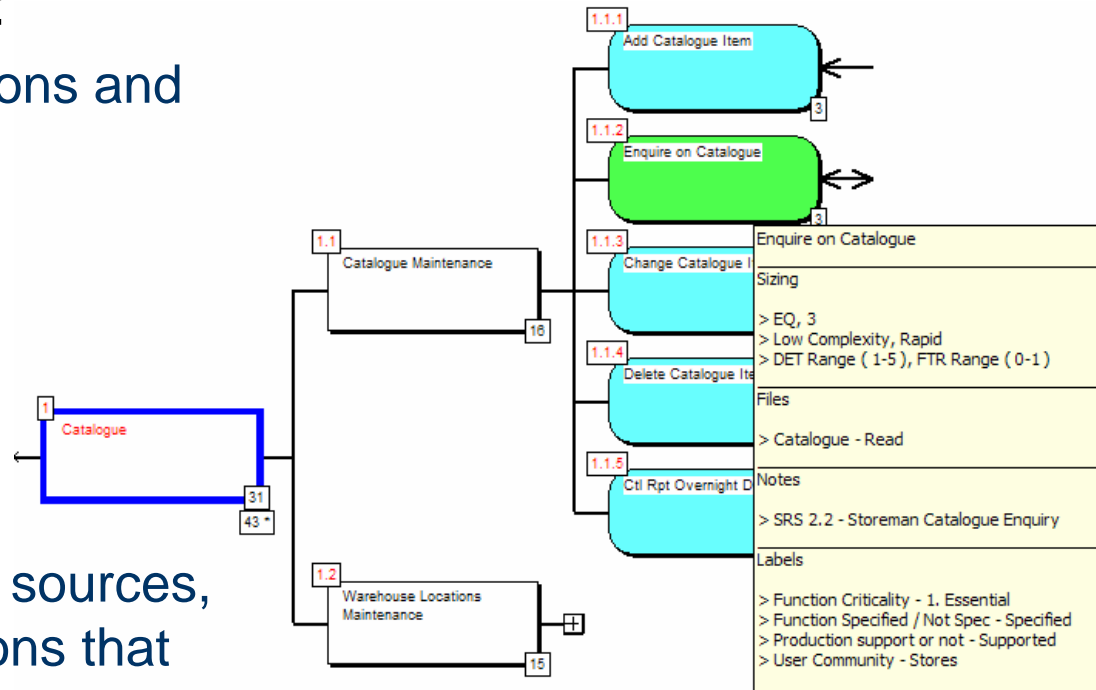
- **Never provide just the functional size.** Always provide the client with a full **Functional Model** of the project.

The model **must** include:

- All the Logical Transactions and Logical Data Groups

For each:

- Its Confidence Indicator
- Cross references back to information source
- Details of all information sources, assumptions and decisions that have impacted the determined functional size.



5. QA the Count

Walkthrough Functionality with the Client

- Always walkthrough the identified project functionality with the client.
- Doing so:
 - Aids in **QAing** the Function Point Count.
 - Allows the client to **take ownership** of the Function Point Count and encourages its ongoing use throughout the project for such tasks as scope management and earned value analysis and assists the client by providing an additional functional based view of the project.
- For prescribed functionality, have the client **confirm that it is correct.**
- For other functionality, let the client **confirm that it is of the correct order of magnitude and is consistent with the client's expectations regarding the software solution.**





6. Report the Count Some Suggestions

- Report functional size as a range, not as a point value. Indicate Low, High and Likely.
- Base the range on a logical analysis of the project or application size, **not** on an arbitrary plus or minus error figure. Your count is probably closer to the Low end of the range.
- Include a tabulation of the Confidence Indicators.

CONFIDENCE Indicator	No of Logical Transactions	Unadjusted Function Points	%
Clearly Stated	431	1841	63
Indicative	126	545	18
Implied	129	515	19
Total	686	2901	





Conclusion

- Make your sizing process an open book. Approach it in a methodical manner.
 - Any prediction about a project's size will be uncertain to some extent. Try to **quantify that uncertainty**. Say what you know and where you are providing a professional assessment for what you don't know.
- "If you don't make a serious effort to size the product, then your schedule estimates are based on nothing more than wishful thinking."

Tom DeMarco & Timothy Lister, *Waltzing with Bears*

Thank You