The following presentation was given at:

SCAF Workshop
"The SCAF 2017 Cost Estimating Challenge"

Tuesday 4th April 2017
BAWA Centre, Bristol

Estimating the cost for development of software applications

Duarte Rodrigues
Dr. John Erkoyuncu
Prof. Andrew Starr

4th April, 2017

www.cranfield.ac.uk

The Team

Duarte Rodrigues
d.rodrigues@cranfield.ac.uk

Areas of expertise:

- Control Systems
- Mathematics and Applications
- Systems Engineering

Dr. John Erkoyuncu
j.a.erkoyuncu@cranfield.ac.uk

Areas of expertise:

- Computing, Simulation & Modelling
- Product and Service Design
- Through-life engineering Services

Prof. Andrew Starr
a.starr@cranfield.ac.uk

Areas of expertise:

- Aeronautical Systems
- Engines, Powertrains & Alternative Fuels
- Industrial Automation
- Systems Engineering
- Through-life engineering Services

Research Aim

The cost challenge

- This research aimed at building an estimate of the cost for development of a third-party application to be used on OS, android and Windows platforms for major automotive platforms in UK.

Research Objectives

1) Understanding the research context;

2) Identifying the attributes impacting on software development cost;

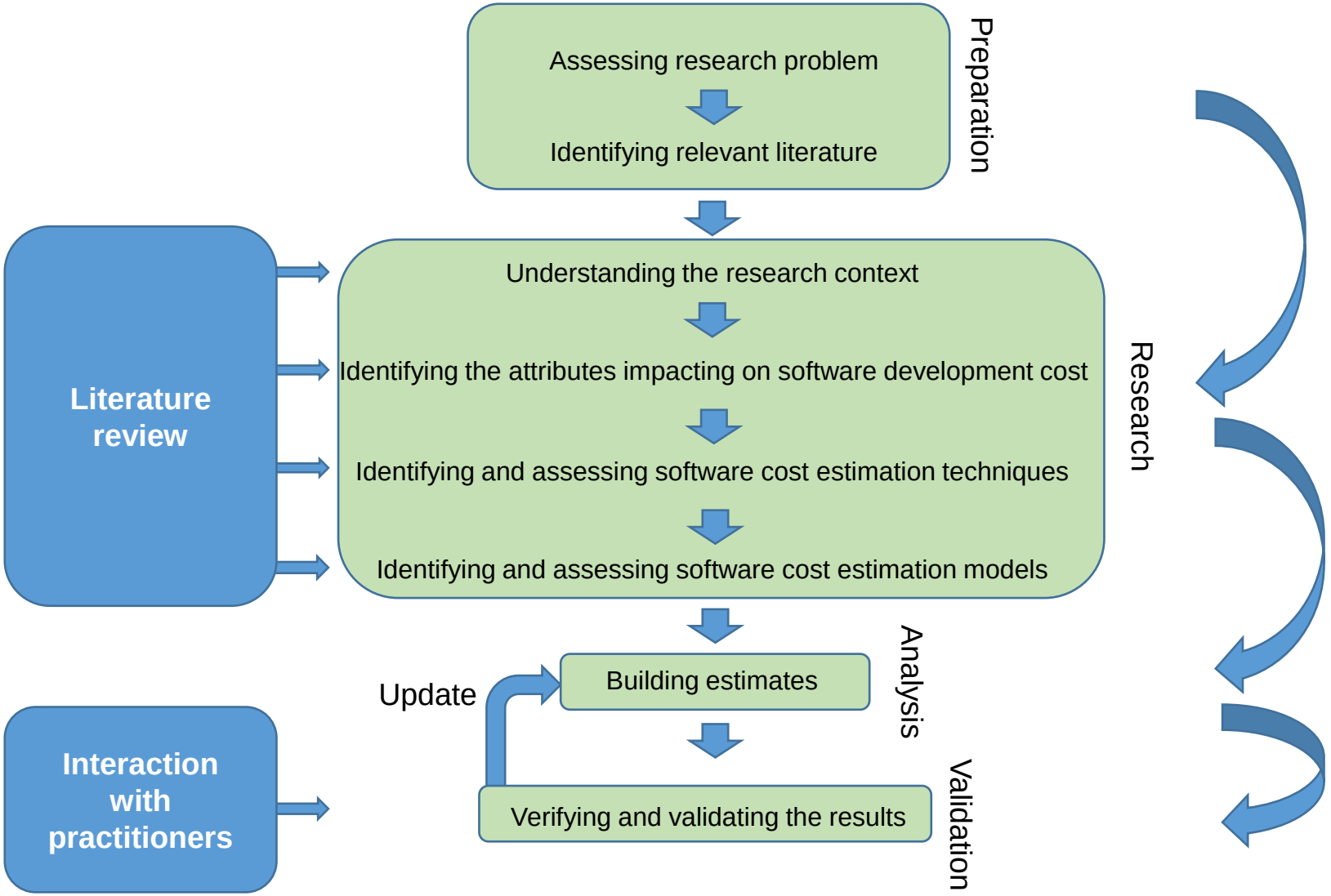
3) Identifying and assessing software cost estimation techniques;

4) Identifying and assessing software cost estimation models;

5) Building estimates;

6) Verifying and validating the results.

Methodology



1) Understanding the research context

2) Identifying the attributes impacting on software development cost

3) Identifying and assessing software cost estimation techniques

4) Identifying and assessing software cost estimation models

5) Building estimates

6) Verifying and validating the results

Literature Findings

- The amount of software in luxury cars is growing exponentially (ref. 1)
- The software cost has also increased
- Premium cars include more and more functions that are enabled and driven by software
- The automotive industry has specific constraints and domain specific requirements in the field of software engineering (ref. 2)

Literature Findings

- The cost of software apps can be broken-down in:

1) Development cost  (~40-80%)

2) Maintenance cost  (~30-60%)

3) Warranty and license cost  (~10-30%)

(Ref. 3)

Literature Findings

Can software applications be reused/adapted?

- Typically, from one car generation to the next only 10% of the software applications are changed and enhanced, while more than 90% of the software is rewritten. The reason is a low level, hardware specific implementation, which makes it difficult to change, adapt and port existing code.

(Ref. 1)

Literature Findings

What makes automotive industry so unique in terms of software engineering?

- Wide range of different users;
- Specific maintenance situations;
- Specific critical functions;
- Specific context of operation of the systems;
- Heterogeneity of functions (e.g. embedded real time control to infotainment, energy management, software download functionality, on board diagnosis, error logging, etc.).

(Ref. 1-2,4)

Literature Findings

What drives the quality of the software apps?

Ability to customize	Lifetime	Risk of fragmentation
Availability/reliability	Performance	Design flexibility
Interoperability	Quality of service and support	Level of difficulty/ease of management
Scalability	Security	

(Ref. 2)

1) Understanding the research context

2) Identifying the attributes impacting on software development cost

3) Identifying and assessing software cost estimation techniques

4) Identifying and assessing software cost estimation models

5) Building estimates

6) Verifying and validating the results

Literature Findings

- The identification of the cost attributes is a critical part of any cost estimating process;
- The appropriate level of granularity required for identifying the cost attributes is specific to each cost model applied;
- During the attributes identification process, smaller cost attributes were aggregated into larger ones to facilitate data acquisition and implementation in cost models.

Cost attributes - List 1:

"Main" attributes	"Sub" attributes		
Software	Purchase price	Upgrades	Intellectual property/licensing fees
Hardware	Purchase price	Upgrades	
Support costs	Installation and set-up Documentations	Maintenance	Troubleshooting
Staffing costs	Project management	Systems engineering/development	Systems administration
Other costs	Peer support	Formal training	Casual learning

(Ref. 2)

Cost attributes - List 2:

What (product)	With what (means)	Who (personnel)	How (project)	For whom (user)
Size of the software	Computer constraints • Execution time • Response time • Memory capacity	Quality of personnel	Requirements project duration: • Stretch out; • Compression.	Participation
Required quality		Experience of personnel		Number of users
Requirements volatility	User of tool	Quality of management	Basis for project control: • Matrix organisation • Project organisation • Prototyping • Incremental • Linear development • Software development	Stability of user organisation, procedures, way of working
Software complexity	Use of modern programming techniques: • Information hiding • Chief programme team • Structured program • Top-down design	Availability of project		Experience of user with automation, level of education in automation
Level of reuse				
Amount of documentation				
Type of application				

(Ref. 5)

Cost attributes - List 3

Required reliability

Database size

Complexity software

Constraints execution time

Memory constraints

Hardware volatility

Response time constraints

Quality analysts

Experience with application

Quality programmers hardware experience

Programming language experience use modern programming techniques

Use software tools

Project duration constraints

(Ref. 5)

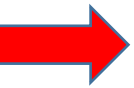
Cost attributes - List 4

Customer complexity	Staff productivity	System size
Customer geography	Skill level development	Project structure
Developer familiarity	Rate at each skill level	Target technology
Business function size	System category	Resources
Target system sophistication	Generic system type	Effort hours
Target system complexity	Operating system	Transaction volume

(Ref. 6)

1) Understanding the research context

2) Identifying the attributes impacting on software development cost



3) Identifying and assessing software cost estimation techniques

4) Identifying and assessing software cost estimation models

5) Building estimates

6) Verifying and validating the results

Software cost estimation techniques:

- **Analogy Method (60.8%);**
- **Expert Judgement (25.5%);**
- Capacity Problem or Parkinson's Law (20.8%);
- **Parametric Models or Algorithmic Cost Modelling (13.7%);**
- Pricing to Win (8.9%).

(Ref. 5,7)

Strengths and weaknesses of software cost estimation techniques

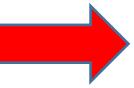
Method	Strength	Weaknesses
Analogy method	Based on representative experience	Representativeness of experience
Expert judgement	Assessment of representativeness, interactions, exceptional circumstances	No better than participants
		Biases, incomplete recall
Parkinson's law	Correlates with some experience	Reinforces poor practice
Parametric models	Objective, repeatable, analysable formula	Subjective inputs
	Efficient, good for sensitivity analysis	Assessment of exceptional circumstances
	Objectivity calibrated to experience	Calibrated to past, not future
Pricing to win	Often gets the contract	Generally produces large overruns

(Ref. 8)

1) Understanding the research context

2) Identifying the attributes impacting on software development cost

3) Identifying and assessing software cost estimation techniques



4) Identifying and assessing software cost estimation models

5) Building estimates

6) Verifying and validating the results

Cost estimation models for software development:

- Extreme Programming Evaluation Framework for Object-Oriented Languages (XP-EF)
- SEER-SEM
- SLIM
- PRICE-S
- Otreva
- Tusnua Designs
- Imason
- Kinvey
- COCOMO(II)
- Cost, Availability and Time Forecasting and Optimisation during Bidding (CATFOB) model

COCOMO model II: General Description

- The accuracy of the estimates produced by the algorithmic model depends on the system information that is available;
- Takes into account different approaches to software development such as code reuse;
- Provides a detailed cost breakdown estimation.

(Ref. 6,7,9,10)

COCOMO-II model: Includes pre-defined list of attributes

Product Attributes:	Required software reliability	Product Complexity	Documentation Match to Lifecycle Needs	Database size	Developed for Reusability	
Personnel Attributes:	Analyst Capability	Programmer Capability	Personnel Continuity	Application Experience	Platform Experience	Language and Toolset Experience
Platform Attributes:	Time Constraint	Storage Constraint	Platform Volatility			
Project Attributes:	Use of Software Tools	Multisite Development	Required Development Schedule			

Product attributes are concerned with required characteristics of the software product being developed.

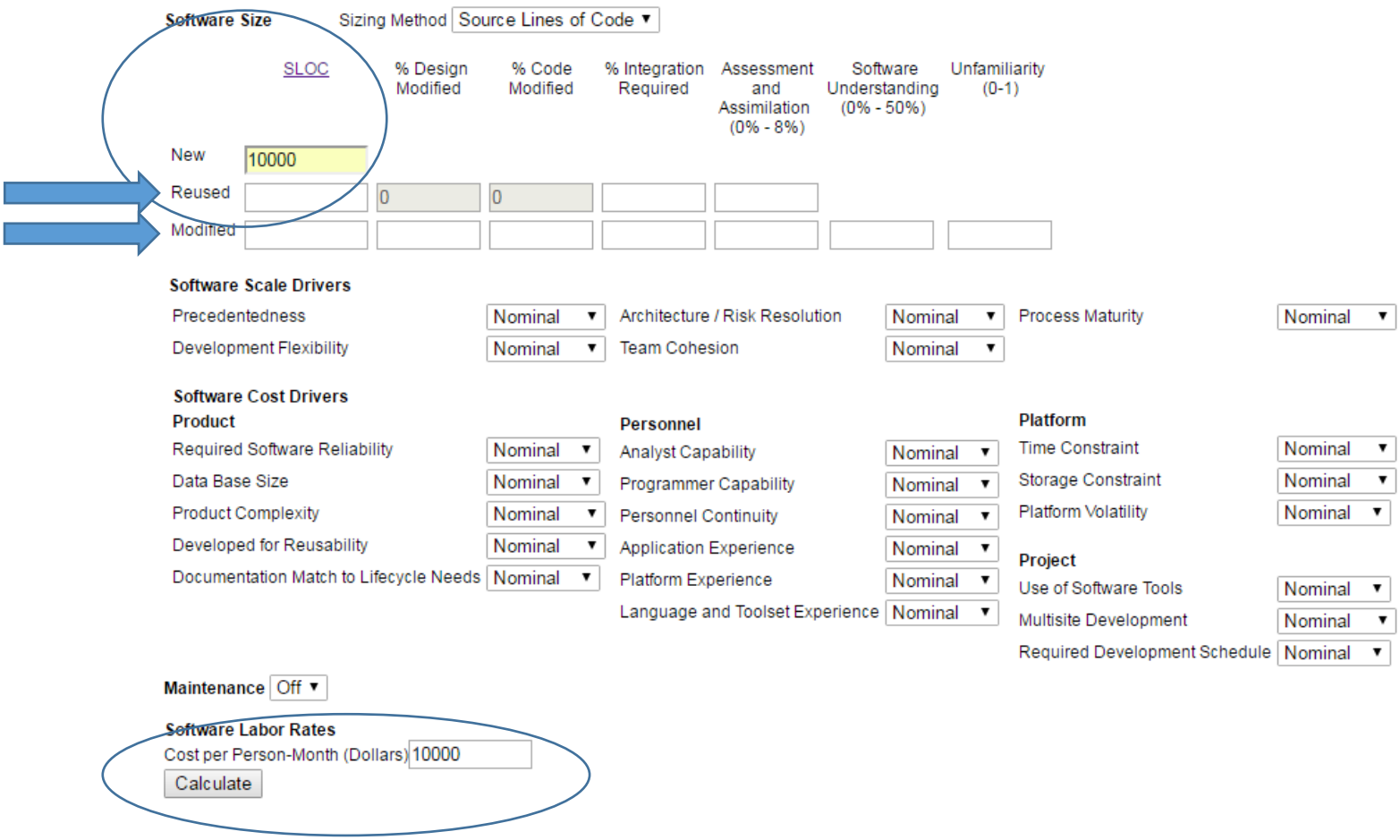
Personnel attributes are concerned with the experience and capabilities of the people working on the software development.

Platform attributes are concerned with the constraints imposed on the software by the hardware platform.

Project attributes are concerned with the particular characteristics of the software development project.

(Ref. 9,10)

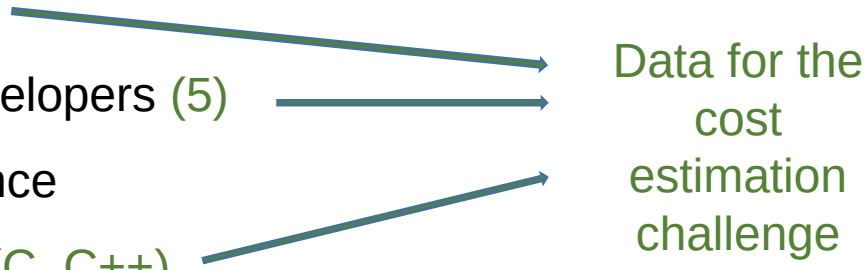
COCOMO-II model: Other input parameters



(Available at: http://csse.usc.edu/tools/cocomoii.php)

COCOMO-II model: Observations

- Performs a detailed cost analysis considering different parameters such as:
  - Code reuse (60%)
  - Number of code developers (5)
  - Developers experience
  - Programming code (C, C++)
  - Complexity



- Requires a considerable amount of input information to produce results
- The outputs are very detailed

CATFOB Model

- This model was formally developed during a PhD project that follow-on a partnership between a Babcock International and Cranfield university, aimed at optimising cost and availability estimate at the bidding stage of performance-based contracts (e.g. contracting for availability);

- The model can adapted to the context software cost estimation considering some assumption/relationships such as:
  - ➢Contract duration = software life-cycle (5 years)
  - ➢System availability = app size/complexity

CATFOB model: Input Requirements

- Historical information from comparable apps:
  - Different cost attributes (ideally 3-6);
  - Cost of each attribute;
  - App life-cycle and app size/complexity;

CATFOB Model Database ← Information Source

Project Database

	Attributes	new_App	new_App	new_App	new_App
	App Size (SLOC):	83517	400000	780000	2560000
	APP Life-Cycle:	10	10	10	10
		Investment (k£):	Investment (k£):	Investment (k£):	Investment (k£):
	Software	300	1000	2000	150000
	Hardware	420	1000	2000	150000
	Support costs	520	2000	3200	300000
	Staffing costs	360	2000	4000	400000
	Other costs	200	1000	2000	200000

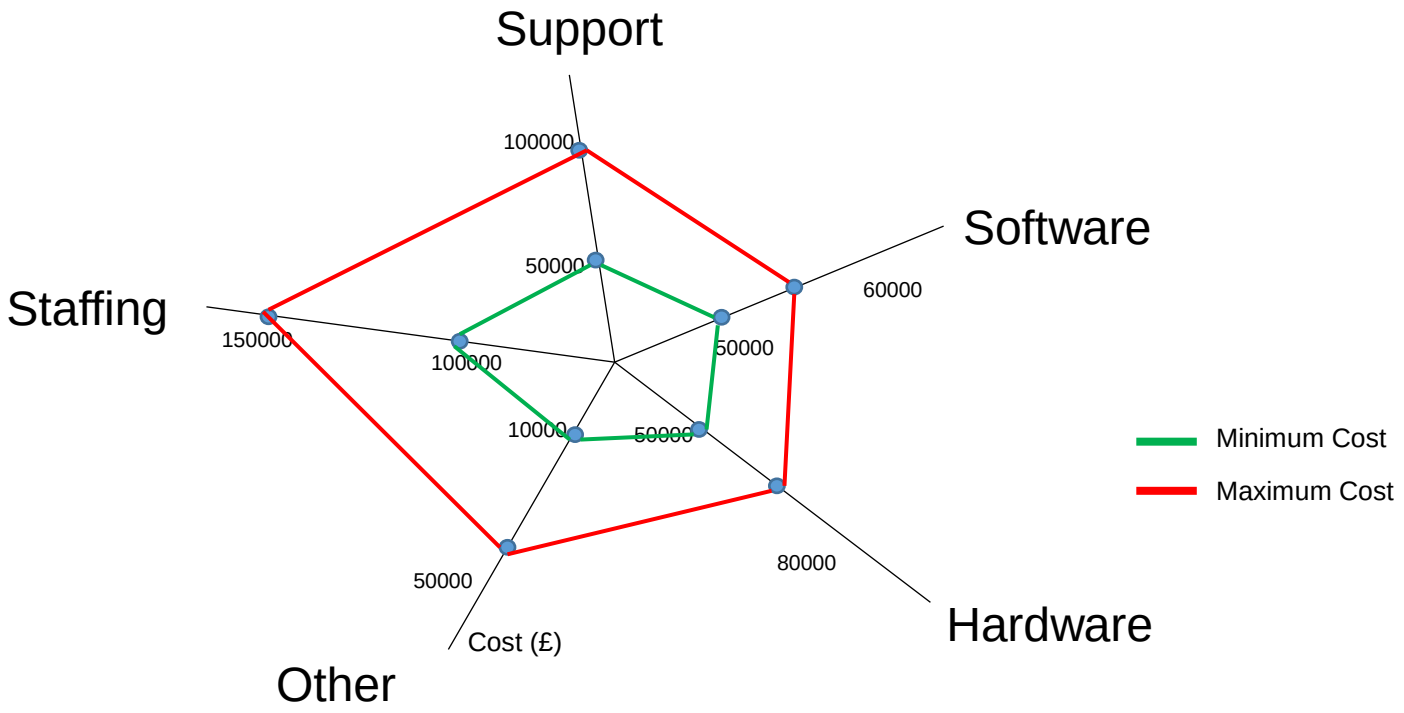
Software Cost and Effort Comparisons

Project	Lines-of-Code	Labor (man-years)	Cost ($ millions)
1989 Lincoln Continental	83517	35	1.8
Lotus 1-2-3 v.3	400000	263	7
Citibank AutoTeller	780000	150	13.2
Space Shuttle	25600000	22096	1200

(Ref. 11)

CATFOB Model: Input Requirements

- Requirements for the current App:
  - Minimum and maximum cost of each attribute;
  - App size and life-cycle targets;



CATFOB model: Observations

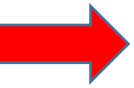
- Requires few cost attributes to be defined
- Database information is not extensive
- The outputs provide a cost break-down estimate per attribute defined
- Have an accuracy estimated between 70-80% depending on the calibration of the database
- Does not consider detailed cost analysis

1) Understanding the research context

2) Identifying the attributes impacting on software development cost

3) Identifying and assessing software cost estimation techniques

4) Identifying and assessing software cost estimation models



5) Building estimates

6) Verifying and validating the results

Possible software size/complexity measures:

(1) Size related measures: are based on some output from the software process.

- **Source Line Of Code (SLOC)**: total number of lines in a source code file, including comments, blank lines, etc.

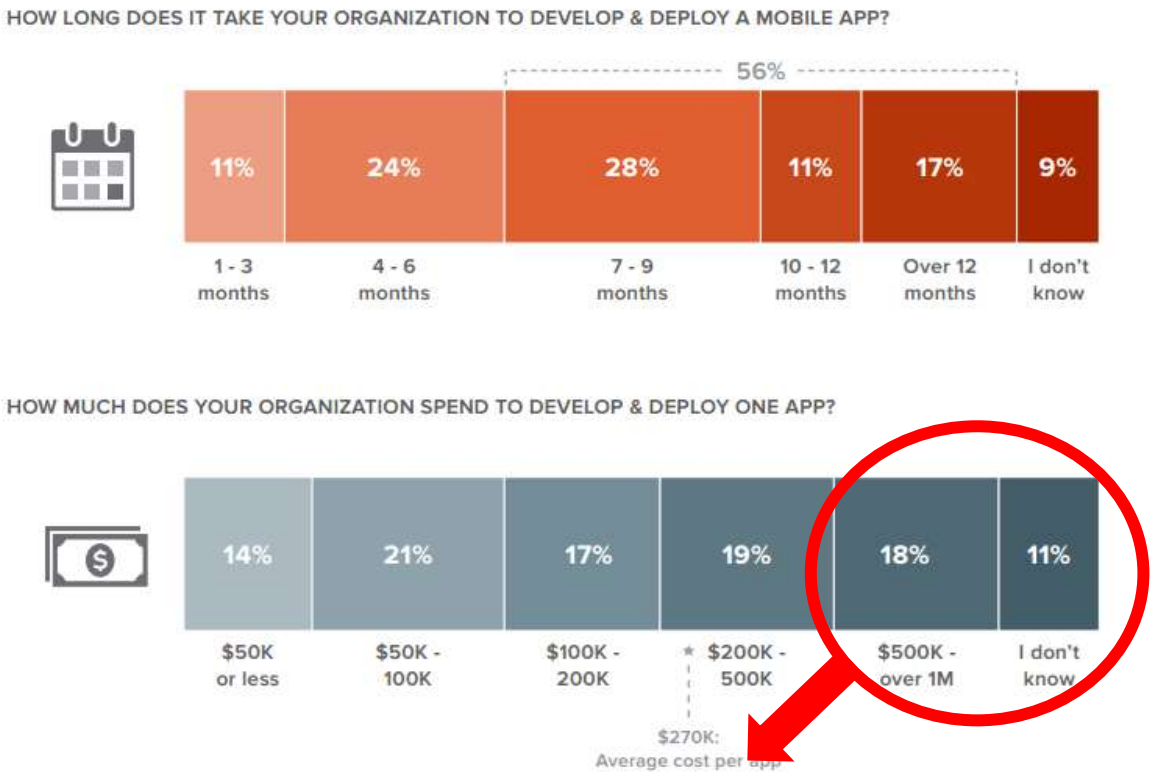
(2) Function-related measures: are based on an estimate of the functionality of the delivered software.

- **Function Points (FP)**: a way of measuring the size and extent of a software system by looking at which functions the system delivers to the user.

Cost challenge data:

- ~60% of code reuse.
- Five developers: one senior developer who is also a software designer, one software tester and three developers.
- Two of the three developers started five months ago and have less than 3 years of experience.
- The company is listed as CMMI level 3.
- Programming language: C or C++ as primary language for development.
- Software development follows agile development cycle with test based approach.
- Use of MySql database connected to application through TOMCAT server.
- Software complexity low software integration complexity high.

Some information from literature to build initial scale for cost:



Expected range

"Enterprise Mobility Survey 2014", available at: http://resources.kinvey.com/docs/State+of+Enterprise+Mobility+Survey+2014+-+Kinvey.pdf

Some information from literature to build initial scale for cost (other perspective):

Low range: $10,000 - $25,000

This is really the minimum it's ever going to cost. Think about an application that does one simple thing without bells and whistles (possibly to calculate something).

Medium range: $25,000 - $100,000

So for this you could possibly get an application that does a few cool things (possibly to record and track something(s)).

Average range: $100,000 - $250,000

Yes, those first two ranges were just to make you feel better. Chances are, your great idea is likely to fall into this price range. This is where you start to get simple reporting on straightforward features.

High range: $250,000 - $800,000

You are starting to get into full enterprise software applications. At this price point you'll get a good feature set and reporting, but still nothing too cutting edge in terms of functionality.

Unique range: $800,000 - $unlimited

Software applications in this range are usually unique and demand in-depth requirements analysis to understand and develop the necessary functionality. They may have a very rich feature set and/or complex calculations.

On-line report from Dr. John Flackett (Software Engineer), available at: https://goo.gl/VJFILP

Estimation using expert opinion and analogy techniques:

Employees hourly rate

Country/Region	Number of Employees	Average Hourly Rate (USD)
U.K.	50-249	$100-149
U.K.	10-49	$100-149
Eastern Europe	50-249	not listed
Eastern Europe	10-49	$50-99
Eastern Europe and U.S.*	50-249	$50-99
Eastern Europe and U.S.*	50-249	$25-49
U.S.	50-249	$100-149
U.S.	10-49	$150-199
U.S.	10-49	$100-149
India	50-249	<$25

Results from a survey made to 12 leading mobile application development companies to determine cost ranges of building an iPhone app and the key variables of cost

Available at: https://clutch.co/app-developers/resources/cost-build-mobile-app-survey

Estimation using expert opinion and analogy techniques:

Assumptions

Team Member	Expected hourly rate	Performance
A - Senior developer &designer	$100 - $ 150	100 SLOC/day
B – Tester	$ 50 - $ 100	
C - Developer 1	$ 25 - $ 100	80 SLOC/day
D - Developer 2	$ 25 - $ 100	80 SLOC/day
E - Developer 3	$ 25 - $ 100	80 SLOC/day

1 month = 4 weeks
1 week = 5 working days
1 working day = 8 working hours
» 1 month = 160 working hours

App Size: 80k SLOC (including removed code);

Constant Parameters:

- Team Size = 5
- Number of SLOC/day = 3*80 + 1*100 = 340 SLOC
- Number of days required for development (construction) = 80k/340 = 236 days
- Construction phase lengths = 236 / 20 = 12

Agile Software Development Cycle

Assumptions made based on information from: Ref 12; website: https://goo.gl/Kh26gb; and website: https://goo.gl/FpbFgH

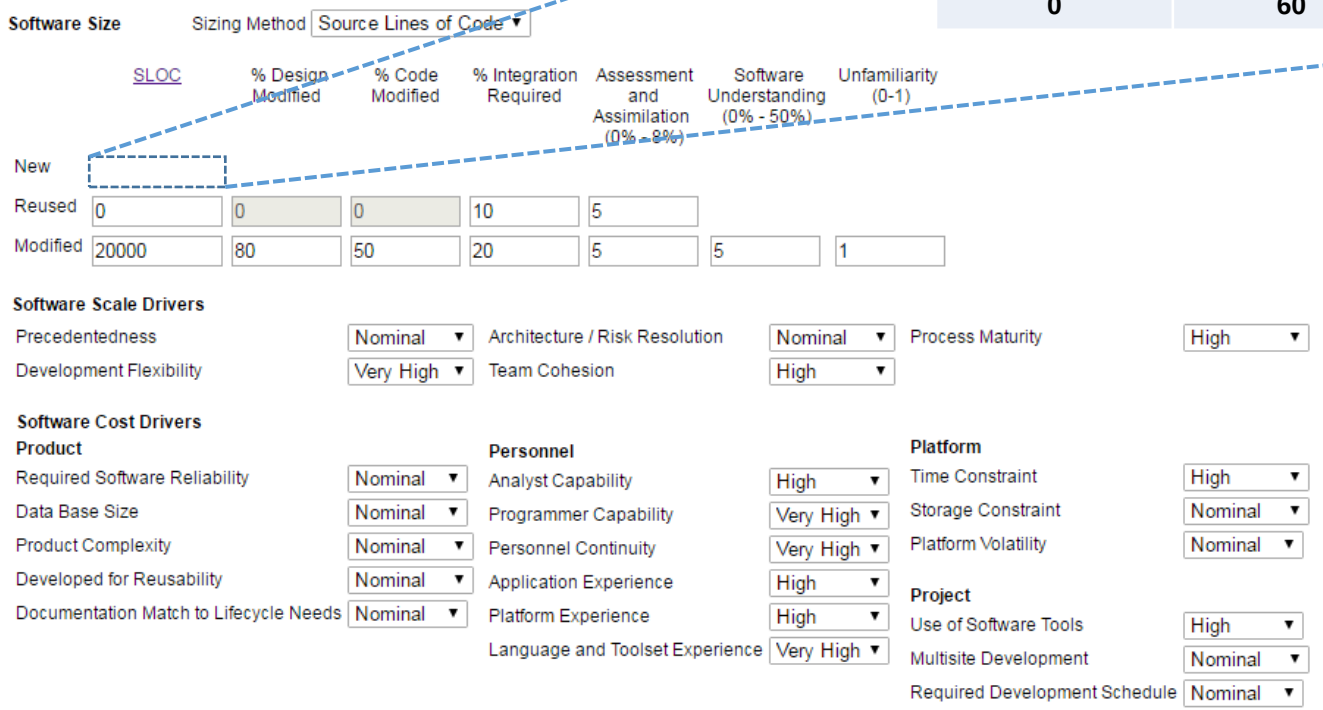
Estimates

	Minimum	Most Likely	Maximum
Concept	2*160*100 = 32 000	2*160*125 = 40 000	2*160*150 = 48 000
Inception	1*160*100 = 32 000	1*160*125 = 20 000	1*160*150 = 24 000
Construction	12*160*(100+3*25) = 336 000	12*160*(125+3*62.5) = 600 000	12*160*(150+3*100) = 864 000
Release	3*160*(100+50+3*25) = 108 000	3*160*(125+75+3*62.5) = 186 000	3*160*(150+100+3*100) = 264 000
Production	2*160*(100+50+3*25) = 72 000	2*160*(125+75+3*62.5) =124 000	2*160*(150+100+3*100) = 176 000
Retirement	1*160*(100+50+3*25) = 36 000	1*160*(125+75+3*62.5) = 62 000	1*160*(150+100+3*100) = 88 000
Total Cost	**$ 600 000**	**$ 1 032 000**	**$ 1 464 000**

Estimation using COCOMO-II

Assumptions/Inputs:

% of code reuse/recycling		
Worse scenario	**Most Likely**	**Best scenario**
0	**60**	**80**



Cost per person per month		
Minimum	**Most Likely**	**Maximum**
• [100+50+(25*3)]/5 = 45 • 45*160 = **$7200**	• [125+75+(62.5*3)]/5 = 77.5 • 77.5*160 = **$12400**	• [150+100+(100*3)]/5 = 110 • 110*160 = **$17600**

Estimation using COCOMO-II

Software Development (Elaboration and Construction)

Effort = 128.7 Person-months
Schedule = 18.2 Months
Cost = $926447

Total Equivalent Size = 97720 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	7.7	2.3	3.4	$55587
Elaboration	30.9	6.8	4.5	$222347
Construction	97.8	11.4	8.6	$704100
Transition	15.4	2.3	6.8	$111174

Minimum
$ 926 447

Software Development (Elaboration and Construction)

Effort = 126.9 Person-months
Schedule = 18.1 Months
Cost = $1573552

Total Equivalent Size = 96440 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	7.6	2.3	3.4	$94413
Elaboration	30.5	6.8	4.5	$377653
Construction	96.4	11.3	8.5	$1195900
Transition	15.2	2.3	6.7	$188826

Most Likely
$ 1 573 552

Software Development (Elaboration and Construction)

Effort = 121.6 Person-months
Schedule = 17.9 Months
Cost = $2139904

Total Equivalent Size = 92600 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	7.3	2.2	3.3	$128394
Elaboration	29.2	6.7	4.3	$513577
Construction	92.4	11.2	8.3	$1626328
Transition	14.6	2.2	6.5	$256789

Maximum
$ 2 139 904

Estimation using CATFOB

Assumptions/Inputs:

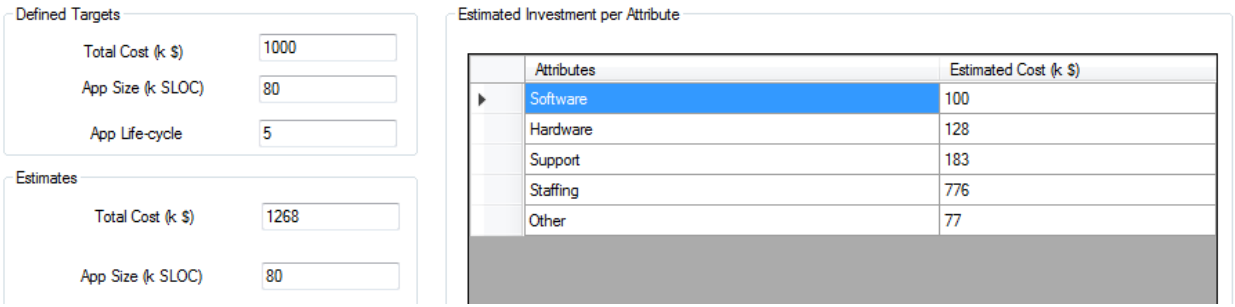
Project Database

Attributes	App_1	App_2	App_3	App_4	App_5	App_6
App size/complexity (k SLOC):	90	90	60	75	65	60
App Life-cycle:	5	6	5	5	5	6
	Investment (k$):	Investment (k$):	Investment (k$):	Investment (k$):	Investment (k$):	Investment (k$):
Software	140	130	90	100	90	88
Hardware	100	90	110	100	130	80
Support	250	300	100	150	80	80
Staffing	1500	1400	808	1100	1000	900
Other	60	50	20	35	15	20

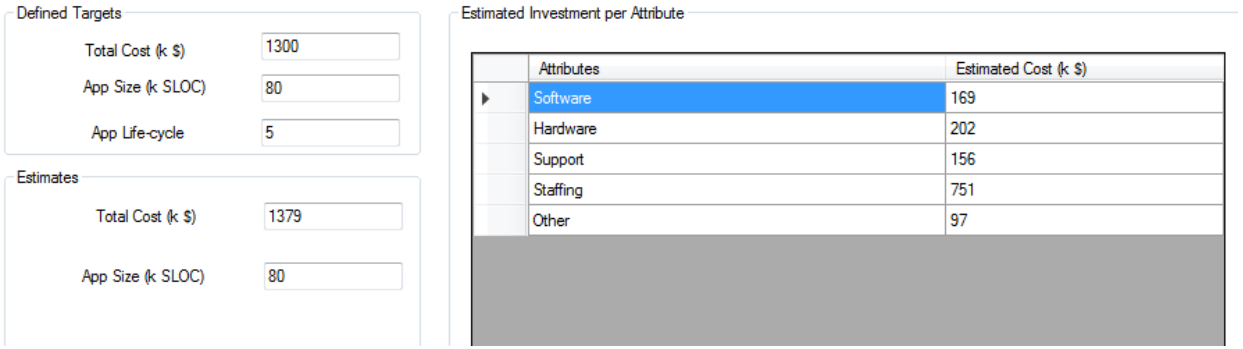
Attributes Cost Bounds

Attribute	Minimum Cost (k £)	Maximum Cost (k £)
Software	100	500
Hardware	100	500
Support	100	200
Staffing	600	1500
Other	10	100

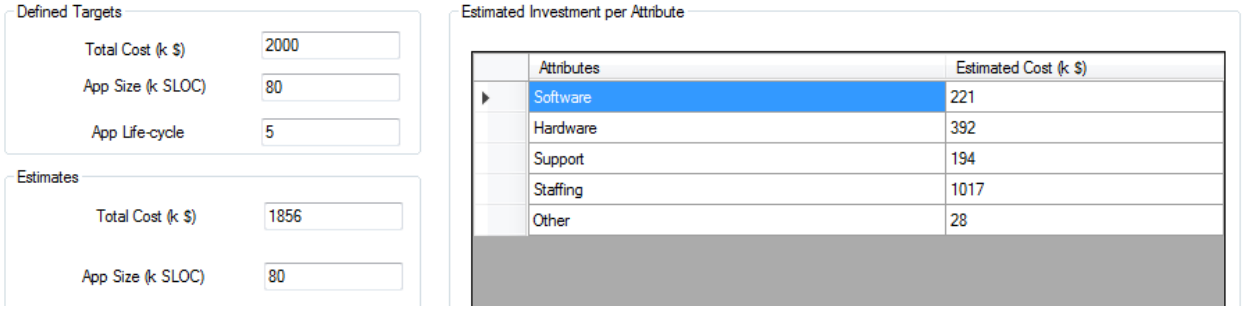
Estimation using CATFOB



Minimum
$ 1 268 000

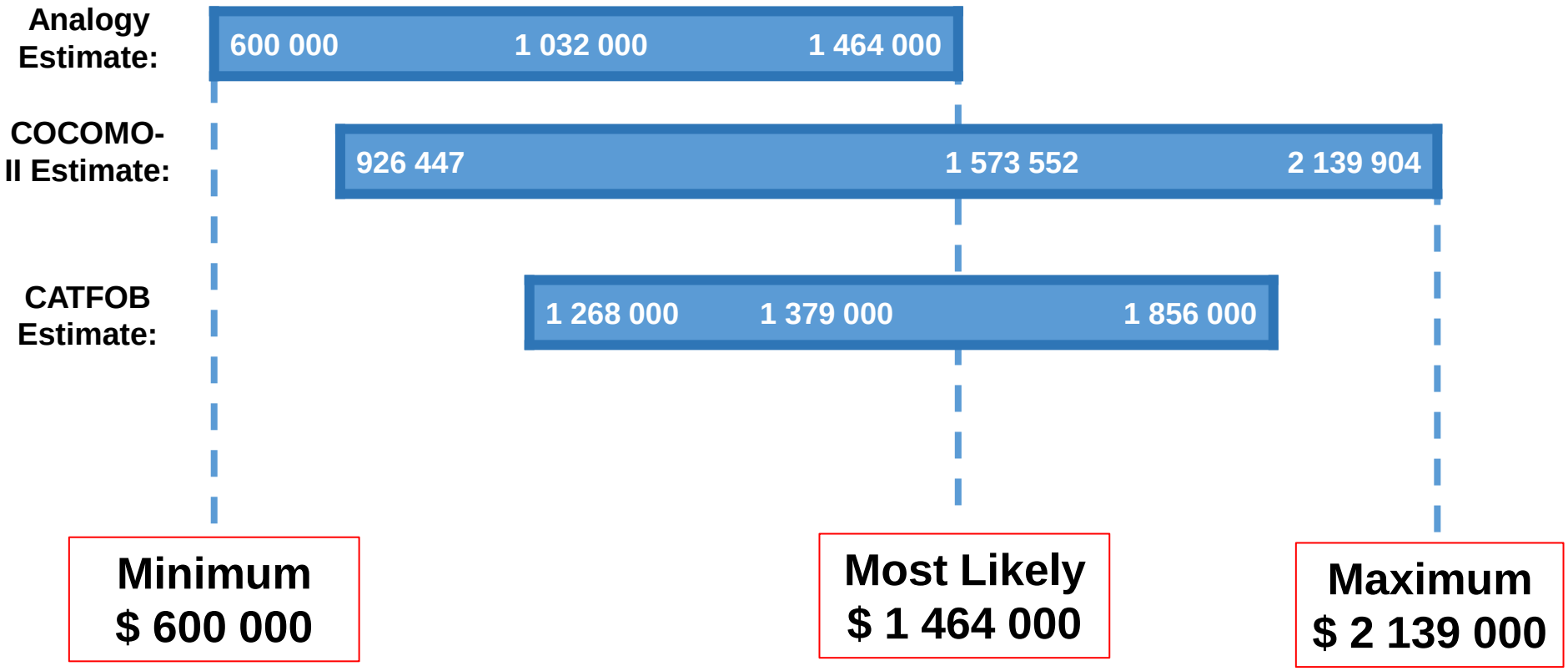


Most Likely
$ 1 379 000



Maximum
$ 1 856 000

Final Estimate



1) Understanding the research context
2) Identifying the attributes impacting on software development cost;
3) Identifying and assessing software cost estimation techniques;
4) Identifying and assessing software cost estimation models;
5) Building estimates;
6) Verifying and validating the results.

Verification

Internal verification of:

- Research objectives
- Research methodology
- Literature results
- Estimates

Validation

Person: Industry expert

Job: Cost engineer

Experience:10+ years

Industry sector: Multiple sectors

Main duties: Cost estimating, modelling, performance forecasting.

Lessons learnt

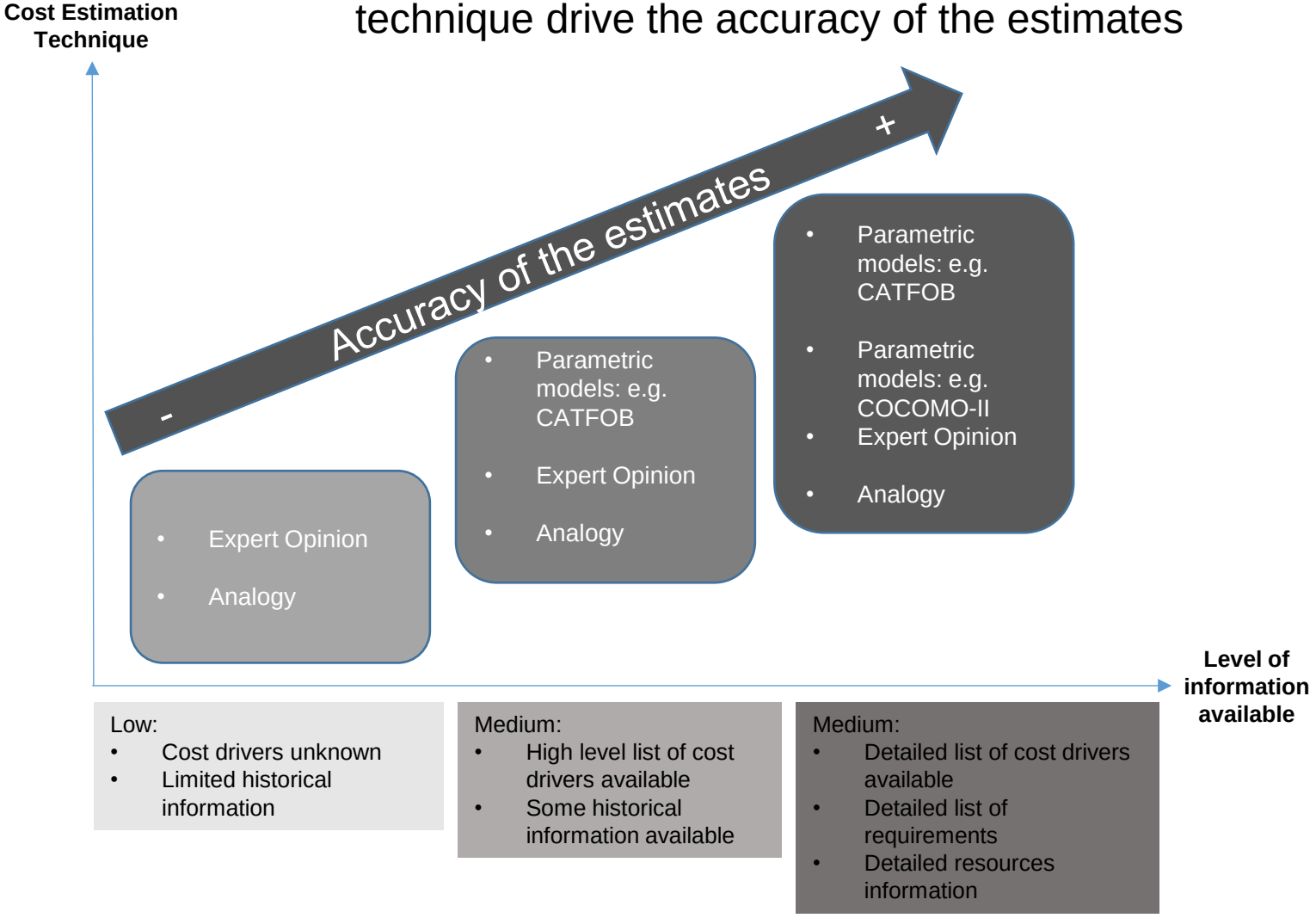
- There is no single/general list of attributes driving the cost of software development;

- Personnel is perhaps the biggest cost driver in software development;

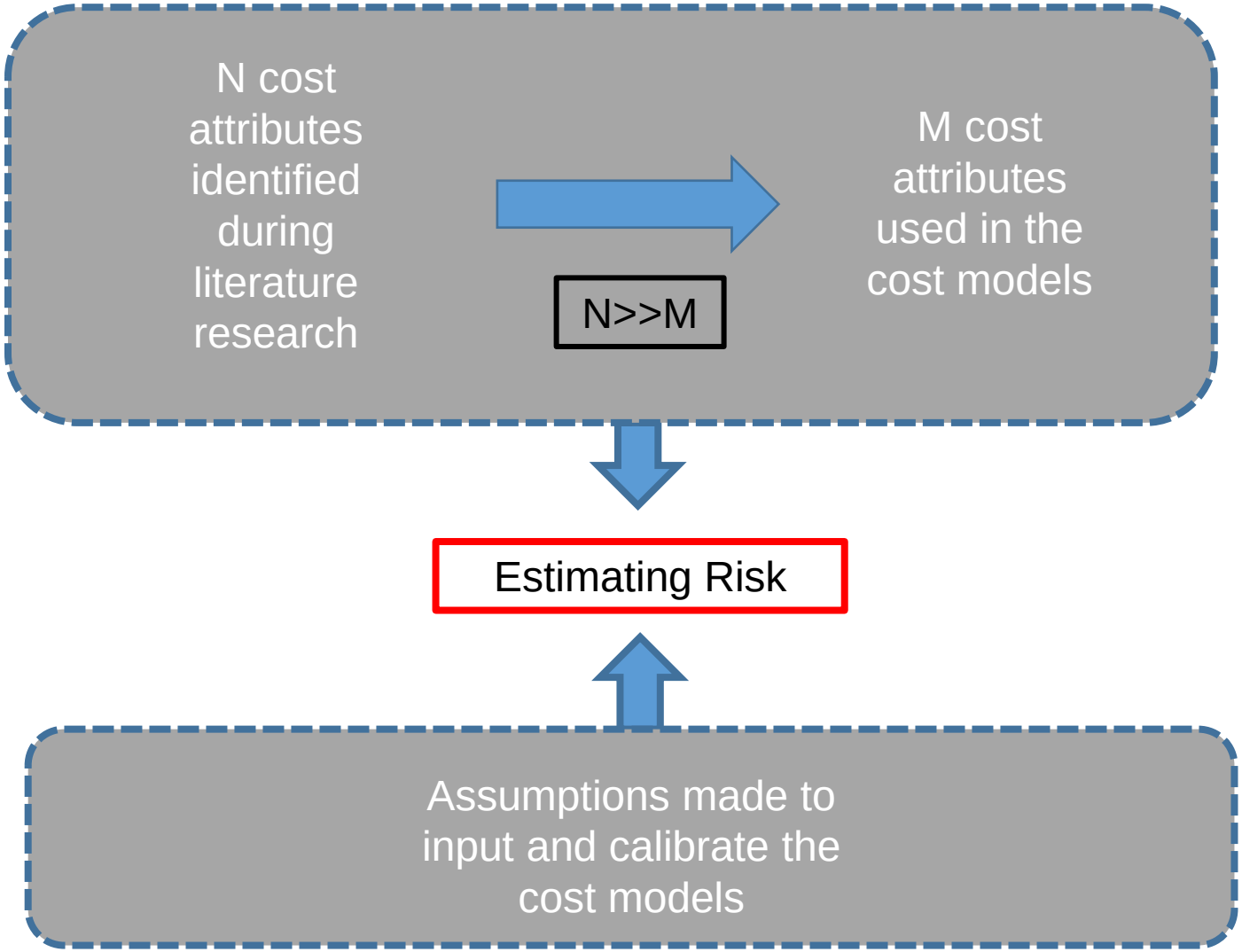
- The level of detailed information available about project requirements and resources has a big impact on the selection of the cost estimation technique(s) and drives the accuracy of the estimates.

Conclusions

Information availability and the selection of the appropriate technique drive the accuracy of the estimates



Limitations of the Results



References

[1] *Challenges in Automotive Software Engineering*, Manfred Broy (2006) - Proceedings of the 28th international conference on Software engineering

[2] *A Business Case Study of Open Source Software*, Carolyn A. Kenwood (2001) – Project Report

[3] *Frequently Forgotten Fundamental Facts about Software Engineering* (2001), Robert L. Glass – IEEE Software

[4] *Extreme Programming Evaluation Framework for Object-Oriented Languages – Version 1.1*, Laurie Williams, William Krebs, Lucas Layman (2004) - 26th IEEE International Conference on Software Engineering Toward an XP Evaluation Framework

[5] *Software cost estimation*, F J Heemstra (1992) - Information-and-Software-Technology. vol.34, no.10; Oct. 1992; p.627-39

[6] Software Development Cost Estimation Approaches – A Survey, Barry Boehm, Sunita Chulani (1998) – PhD report

[7] *Chapter 26: Software cost estimation* , Ian Sommerville (2004) - *Software Engineering Book, 7th edition*

[8] *Software Engineering Economics* (1983), Barry W. Boehm

[9] *COCOMO-II Model Definition Manual (2000),* Center for Software Engineering, USC

[10] *Handbook for Software Cost Estimation* (2003), Jairus Hihn, Frank Kuykendall – Jet Propulsion Laboratory, research report

[11] The Application of Function Points to predict source lines of code for software development (1992), Garlan S. Henderson – MsC Thesis

[12] *Measuring Productivity Using the Infamous Lines of Code Metric* (2008), Benedikt Mas y Pararela, Markus Pizka

Thank you for attending

Any Questions?

Email: d.rodrigues@cranfield.ac.uk
Phone: +44 (0)7928686095