Metrics and Estimation

Rahul Premraj + Andreas Zeller

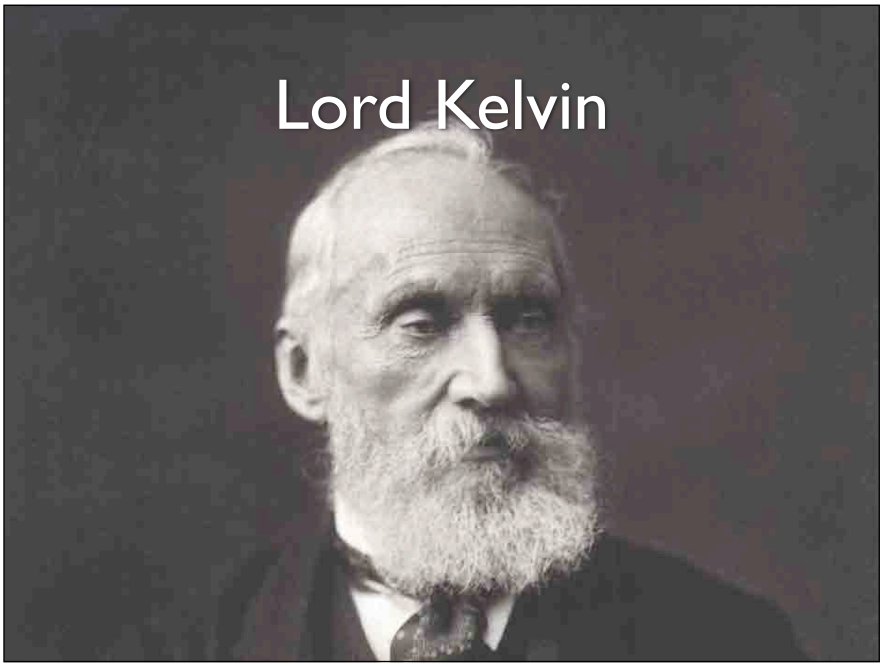
1

These slides are based on Pressman, Chapter 15 “Product Metrics”, Chapter 22 “Metrics for Process and Projects” and Chapter 23 “Estimation”

Metrics

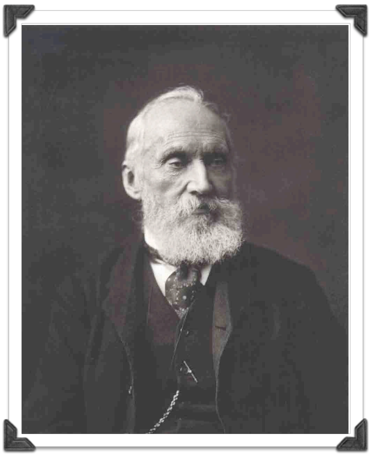
- Quantitative measures that provide *insight into the efficacy* of the process
- Analyzed and assessed by *software managers*
- Avoids basing judgements solely on subjective evaluation

2



3

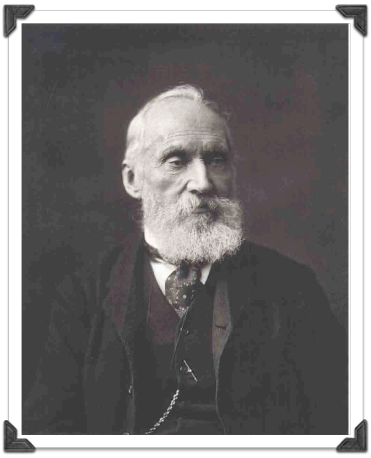
Lord Kelvin



When you can measure what you are speaking about and express it in numbers, you know something about it…

4

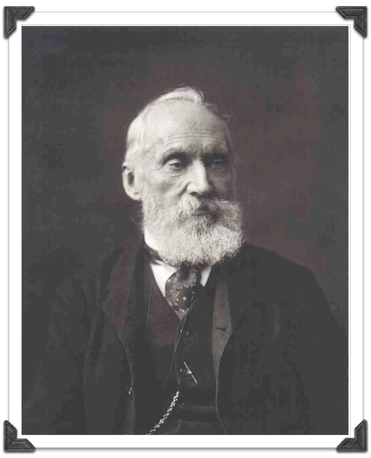
Lord Kelvin



…but when you cannot measure and express it in numbers, your knowledge is of a meager and unsatisfactory kind

5

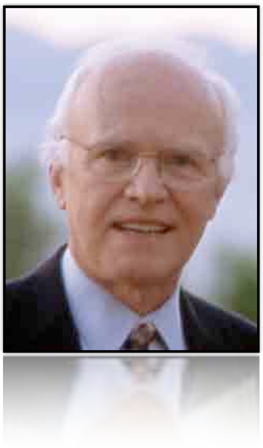
Lord Kelvin



…it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science.

6

Tom DeMarco



You can't control what you can't measure.

7

Measures and Metrics

- Measure
  Provides a quantitative indication of a product or process attribute
- Metric
  A quantitative measure of the degree to which a product or process possesses a given attribute
- Use *measures* to obtain *metrics*

8

Measurement Process

1. Formulation
   Deriving appropriate measures and metrics
2. Collection
   The mechanism used to accumulate required data
   (i.e., survey / observation / experimental study…)
3. Analysis
   Computation of metrics and application of mathematical tools
4. Interpretation
   Computation of metrics and application of mathematical tools
5. Feedback
   derived from interpretation; passed to software team

9

Goal / Question / Metric

1. Establish an explicit *measurement goal*
   that is *specific* to the activity or characteristic to be assessed – e.g.,
   - Are function and related data properly *compartmentalized*?
   - Is the *complexity* of each component within proper bounds?
2. Define a *set of questions*
   that must be answered in order to achieve the goal
3. Identify *well-formulated metrics*
   that help to answer these questions

The Metrics Landscape

Analysis | Design | Code

Analysis

Function Points

- Measure *functionality* delivered by a system
- Proposed by Albrecht in the 80s.
- ISO recognised software metric to measure software size.
- Based on functionality as perceived by the user and independent of the technology.

Function Points

- *Input:* A set of related inputs as one input.
- *Output:* A set of related outputs as one output.
- *Inquiries:* Each user query type is counted.
- *Files:* Files are logically related data, i.e., data structures or physical files.
- *Interface:* Data transfer to other systems.

Function Points

Information Domain Value	Count	Weighting factor	FPs
External Inputs	3	3 / 4 / 6	9
External Outputs	2	4 / 5 / 7	8
External Inquiries	2	3 / 4 / 6	6
Internal Logical Files	1	7 / 10 / 15	10
External Interface Files	2	5 / 7 / 10	10
Count total			53

Value Adjustment

To compute function points (FP), use

$$FP = count\ total \times (0.65 + 0.01 \times \sum(F_i))$$

- F_i : one of 15 *value adjustment factors* [0–5]
  1. Does the system require backup?
  2. Are specialized data communications needed?
  3. Are there distributed processing functions?
  4. Is performance critical? – etc.
- All constants determined empirically

Function Points

- Proponents claim:
  - FP is language independent.
  - Size can be derived from problem description.
- Opponents claim:
  - it's subjective: different people arrive at different estimates.

16

Function Points

- Counting FP itself takes a long time.
- Difficult to count consistently without extensive training.
- Difficulty of using automated tools.
- Difficulty of counting embedded, highly algorithmic modules, and web systems.

17

Analysis

FP Associations

Netherlands Software Metrics Users Association

Nesma

International Function Point Users' Group

IFPUG

Finnish Software Measurement Association

FiSMA

and others...

18

The Metrics Landscape

Analysis | Design | Code

19

Design

Design Metrics

- Weighted Methods per Class
- Depth of Inheritance Tree
- Number of Children
- Coupling between Object Classes
- Response for a Class
- Lack of Cohesion in Methods

So-called Chidamber and Kemerer (CK) Metrics – most frequently referenced

20

The Metrics Landscape

Analysis | Design | Code

21

Code

Lines of Code

- Simplest and most widely used metric
- Comments and blank lines usually left out.
- Numerous tools available that do this for you :-)
- Don't forget to check how accurate they are first!

Disadvantages of LOC

- Size varies with coding style.
- Focuses on coding activity alone.
- Correlates poorly with quality and efficiency of code.
- Penalises higher level programming languages, reuse, etc.

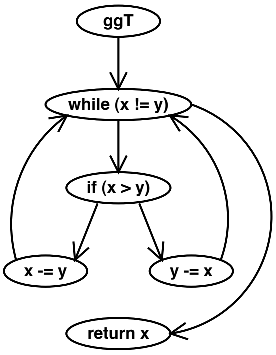
Disadvantages of LOC

- Measures lexical/textual complexity only.
- Difficult to estimate LOC from problem description.
- Does not address issues of structural and logical complexity.

Code

Code Metrics

- Halstead Metrics
  measure *size of vocabulary*
- McCabe Metrics
  measure *complexity of control flow* as
  P(V) = *edges* – *statements* + 2 × *entry points*
- No longer appropriate for modern OO systems
  ⇒ use OO metrics for complexity



So-called Chidamber and Kemerer (CK) Metrics – most frequently referenced

25

The Metrics Landscape

Analysis

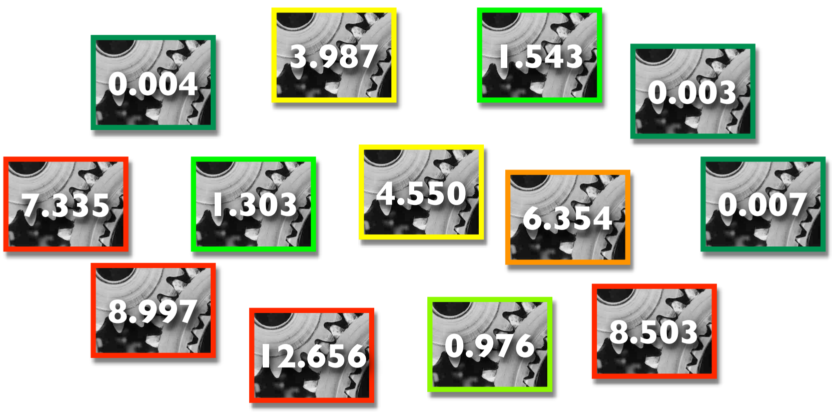
Design

Code

Testing
(next week)

26

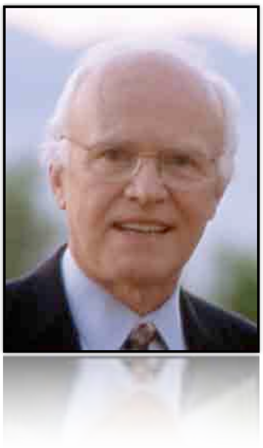
All these Metrics!



Once we have obtained all these metrics, what do we do with them?

27

Tom DeMarco



You can't control what you can't measure.

Now that we talked about measurement, let's talk about control.

28

Estimation

- During project management, one needs to *estimate* resources, cost, and schedule
- Crucial for project success
- As much an *art* as it is *science*…
- …but need not be conducted in a haphazard manner!

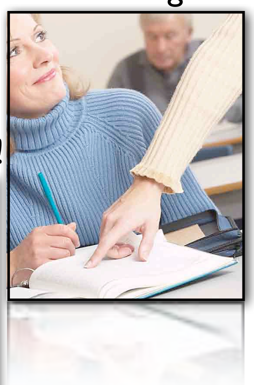
29

What costs are incurred?

Staff

Travel for serious work!

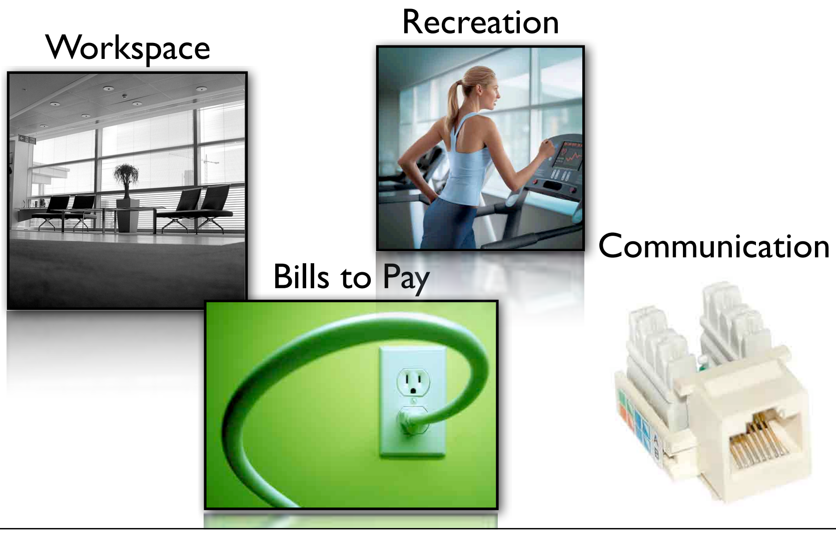
Training



Hardware

30

What costs are incurred?



31

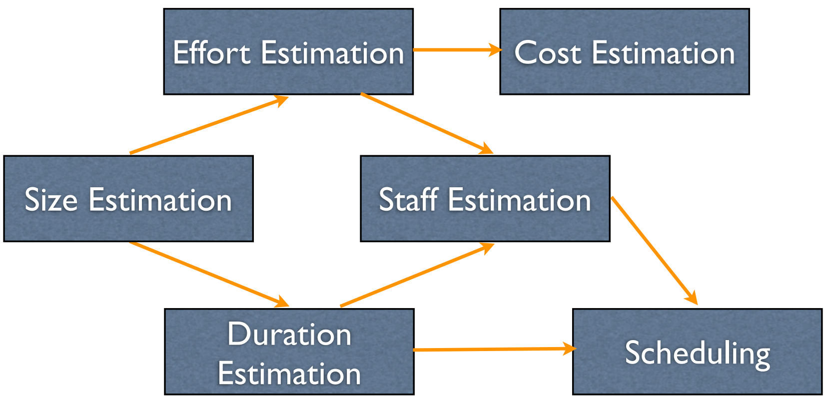
(c) Ian Sommerville

Why bother to Estimate?

- To establish a budget for a software project.
- To provide a means of controlling project costs.
- To monitor progress against that budget by comparing planned with estimated costs.
- To establish a cost database for future estimation.
- Cost estimation and planning/scheduling are closely related activities.

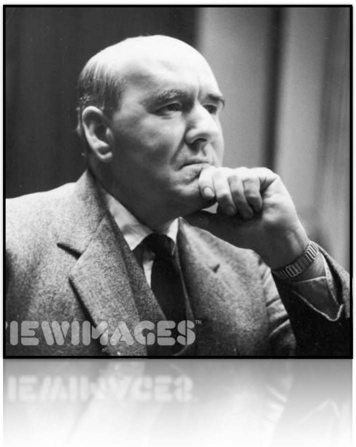
32

Why bother to Estimate?



33

Parkinson's Law



Work expands so as to fill the time available for its completion.

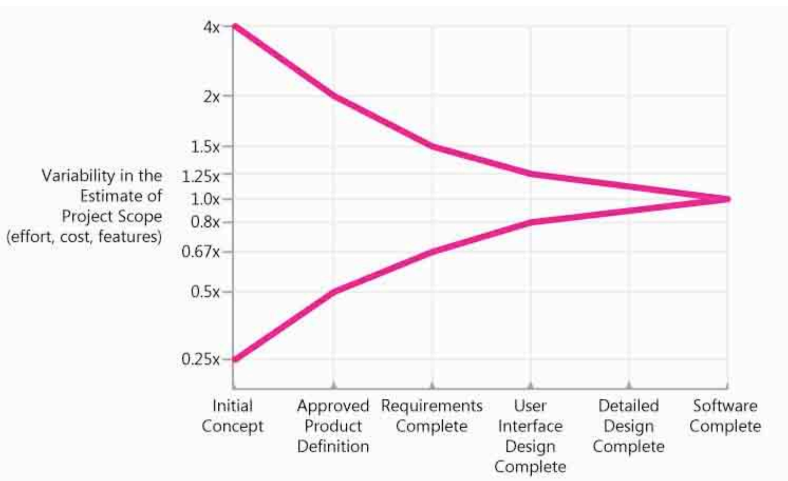
34

Why try to be accurate?

- Under-estimation
  - Loss of money
  - Damage to company's reputation
- Over-estimation
  - Loss of bids to competitors.
  - Opportunity cost of not doing other projects.
  - Wastage of ideal or untilised resources.

35

Cone of Uncertainty



36

Metrics Needed

- Metrics need to be defined.
- They need to be collected.
- Then, validated.
- And lastly, if deemed suitable, used to base estimates upon.

37

Cost Estimation Techniques

- Expert Judgement
- Algorithmic Techniques
- Case-Based Prediction

38

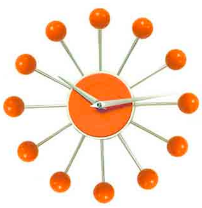
Emphasis on Size!

Boehm's third law

Development effort is a function of product size.

Less	Less	Less
More	More	More

=



=

39

Size measures

Function Points

Information Domain Value	Count	Weighting factor	FPs
External Inputs	3	3 / 4 / 6	9
External Outputs	2	4 / 5 / 7	8
External Inquiries	2	3 / 4 / 6	[illegible]
Internal Logical Files	1	7 / 10 / 15	[illegible]
External Interface Files	2	5 / 7 / 10	[illegible]
Count total			[illegible]

Design Metrics

- Weighted Methods per Class
- Depth of Inheritance Tree
- Number of Children
- Coupling between Object Classes
- Response for a Class
- Lack of Cohesion in Methods

Lines of Code

- Simplest and most widely used metric
- Comments and blank lines usually left out.
- Numerous tools available that do this for you :-)
- Don't forget to check how accurate they are first!

40

Various ways to measure size

Criteria for Size Measure

- Relationship to development effort
- Precision
- Machine countable
- Suitable for early planning

41

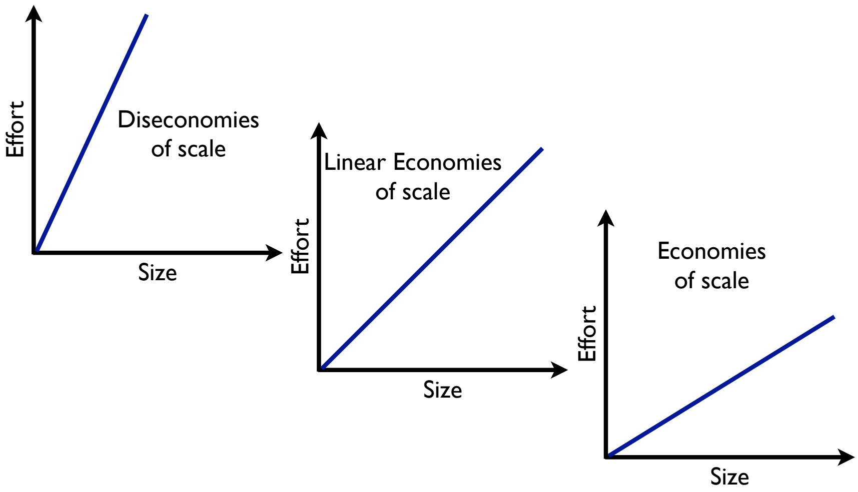
Estimate from FPs

- Compute function points for new project
- Multiply with *organizational average productivity* (e.g., 6.5 PM / person month)
- Obtain effort estimate

Information Domain Value	Count	Weighting factor	FPs
External Inputs	3	3 / 4 / 6	9
External Outputs	2	4 / 5 / 7	8
External Inquiries	2	3 / 4 / 6	6
Internal Logical Files	1	7 / 10 / 15	10
External Interface Files	2	5 / 7 / 10	10
Count total			53

42

Size vs. Effort Relationship



43

Note that all three examples are linear… but the 'diseconomies' of scale differ. If the distinction is unclear, please refer to a simple explanation at Wikipedia – http://en.wikipedia.org/wiki/Diseconomies_of_scale

Expert Judgement

44

Expert Judgement

- You approach an in-house or external expert
  independent consultants or big consultancies such as Accenture, Cap Gemini
- You tell them what needs to be developed.
- He makes an estimate and gives you a number!

  Simple… eh?

45

Expert Judgement

How do they arrive at an estimate?

Activity Decomposition

System Decomposition

46

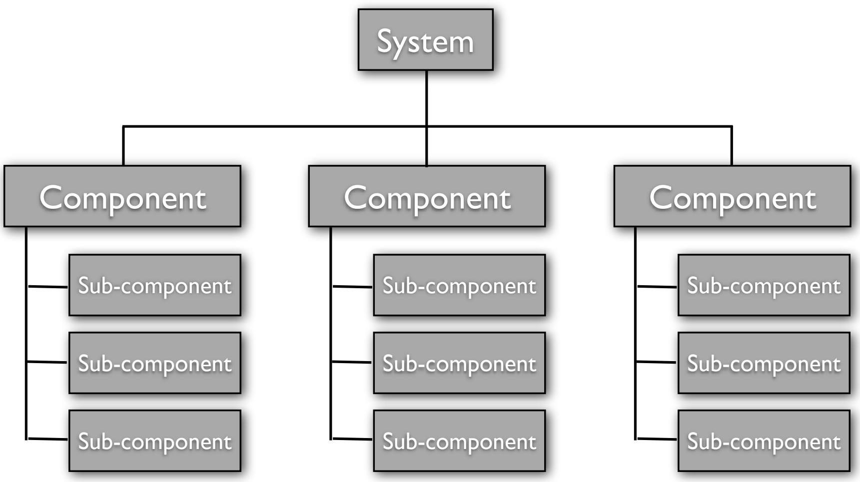
Activity Decomposition

	Months															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
Requirements	4	4	6	8	6	4	1	0	0	0	0	0	0	0	0	33
Scenario Development	3	3	3	2	1											
Data Model Development	1	1	1	1												
Sequence Diagrams			2	2	2	2										
Detailed Requirements				3	3	2	1									
Design	0	1	3	4	5	4	3	1	0	0	0	0	0	0	0	21
High Level Architecture		1	1	1	1											
Performance Model					1	1	1									
Data Base Design			1	1	1	1										
Detailed Architecture			1	2	2	2	2	1								
Coding & Unit Testing	0	0	0	3	8	14	14	14	13	12	9	8	7	2	2	106
Module Design & Coding				3	6	9	9	9	7	5	3	3	3	1	1	
Data Base Implementation					2	2	2	2	2	1						
OA&M Implementation						3	3	3	2	1	1					
Unit Testing									2	5	5	5	4	1	1	
System Testing	0	0	4	4	4	4	3	3	6	6	6	5	5	5	3	58
Test Case Development			3	3	3	2										
Test Environment			1	1	1	2	3	3	1	1	1					
Test Case Execution									5	5	5	5	5	5	3	
Project Mgmt & Administration	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	26
Training & Documentation				2	2	2	4	4	2	2	1	1	1	1	1	23
Total Project Staffing	5	6	14	22	27	30	27	24	23	22	18	16	15	10	8	267
Total Project Cost (staff years)																22.3

From *Software Measurement and Estimation: A Practical Approach* by Linda Laird and Carol Brennan

47

System Decomposition



48

Expert Judgement

Advantages

- Approachable
- Only resort when historical and quantitative data is absent.
- Can understand your project and development environment.
- Lots of experience (to reuse)
- Can customise solutions
- Can tackle exceptions (e.g., different technology used)

Expert Judgement

Disadvantages

Cost

Expert Judgement

Disadvantages

...an estimate is only as good as the expert's opinion, and there is no way usually to test that opinion until it is too late to correct the damage if that opinion proves wrong.

Expert Judgement

Disadvantages

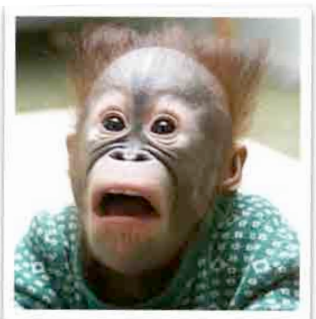
Humans are prone to bias, optimism, pessimism, instinct, and desire to win!

52

Expert Judgement

Disadvantages

Justification



... often, can't explain or don't want to share!

53

Expert Judgement

Disadvantages

Different Opinions

$1 million

Today: $2 million

54

Expert Judgement

Disadvantages

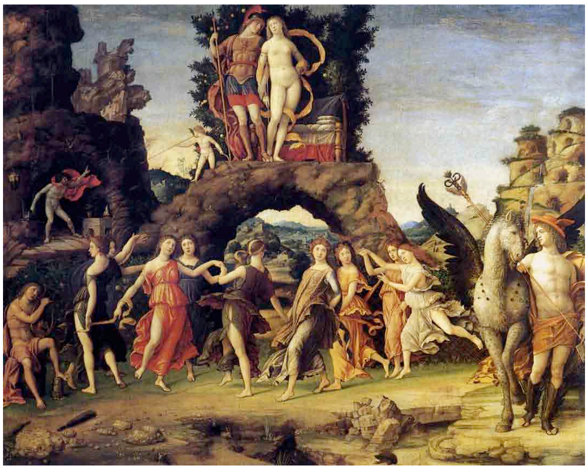
Repeatability

Today: $2 million

10 days later: $4 million

55

Delphi Method



56

Delphi Method

- You take your project to a group of experts.
- Each expert comes up with a list of tasks and estimates.
- Each experts' list is anonymously distributed amongst others.
- Lists are discussed and refined to make a single big list.
- Several rounds (usually four) take place to further refine the list and arrive at a final estimate.

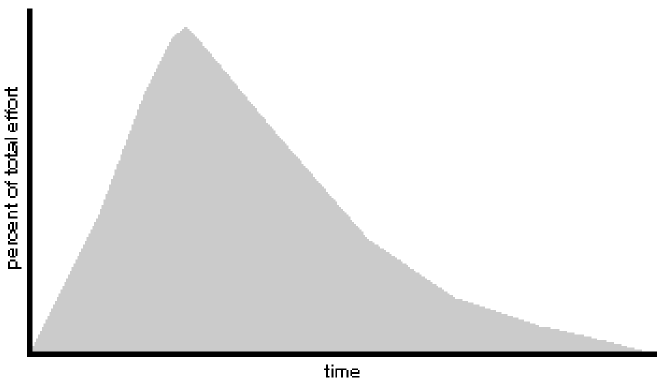
57

Algorithmic Models

58

Rayleigh's Curve

Putnam-Norden-Rayleigh (PNR) Curve



59

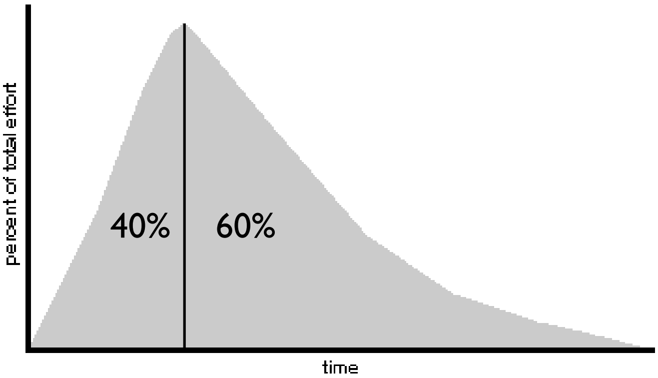
Rayleigh Curve

- Rayleigh curve represents the number of full-time personnel required at any time.
- At project start, small number of engineers needed.
- As project progresses, more work is required that needs more staff. At one point, the number of staff *peaks* (system testing and product release).
- It then drops down (installation and delivery).

60

Rayleigh's Curve

Putnam-Norden-Rayleigh (PNR) Curve



61

http://en.wikipedia.org/wiki/Software_equation

Software Engineering Equation

derived from observing over 4,000 projects

$$E = \left[\frac{\text{LOC} \times B^{0.333}}{P}\right]\left(\frac{1}{t^4}\right)$$

E = Project effort in person months or person years
LOC = Lines of code estimate for the project
t = Length of project measured in months or years
B = a "special skills factor
P = a "Productivity Parameter

62

Different values of LOC will result in different values of effort.

Wait! What about these?

1 KLOC = 1,000 LOC

Walston-Felix Model
$E = 5.2 \times KLOC^{0.91}$

Bailey-Basili Model
$E = 5.2 + 0.73 \times KLOC^{1.61}$

Boehm simple Model
$E = 3.2 \times KLOC^{1.05}$

Doty Model for KLOC > 9
$E = 5.288 \times KLOC^{1.047}$

and more...

63

Maintainability Index

Size of vocabulary

McCabe complexity

$$\text{Maintainability} = 171 - 5.2\ln(\overline{V}) - 0.23\overline{V(G)} - 16.2\ln(\overline{L}) + 50\sin\left(\sqrt{2.4\overline{C}}\right)$$

code lines

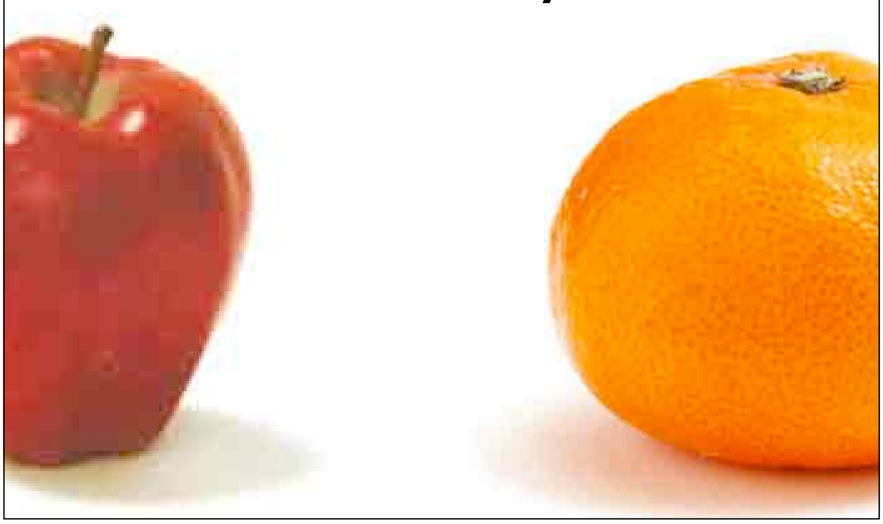
Percentage of comment lines

Oman, P. & Hagemeister, J. "Constructing and Testing of Polynomials Predicting Software Maintainability." Journal of Systems and Software 24, 3 (März 1994): 251–266.

64

And one can come up with very very elaborated curve fitting models…

Diversity



65

… but if something works at HP, will it work for you, too? The problem is: Software engineering data is usually very diverse...

Diversity

66

projects originate from different countries! (what are the implications of this?)

Diversity

Infosys
POWERED BY INTELLECT
DRIVEN BY VALUES
ORACLE
SAP
IBM
Google
hp
invent
Microsoft

67

Projects originate from different companies!
These companies are poor examples because they will almost never share their data. However, the point to learn is that their products and processes limit transfer to techniques and knowledge between each other.

Diversity

programming language domain process model
experience staff requirements
customers

68

and there are many more sources for diversity

69

so check your data carefully before using it!

Model Calibration

Cross-specific models

Company-specific models

70

The best way to use these models is to *calibrate* them on earlier products (from other or the same companies).

Company Specific Models (four studies)

Cross Company Models (four studies)

No Trend (two studies)

Barbara Kitchenham

2

Emilia Mendes

1
2

Katrina Maxwell

1
1

Isabella Wieczorek

3
2

Lionel Briand

2

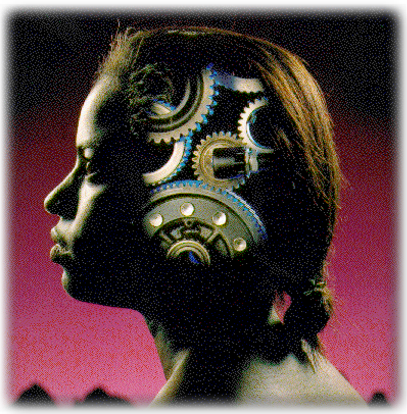
Martin Shepperd

1

71

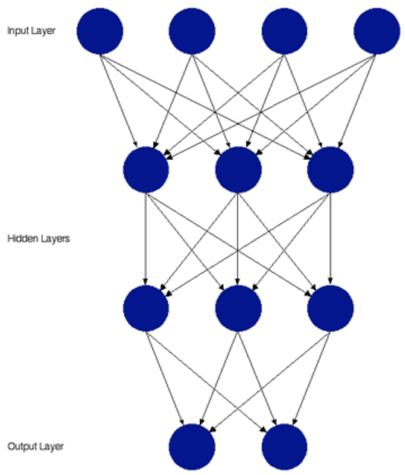
It generally turns out that models that are calibrated on earlier projects are much more effective.

Machine Learning Approaches



72

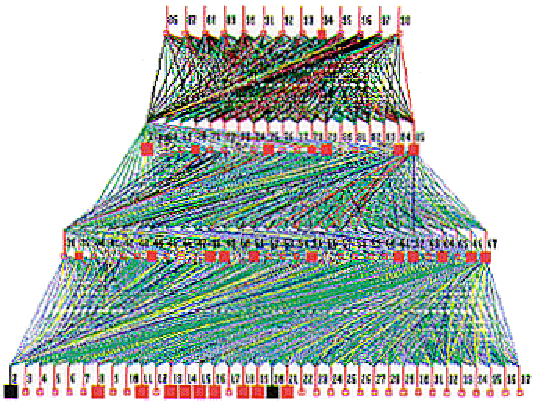
Neural Networks



You *do not* need to learn neural networks for exams.

73

Neural Networks



You *do not* need to learn neural networks for exams.

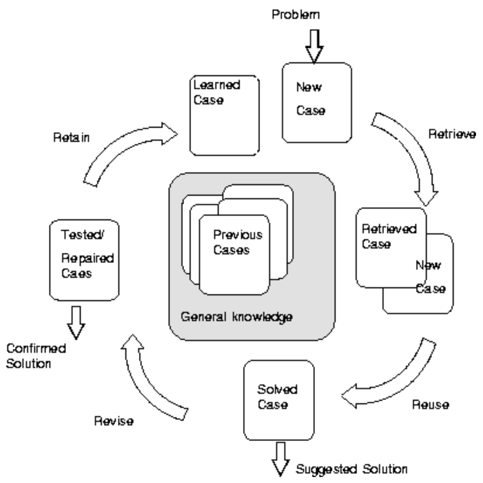
74

Neural Networks

- Neural Networks (NN) are relatively good predictors of effort.
- But they function like a black-box.
- Extremely time-consuming and difficult to train and optimize.
- Users need to have substantial training themselves.

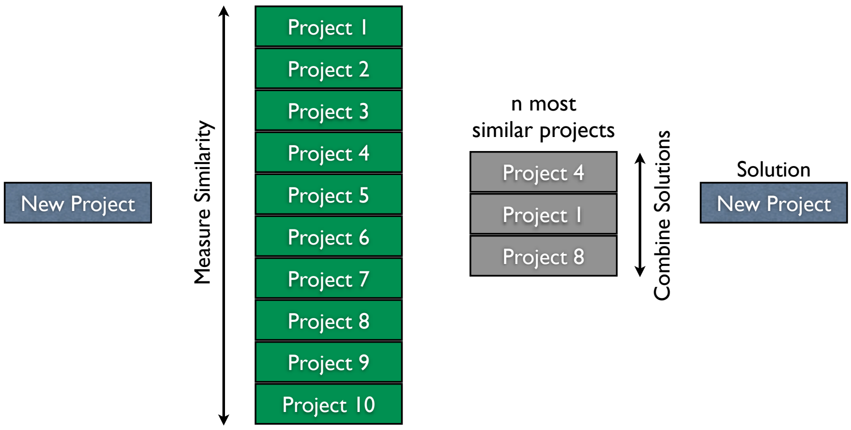
75

Case-Based Prediction



76

Case-Based Prediction



77

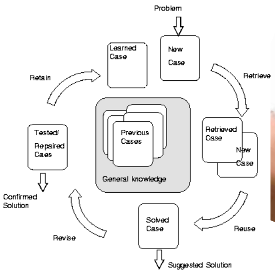
Case-Based Prediction

Advantages

- Very favourable for weak-theory domains.
- Less knowledge engineering required.
- Can be shared across a multitude of users and corporations.
- Possible to give justification for proposed solutions.
- Constant and potentially automated learning!

78

Experts and Analogy



79

De-Marco-Glass Law

Most cost estimates tend to be low.

80

Estimation Accuracy

On the whole, we are still no good at this task.
Sometimes, we estimate and get lucky :-)
Residuals range to thousands of hours.

But *some* estimate is better than *no estimate*!

81

Summary

- Software Engineering Estimation is crucial for the project's success.
- There are several models proposed.
- None seem to perform well consistently.
- Calibrated models can be very helpful.
- An estimate having *some confidence* is still better than *no estimate at all!*