Effort Estimation with Full Function Points – Practical Experiences and Guidelines

Christof Ebert, Vector Consulting Services
Hassan Soubra, ESTACA, France
Agenda

- Introduction
- Estimation Principles
- Functional Size Measurement and Function Points
- Case Study
- Summary and Conclusions
How much time does the blue car need to its final destination?

It depends on several parameters:
- distance to drive (quantitative: Km, absolute scale)
- average speed (quantitative, ratio scale)
- weather conditions (qualitative: ordinal scale)
- traffic jams (unpredictable: risk to be evaluated)
- ...
Estimation – Daily Life (At Least Ours)

- **How much effort** will the project need to complete

- It depends on several **parameters**, such as:
  - functional size (quantitative: absolute scale)
  - typical productivity (quantitative, ratio scale)
  - employee motivation (qualitative: ordinal scale)
  - supplier delays (unpredictable: risk to be evaluated)
  - …
Agenda

- Introduction
- Estimation Principles
- Functional Size Measurement and Function Points
- Case Study
- Summary and Conclusions
Goals, Estimates and Plans

Goals
- External
- Business needs
- Examples: requirements, target cost

Estimates
- Internal
- Constrained by dependencies, uncertainties
- Examples: effort, duration

Plan
- Break-down of a goal to activities and milestones in order to reach this goal
- Relates goals and estimates to best possibly reach the goals
- Approach: Win-win
- Needs clear commitments of all impacted stakeholders

Understand, adapt, commit
Use Predictions to Assess Risks, Feasibility and Progress

Assess supplier viability, contract reliability, cost and product quality – independent whether it is components or outsourcing

- Standard estimation techniques and tools (e.g., COSMIC FP, QSM) for feasibility, health and cost
- Risk assessments upfront and during the project
- Predictions for progress related to your needs
- External audits if you don’t trust process quality
Agenda

- Introduction
- Estimation Principles
- Functional Size Measurement and Function Points
- Case Study
- Summary and Conclusions
Functional Change Requests Between Customer and Supplier

SW change request

Effort estimation

Bench-marking
Functional Size Measurement and Function Points

Function Point method application

- Analysis and translation
- Modeling and mapping
- Unadjusted functional size
- Adjusted functional size

Measurement context
Mapping rules
Estimation method
Proprietary parameters

Note: COSMIC functional size measurement is based on the ISO 19761. We refer here to COSMIC and Function Points covering functional size measurement and COSMIC function points
Need for Defined and Repeatable Development Process

- **Development process:** Defined process with known parameters
- **Project management:** Consistent tracking of all activity categories
- **Configuration management:** Consistent documentation of all changes
- **Quality management:** Consistent level of reviews, test, regression, etc.
Need for Consistent Specification of Requirements and Design

- **Functional change request**
  - SW requirements analysis
  - SW architecture design
  - SW module design
  - SW module implementation
  - SW integration and integration test
  - SW module test
  - SW validation before delivery

**Requirements and design specification:** Consistent level of documentation

**Modeling:** Same method, notation, semantics and visibility

**Change documentation:** All changes are covered and clearly marked
Generic Rules for Measuring Functional Size

- Ensure consistent application of a defined process during analysis of functional requirements and their mapping to a design proposal in the modeling language.

- Ensure a consistent effort reporting.
  - Focus on the changed part only.
  - No mix of different variants.
  - Always the same tasks are detailed.

- Always determine functional size on same level of detail.

- Each data movement is rated with 1 function point.

- Data movements are counted once.

- For changes only changed data movements are counted.

There is no automatic standard for counting because each modeling approach and environment need specific “translations”.

Insufficient data quality and environmental constraints need experienced counting to avoid errors and weakening the method.
Specific Rules for Functional Size of Embedded Systems

- **Data Flow**
  - All external data movements (e.g. external signals) which are variables (read/write) are counted.
  - Inputs that are used several times are counted only once.
  - Do not count fixed external values (e.g., 0, 1, 2, ..) that are changing (e.g., before: C=1, after: C=2) or that are used for comparisons only (e.g., C>1).
  - Data type conversion, unit changes etc. are not counted.
  - Measurement points, which are not part of regular functionality, are not counted.

- **State Machine**
  - State machine automatically count 1R and 1W.
  - Count 1X for the data (variable) leaving the state machine.
  - Count 1E for the data (variable) coming from outside the state machine.
  - Unique Data staying in the state-flow is not counted.

- **Hierarchy**
  - E and X data movements are only counted on the lowest abstraction level, not again on the higher level.

- **Reuse**
  - Internal reuse: If the same pattern is used at different places in the same context, with exactly same structure (flow, labels, operations), it is only counted once.
  - External reuse: If a structure is reused from a different context as copy-paste without changes, it is counted with a fixed FP amount (typically 1-3 FPs) which depends on integration effort, complexity, availability of test cases.
Agenda

- Introduction
- Estimation Principles
- Functional Size Measurement and Function Points
- Case Study
- Summary and Conclusions
Case Study: Focus on Functional Changes

Initial structure BEFORE the change

Function 1

in 4

in 2
const 1

in 3
false

x
&

Functional Change

in 1
const 2

in 5

+ out 2

out 1
Case Study: Levels of Hierarchy

2 Levels of Hierarchy are described in this sample functional change request.
Case Study: Measuring Functional Size of Changes

\[
\text{Size}_{\text{FP}} (\text{Change}) = \text{size(added data)} + \text{size(modified data)} + \text{size(deleted data)} = 7
\]

(\textbf{Function Point})

Initial structure BEFORE the change. Not counted.
Case Study: Mapping Functional Size to Effort

Unadjusted Functional Size

Model application (Filter criteria, environmental factors, mapping size to effort)

Effort

Project size (in FP)

Project effort
FP Counting is the Beginning and not the End

Possible explanations for lower effort compared to predictive model:

- Unrealistic estimation
- Bad reporting
- Better performance
- Lower overhead cost

Experience database (projects with the same context)
- Predicted project parameters
- Currently planned project parameters

Note: Relationship of Function Points to duration and effort needs always to be tailored to environmental conditions.
Agenda

- Introduction
- Estimation Principles
- Functional Size Measurement and Function Points
- Case Study
- Summary and Conclusions
Benefits of Function Points

- Agreed model for measuring functional size
- Solid baseline for benchmarking
- Transparent effort estimations on the basis of functional changes
- Ad-hoc and fuzzy evaluations and negotiations for single SW changes are reduced
- Significantly increased efficiency and trust for better collaboration between supplier and customer
Recommendations – Method

- **Consider business impacts**
  - Clearly distinguish goals, estimates and plans
  - Improve estimation accuracy in line with your business needs

- **Establish repeatability**
  - Immature processes invalidate your overall estimation and ruin trust
  - Establish a robust process to report and store data

- **Use estimation to grow**
  - Continuously improve.
  - Don’t stay with the same parameters for longer than one year.
  - Challenge results and improve your efficiency each year

We achieved with many clients a preciseness of 10-20% within one year, which in most cases is sufficient.
Recommendations – Tools

- **A fool with a tool remains a fool**
  - Garbage in, garage out
  - Clarify the underlying data collection and estimation approach
  - Never use a tool to camouflage insufficient process (“the tools says so”)

- **Data is the resource, measurement brings the value.**
  - Verify and validate data before storing in history databases
  - Use standard measurements (e.g. what is a FP or a defect?)
  - Always analyze data to add information for the decision-making process

- **Carefully introduce an estimation tool**
  - Consider Total Cost of Ownership (TCO)
  - Provide adequate training and coaching on estimation principles

Do not rely on a tool, if there is no mature development process. Rather coach management and push for keeping commitments.
Software Measurement
Establish, Extract, Evaluate, Execute

by Christof Ebert and Reiner Dumke

The book to support this keynote with backup data, case studies and industry experiences – for better performance.

"Few organizations have really institutionalized measurement of their products and processes. This book is bang up-to-date in both fields and packed with practical advice. For every software engineer."
- Charles R. Symons
Contact us – we would be happy to support you!

Phone +49 711 80670-0  www.vector.com/consulting
Fax +49 711 80670-444  consulting-info@vector.com