Estimating Real-time software projects with the COSMIC FSMM and the ISBSG data repository

SOGETI

COSMIC-FFP

ISBSG
Delivering IT Confidence

Estimating Real-time software projects with the COSMIC FSMM and the ISBSG data repository

Harold van Heeringen
Software Cost Engineer, Sogeti Nederland B.V.
ISBSG president
NESMA board

iCEAA conference
New Orleans, (LA, USA) June 2013

SOGETI

Overview

Software Projects
Software Project Estimation
Software Cost Engineering
Real-time and embedded software
FPA and COSMIC
ISBSG repository
How to estimate RT projects
Conclusions

Software projects

Software project industry : low maturity

- Low estimation maturity
- No or little formal estimation processes
- No or little use of historical data

Lots of schedule and cost overruns

- Standish Chaos reports: Most projects fail or are at least unsuccessful

Low customer satisfaction rates

- In Europe: only slightly higher than the financial sector

Software project estimation

Most of the projects are estimated by 'experts'

- Bottom up, task by task effort estimation

Usually very optimistic (>30%)

- Experts estimate, but other people (juniors) do the job
- Forgotten activities (e.g. testscript reviews)
- No feedback loop with past projects: experts don't learn from past estimates and actuals

No scenario's: duration, team size, etc.

Not objective, transparent, verifiable and repeatable

Not defendable!

- **'Easy' to push back by stakeholders**

No risk assessment (distribution of the estimate)

Software Cost engineering

Not a real profession yet

- Consultant software metrics
- Estimation officer
- Bid specialist

Parametric estimates

- Functional size measurement → size of the software
- Productivity rates from historical data or industry data
- Parametric estimation tools

Objective, repeatable, transparent and verifiable

Defendable!!

- **'Impossible' to push back by stakeholders**

Risk assesment (distribution of the estimate)

Real-time vs. Business software

Real-time software: Software whose correct functioning depends on the results produced and the time at which these results are produced

- Event driven (timer events)
- Interrupt driven

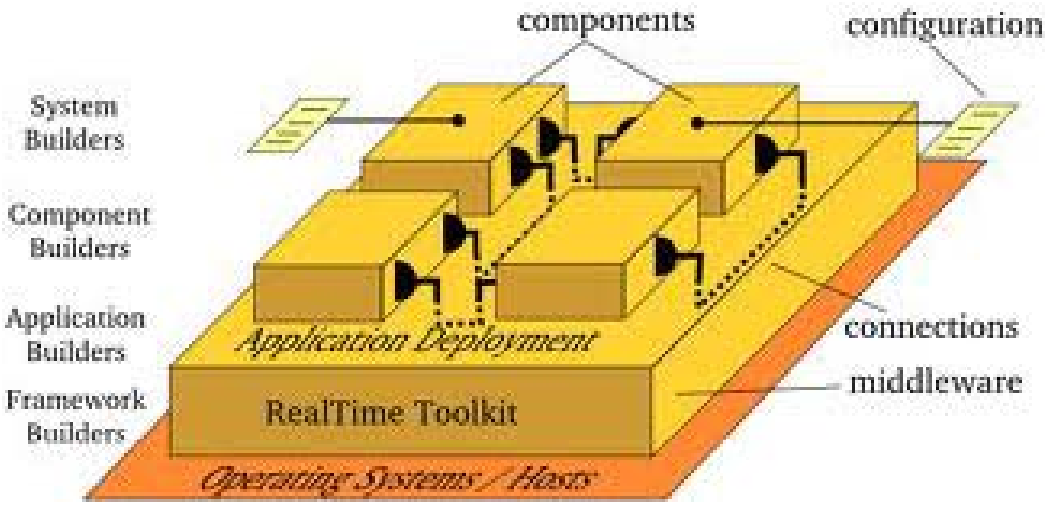
Automotive industry, Avionics industry, etc.

Business Application software

- Store administrative transactions
- User interaction through keyboards, screens and printers

Banking, Insurance, etc.

Real-time vs. Business software



Estimating RT Software projects

Most software cost engineering methods are based on sizing the functional requirements of the software

- User requirements documents
- Functional designs
- Data models

Since the 70's, the method Function Point Analysis (FPA) was the standard for this (IFPUG / NESMA)

Many projects and systems have been measured in FPA and loads of historical project data is stored

However, FPA is only applicable to administrative software domain

So… how to measure the size of real-time and embedded software?

FPA vs. COSMIC

	FPA	COSMIC
Domain	Business applications	Business applications, Real-time applications, Infrastructure software
Data model required?	Required	Not required (but useful)
Measurement of separate components	Not possible	Possible
Size limit per function	Yes	No
Benchmarking data	Many	Some (ISBSG R11: 450)
Early sizing	Based on data model	Based on process model

The COSMIC method

End 1990's – **COSMIC** was founded by practitioners and scientists
Non-profit, volunteers
Open method, free documentation (www.cosmicon.com)

Overcomes a number of drawbacks of the FPA method
- Based on solid software engineering principles
- Ratio scale of the size
- Applicable to Real-time, Embedded and Infrastructure Software!

Fairly new method – less historical data available

Use is growing, especially in Real-time and Embedded software
- Renault, Siemens, Nokia, Eurocopter, Philips

COSMIC documentation

Documentation Overview & Glossary of Terms

PRINCIPLES & RULES for the COSMIC METHOD Version 3.0.1:

ISO/IEC 19761:2002 COSMIC-FFP A functional size measurement method*

Method Overview	Measurement Manual (V3.0.1)	Advanced & Related Topics

← Beginners →

← Practitioners →

← Experienced Practitioners →

DOMAIN-SPECIFIC SUPPORT DOCUMENTS

Guidelines
- Business Application v1.1
- Real-time Software**
- Data Warehouse**
- Service-Oriented Architecture**

Case Studies (x n)
- Business Application
- Real-time software

SOGETI

COSMIC: layers

ISO standard - *ISO/IEC 19761:2011*

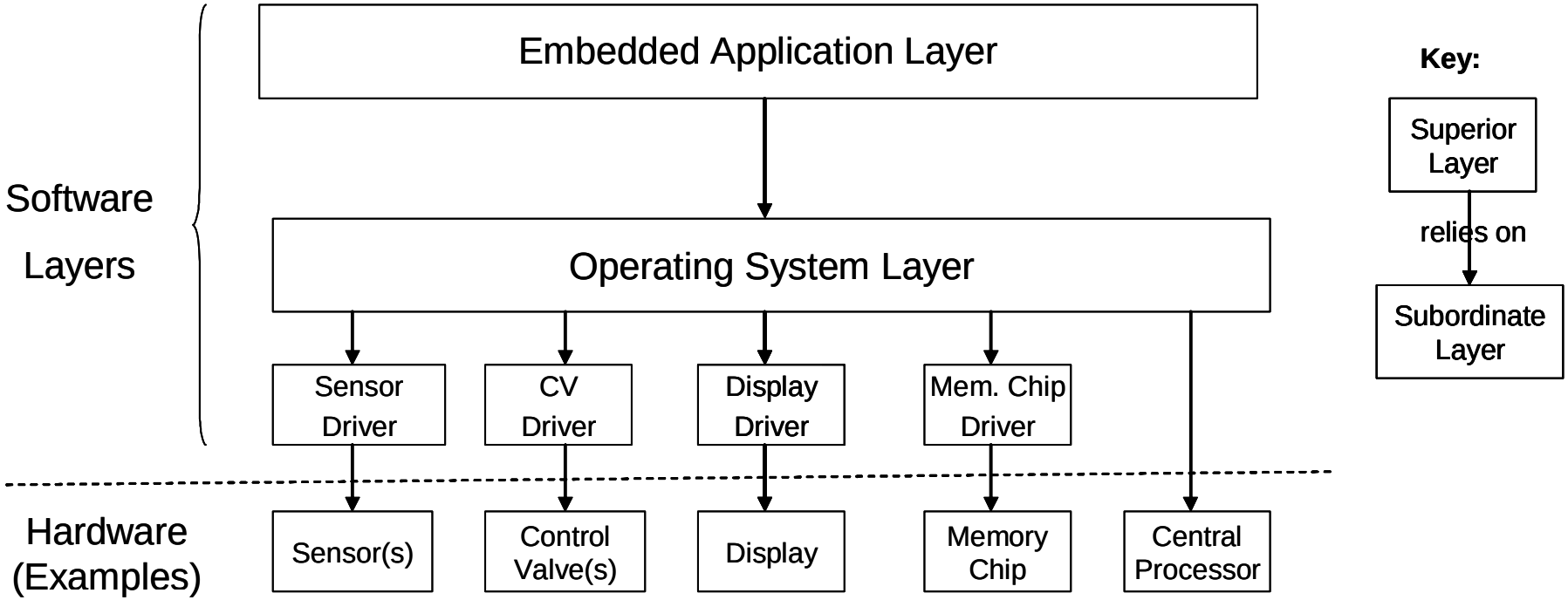
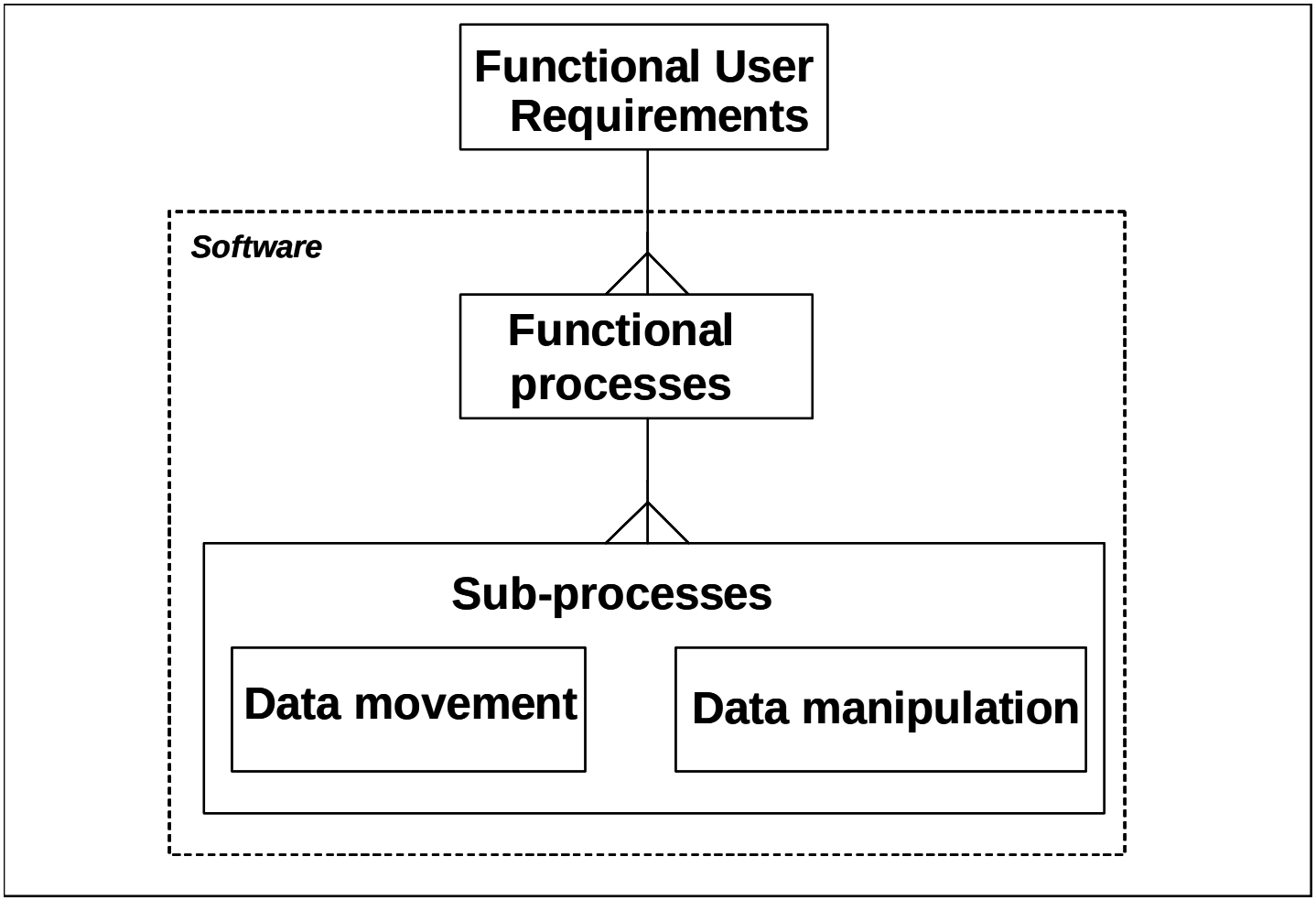
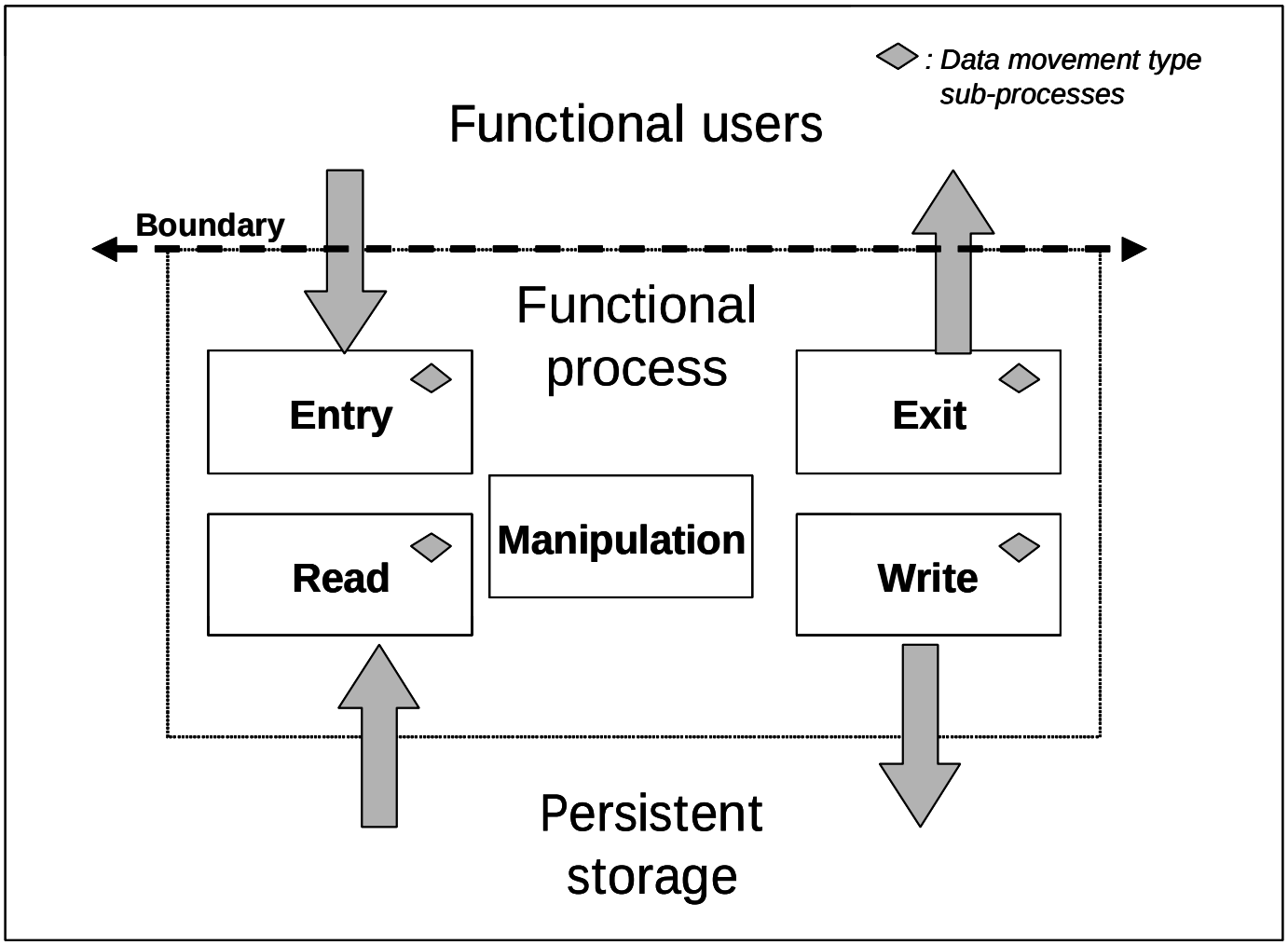


Fig. 2.2.2.2 Typical layered architecture for a real-time embedded-software computer system

Functional processes



Data movements



Example RT COSMIC

Context diagram

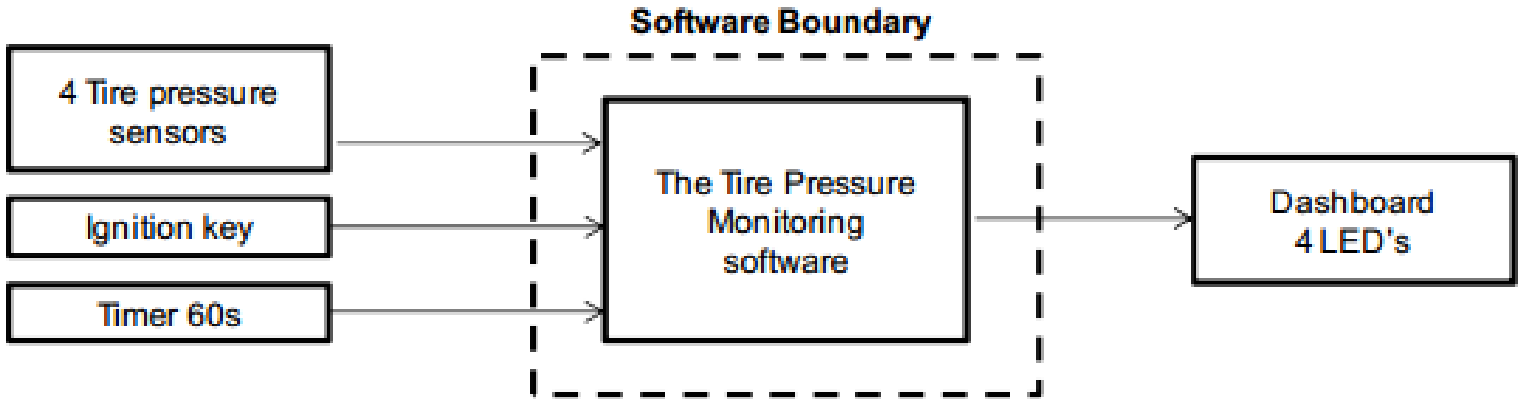
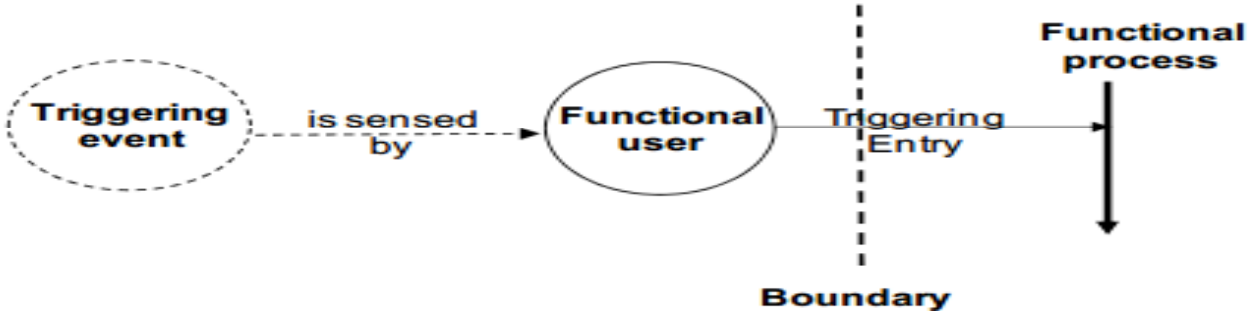
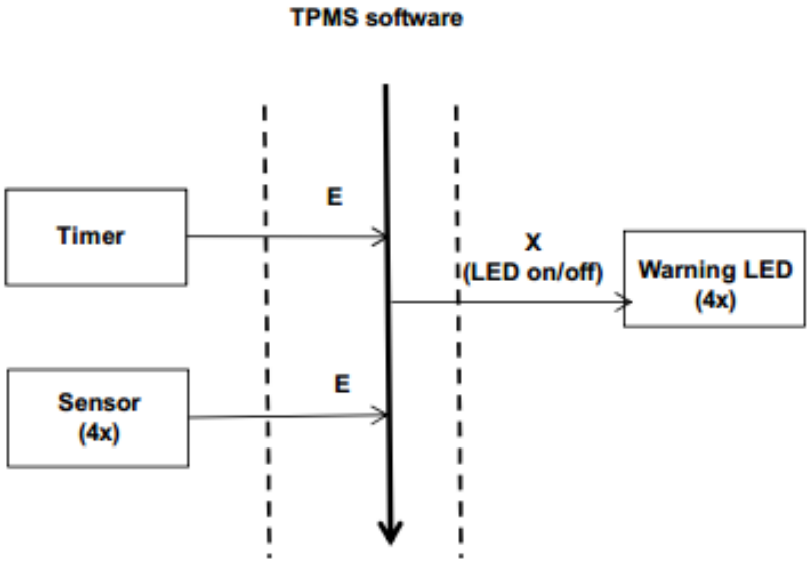


Figure 4.4.1 – TPMS, context diagram

Triggering event	Functional user	Functional process
Timer signal	Timer	Start TPMS software



Example: TPMS start



Functional process Start TPMS software

DM	Data group	Remark
E	Start signal	Triggering event, by timer
E	Wheel sensor pressures	Acquisition of the four 'dumb' sensors' data, one for each wheel
X	Signal to array of four warning LED's	LED on, if needed, one for each wheel. Check of pressure is data manipulation and accounted for by this Exit data movement

SOGETI

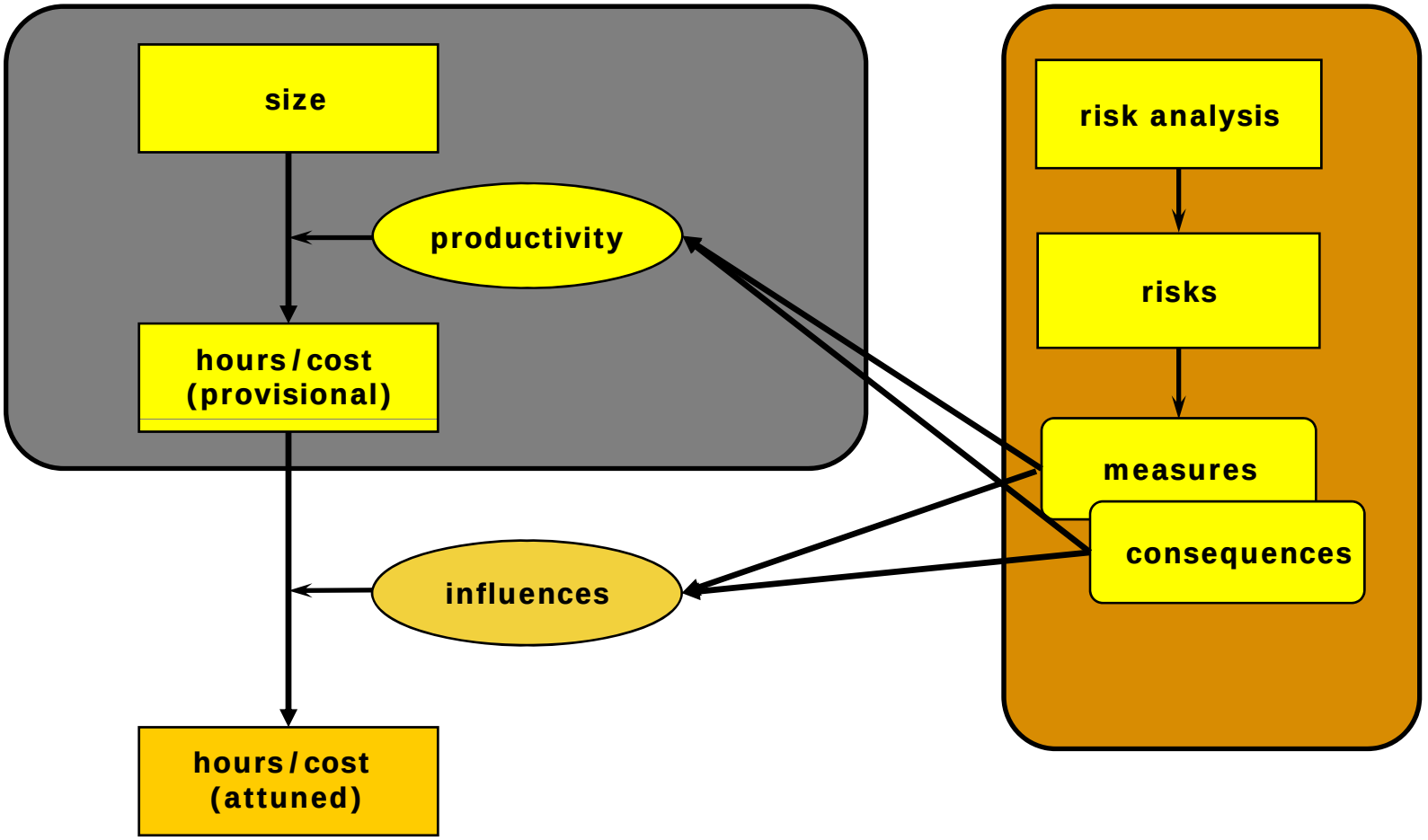
What to do with COSMIC size

Now we can size real-time and embedded software
- Functional Size per layer
- Functional Size per component
- Objective, repeatable

What to do with the results?
- Project effort estimation
- Project benchmarking
- Use in Request for Proposal management

What about historical data?
- Company data
- **Industry data**

Estimation model



SOGETI

Historical data: ISBSG repositories

International Software Benchmarking Standards Group
Independent and not-for-profit
Members are non-profit organizations, like IFPUG and NESMA

Grows and exploits two repositories of software data:
- New development projects and enhancements (> 6000 projects)
- Maintenance and support (> 1000 applications)

Everybody can submit project data
- DCQ on the site
- Anonymous
- Free benchmark report in return

Special reports, Practical Project Estimation book, Compendium
Portal to access the project data

ISBSG

Mission: “To improve the management of IT resources by both business and government, through the provision and exploitation of public repositories of software engineering knowledge that are standardized, verified, recent and representative of current technologies”.

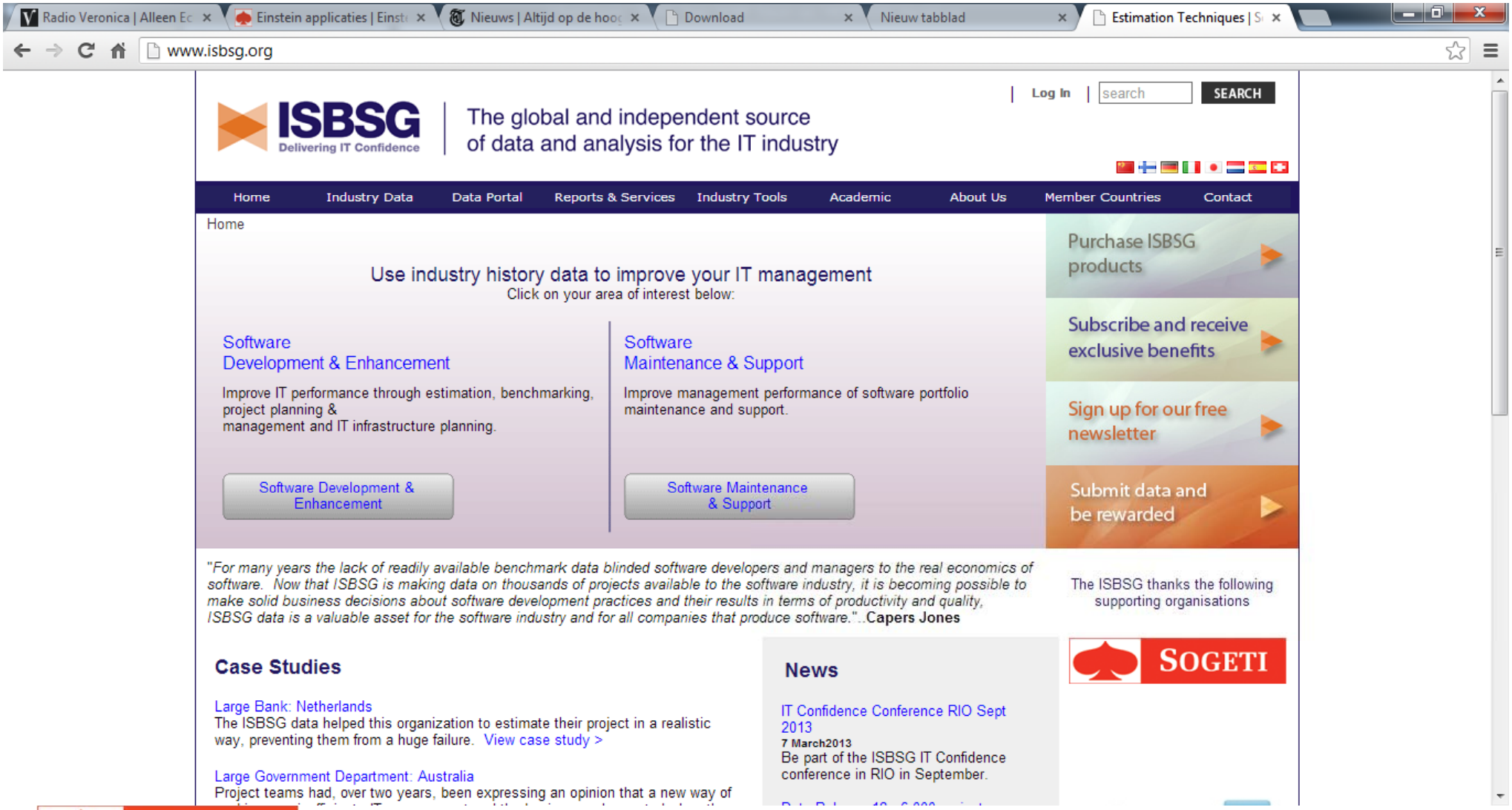
All ISBSG data is

- validated and rated in accordance with its quality guidelines
- current
- representative of the industry
- independent and trusted
- captured from a range of organization sizes and industries

Industry leaders around the world contribute to the ISBSG’s development, offering the highest metrics expertise worldwide

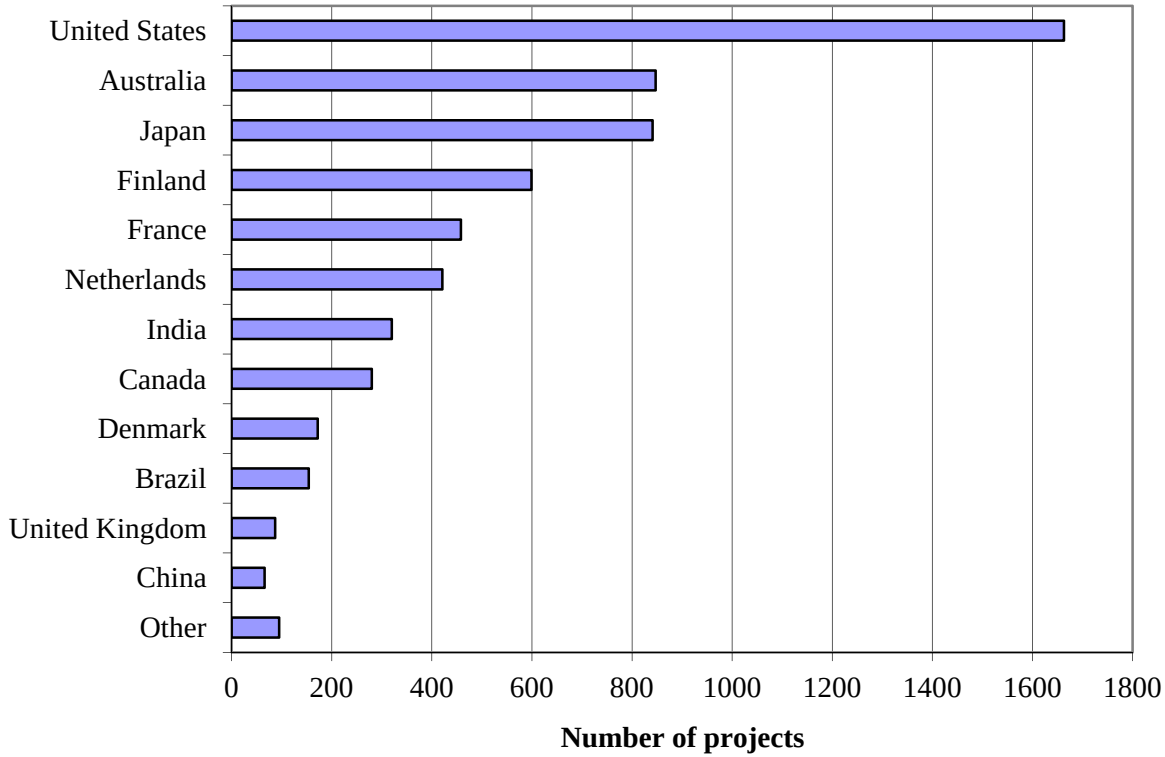
SOGETI

www.isbsg.org



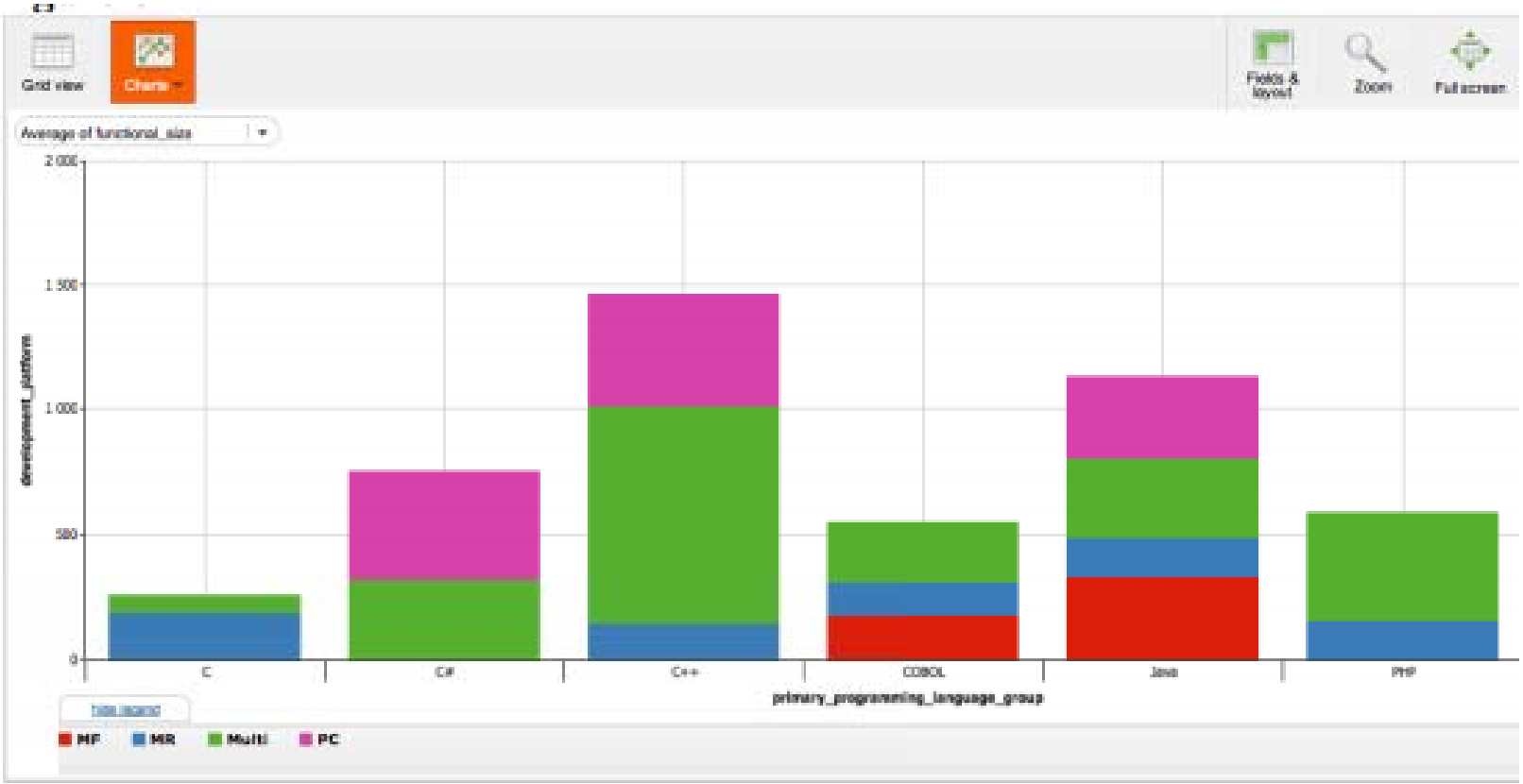
ISBSG New developments

R12: >6000 projects
Projects measured with COSMIC : >450 projects
Real-time projects: 272 projects



ISBSG data

Repository in MS Excel (open)
Use the data portal (portal.isbsg.org)



SOGETI

Estimate effort of the RT project

1. Size the separate components in COSMIC CFP

2. Determine the most appropriate PDR (Project Delivery Rate)
   Hours/CFP

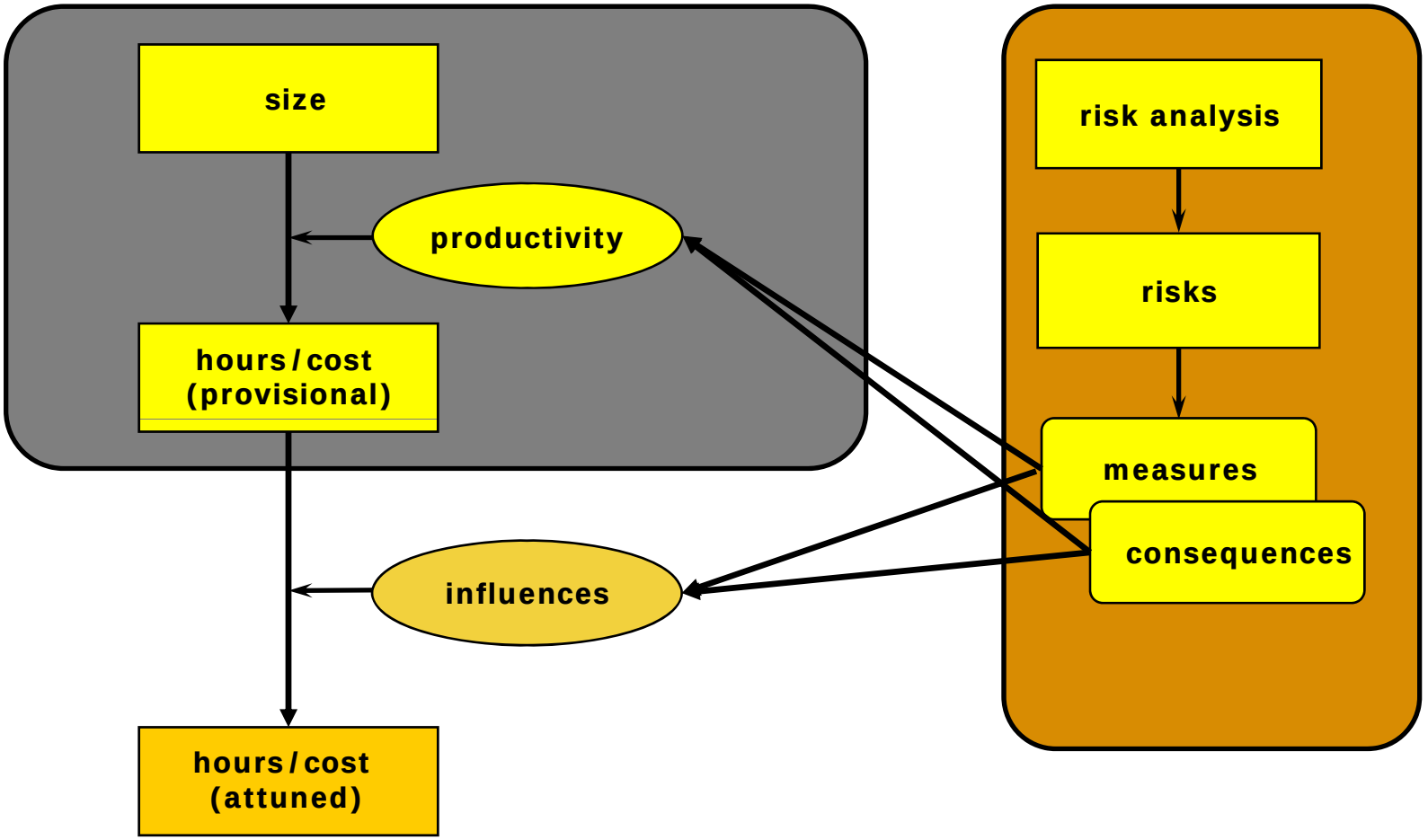
3. Calculate 'gross hours': Size per component * PDR

4. Determine influences and effect
   Schedule requirements
   Other non-functional requirements
   Other influences

5. Calculate total effort estimate

SOGETI

Estimation model



SOGETI

Example

Real-time system: 3 components

- Component 1: C++
- Component 2: Ada95
- Component 3: Java

Estimate the effort needed for the development of this system

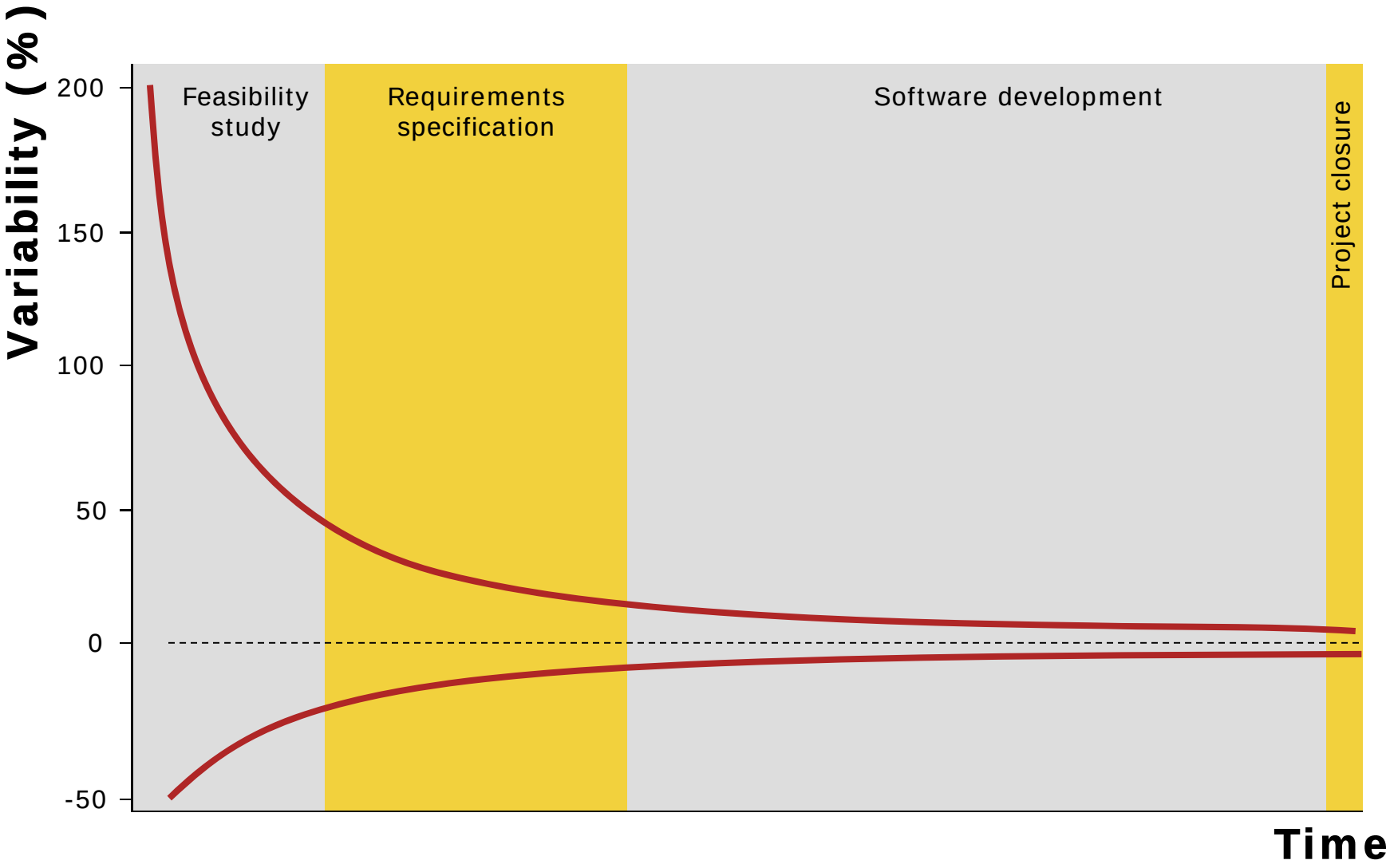
1. Size the components in CFP

Example

Component	Minimum Size	Likely Size	Maximum Size
A	100	150	220
B	30	50	100
C	210	250	300
Total	**340 CFP**	**450 CFP**	**620 CFP**

SOGETI

Uncertainty is part of the deal



SOGETI

2. Determine the PDR (h/CFP)

Data analysis in the ISBSG repository

- Select most appropriate projects
- Analyze the productivity data

Example:

- Type of application: Complex process control
- Real-time software
- Programming language: Visual Basic

Result: 30 projects,

- P25: 12 hours/CFP
- Median: 16 hours/CFP
- P75: 24 hours/CFP

3. Calculate 'gross hours

Component A	Minimum	Likely	Maximum
Size	100 CFP	150 CFP	220 CFP
PDR C++	12 h/CFP	16 h/CFP	24 h/CFP
Effort	1200 hours	2400 hours	5280 hours

Component B	Minimum	Likely	Maximum
Size	30 CFP	50 CFP	100 CFP
PDR Ada95	24 h/CFP	28 h/CFP	34 h/CFP
Effort	720 hours	1400 hours	3400 hours

Component C	Minimum	Likely	Maximum
Size	210 CFP	250 CFP	300 CFP
PDR Java	8 h/CFP	12 h/CFP	20 h/CFP
Effort	1680 hours	3000 hours	6000 hours

Total	3600 hours	6800 hours	14680 hours

SOGETI

4. Assess influences

Risk analysis
Which factors are different from ‘usual’

Example:

- Extreme quality control because of safety requirements
  - Min +500 hours, likely +800 hours, Max +1000 hours
- Numerous parties involved that have to discuss and agree on interface
  - Min +200 hours, likely +400 hours, Max +500 hours
- Total : Min. +700 hours, Likely +1200 hours, Max. +1500 hours

5. Calculate total effort estimate

	Minimum	Likely	Maximum
Gross hours	3000 hours	6800 hours	14680 hours
Influences	700 hours	1200 hours	1500 hours
Total effort	3700 hours	8000 hours	16180 hours

Calculate the cost of the effort: hours * average hour rate

SOGETI

Conclusions

Software Cost Engineering of Real-time and embedded software projects is now possible!

- COSMIC functional size measurement
- ISBSG data repository

Improve maturity of RT and embedded software suppliers
Increase transparency, predictability and defendability of this type of projects

Harold van Heeringen
Sogeti Nederland B.V.

@haroldveendam
Harold.van.heeringen@sogeti.nl

SOGETI