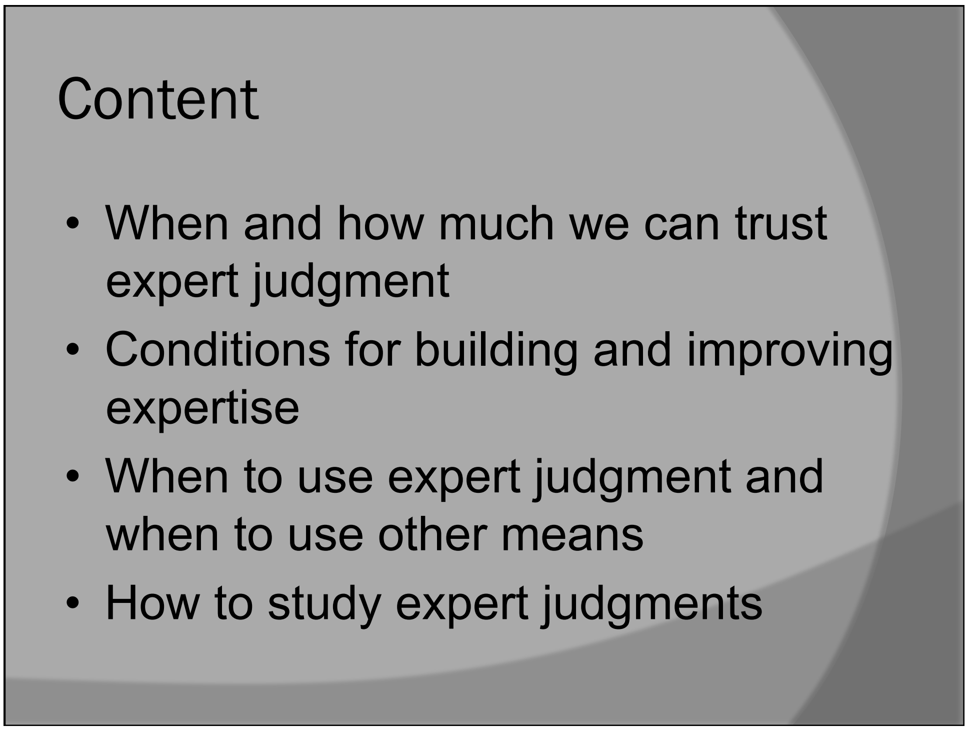
Judgment and decision-making in software engineering

When are the 'experts' experts? How can we know?

Magne Jørgensen
Simula Research Laboratory



Content

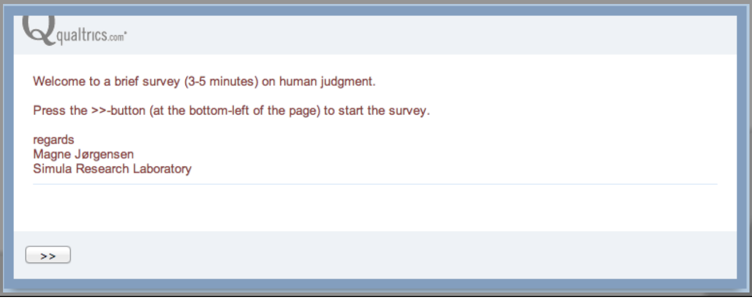
- When and how much we can trust expert judgment
- Conditions for building and improving expertise
- When to use expert judgment and when to use other means
- How to study expert judgments

BUT FIRST!

All of you who have a smartphone or a laptop (or can borrow one), please answer four questions (3-5 minutes) at:

goo.gl/LBaLa

The answers will be summarized as part of my keynote.



Welcome to a brief survey (3-5 minutes) on human judgment.

Press the >>-button (at the bottom-left of the page) to start the survey.

regards
Magne Jørgensen
Simula Research Laboratory

Worse Than Chance? Performance and Confidence Among Professionals and Laypeople in the Stock Market

Clouds make nerds look good: field evidence of the impact of incidental factors on decision making

Uri Simonsohn*

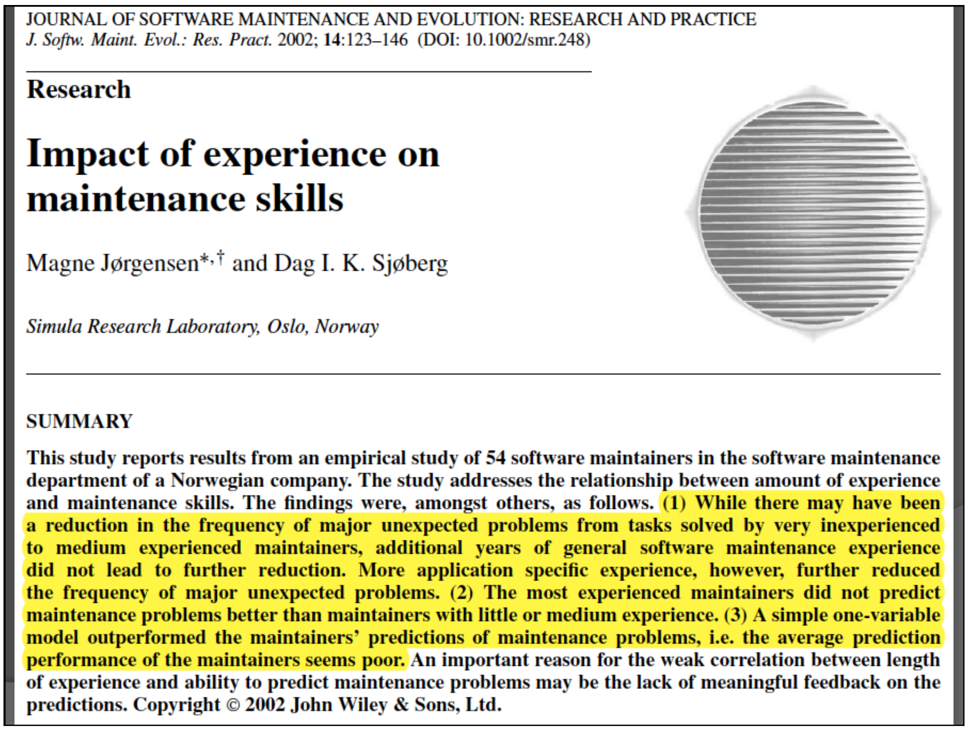
Article first published online: 9 OCT 2006

DOI: 10.1002/bdm.545

Issue

Journal of Behavioral Decision Making

Volume 20, Issue 2, pages 143–152, April 2007



JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE
J. Softw. Maint. Evol.: Res. Pract. 2002; **14**:123–146 (DOI: 10.1002/smr.248)

Research

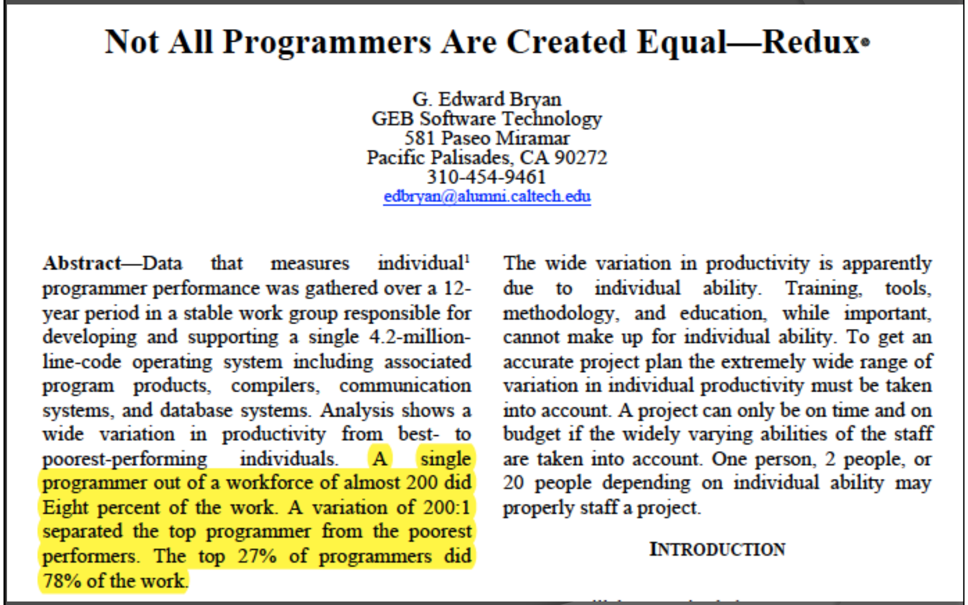
Impact of experience on maintenance skills

Magne Jørgensen*,† and Dag I. K. Sjøberg

Simula Research Laboratory, Oslo, Norway

SUMMARY

This study reports results from an empirical study of 54 software maintainers in the software maintenance department of a Norwegian company. The study addresses the relationship between amount of experience and maintenance skills. The findings were, amongst others, as follows. (1) While there may have been a reduction in the frequency of major unexpected problems from tasks solved by very inexperienced to medium experienced maintainers, additional years of general software maintenance experience did not lead to further reduction. More application specific experience, however, further reduced the frequency of major unexpected problems. (2) The most experienced maintainers did not predict maintenance problems better than maintainers with little or medium experience. (3) A simple one-variable model outperformed the maintainers' predictions of maintenance problems, i.e. the average prediction performance of the maintainers seems poor. An important reason for the weak correlation between length of experience and ability to predict maintenance problems may be the lack of meaningful feedback on the predictions. Copyright © 2002 John Wiley & Sons, Ltd.



Not All Programmers Are Created Equal—Redux•

G. Edward Bryan
GEB Software Technology
581 Paseo Miramar
Pacific Palisades, CA 90272
310-454-9461
edbryan@alumni.caltech.edu

Abstract—Data that measures individual[1] programmer performance was gathered over a 12-year period in a stable work group responsible for developing and supporting a single 4.2-million-line-code operating system including associated program products, compilers, communication systems, and database systems. Analysis shows a wide variation in productivity from best- to poorest-performing individuals. A single programmer out of a workforce of almost 200 did Eight percent of the work. A variation of 200:1 separated the top programmer from the poorest performers. The top 27% of programmers did 78% of the work. The wide variation in productivity is apparently due to individual ability. Training, tools, methodology, and education, while important, cannot make up for individual ability. To get an accurate project plan the extremely wide range of variation in individual productivity must be taken into account. A project can only be on time and on budget if the widely varying abilities of the staff are taken into account. One person, 2 people, or 20 people depending on individual ability may properly staff a project.

INTRODUCTION

Domains where we find and do not find good expert performance

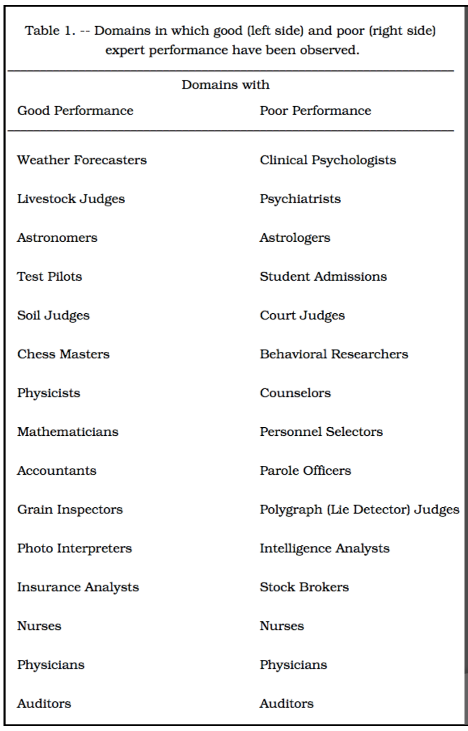
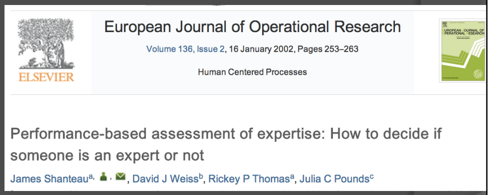


Table 1. -- Domains in which good (left side) and poor (right side) expert performance have been observed.

Domains with	
Good Performance	Poor Performance
Weather Forecasters	Clinical Psychologists
Livestock Judges	Psychiatrists
Astronomers	Astrologers
Test Pilots	Student Admissions
Soil Judges	Court Judges
Chess Masters	Behavioral Researchers
Physicists	Counselors
Mathematicians	Personnel Selectors
Accountants	Parole Officers
Grain Inspectors	Polygraph (Lie Detector) Judges
Photo Interpreters	Intelligence Analysts
Insurance Analysts	Stock Brokers
Nurses	Nurses
Physicians	Physicians
Auditors	Auditors



European Journal of Operational Research
Volume 136, Issue 2, 16 January 2002, Pages 253–263
Human Centered Processes

Performance-based assessment of expertise: How to decide if someone is an expert or not

James Shanteau[a], David J Weiss[b], Rickey P Thomas[a], Julia C Pounds[c]

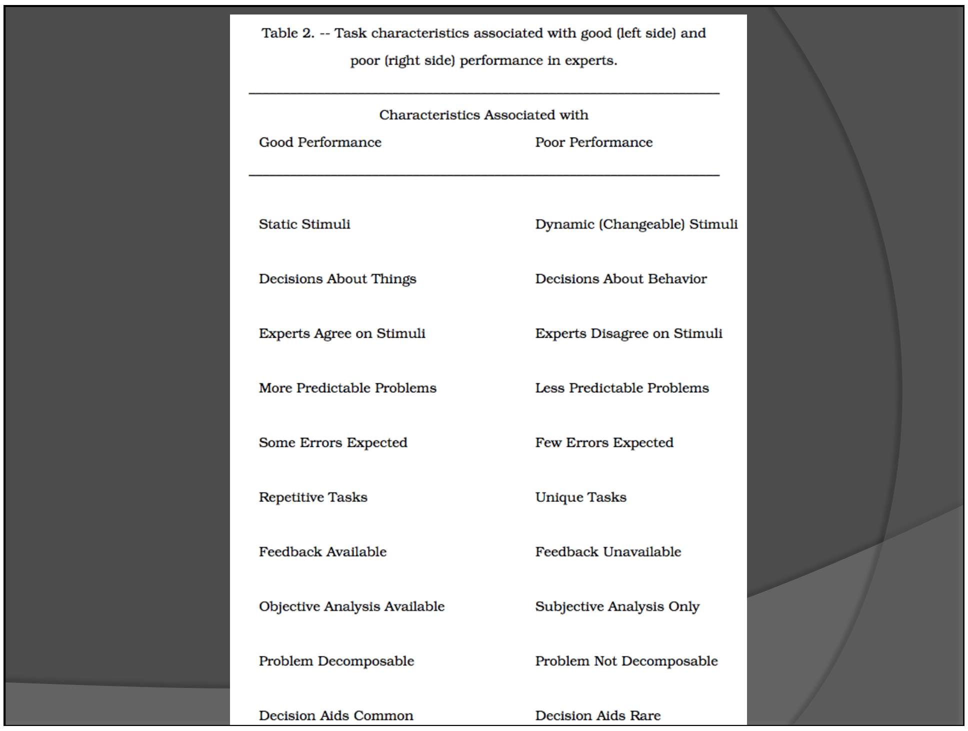
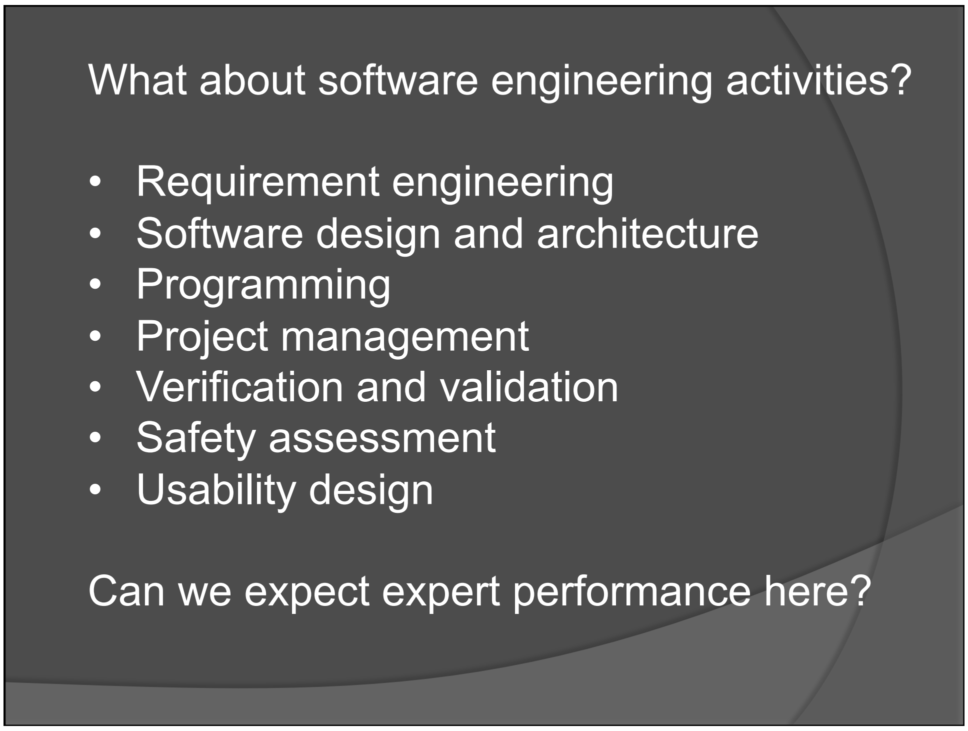


Table 2. -- Task characteristics associated with good (left side) and poor (right side) performance in experts.

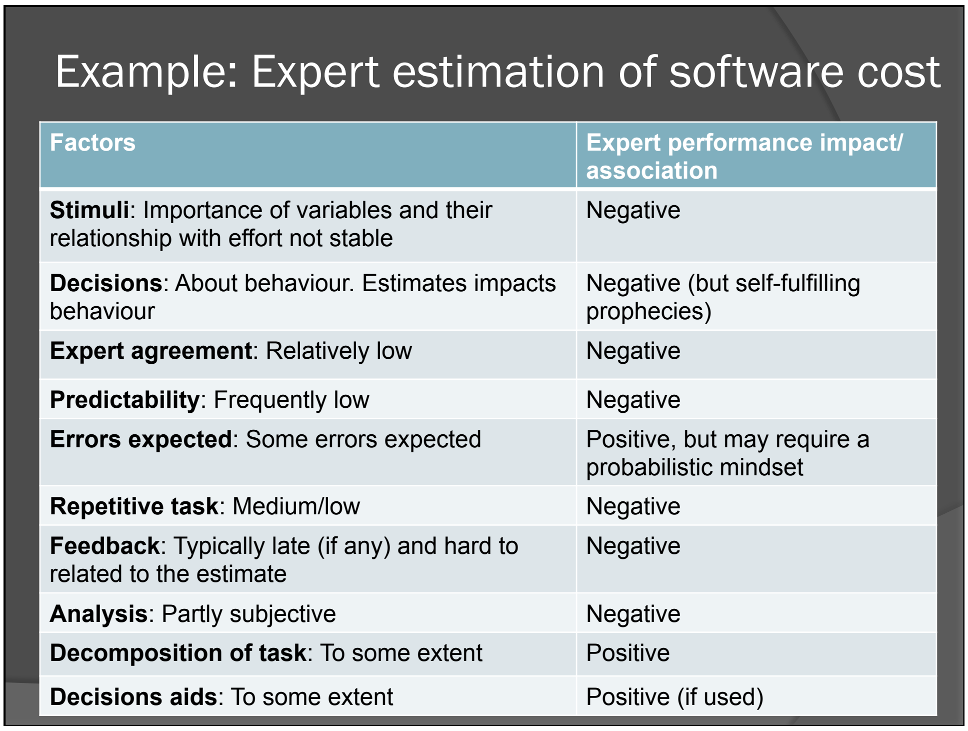
Characteristics Associated with	
Good Performance	Poor Performance
Static Stimuli	Dynamic (Changeable) Stimuli
Decisions About Things	Decisions About Behavior
Experts Agree on Stimuli	Experts Disagree on Stimuli
More Predictable Problems	Less Predictable Problems
Some Errors Expected	Few Errors Expected
Repetitive Tasks	Unique Tasks
Feedback Available	Feedback Unavailable
Objective Analysis Available	Subjective Analysis Only
Problem Decomposable	Problem Not Decomposable
Decision Aids Common	Decision Aids Rare



What about software engineering activities?

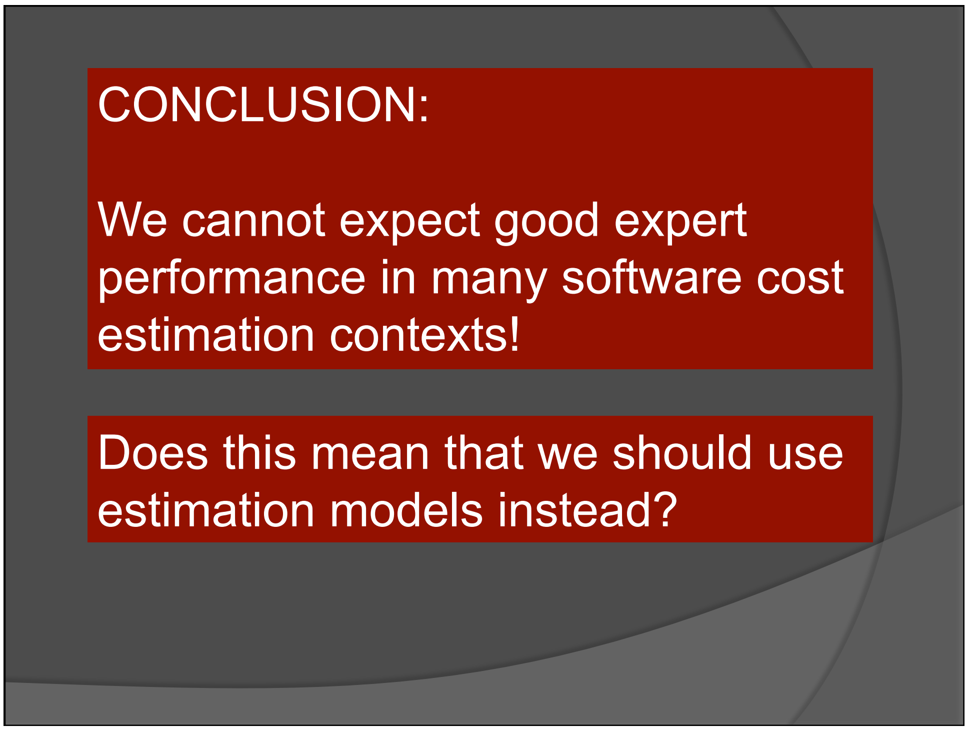
- Requirement engineering
- Software design and architecture
- Programming
- Project management
- Verification and validation
- Safety assessment
- Usability design

Can we expect expert performance here?



Example: Expert estimation of software cost

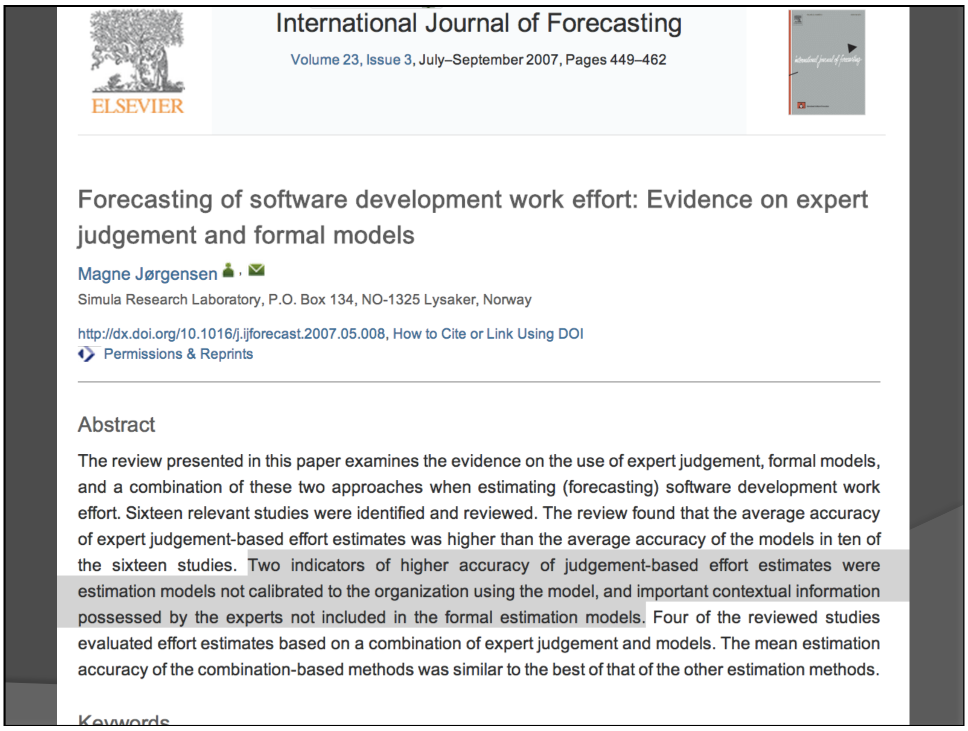
Factors	Expert performance impact/ association
Stimuli: Importance of variables and their relationship with effort not stable	Negative
Decisions: About behaviour. Estimates impacts behaviour	Negative (but self-fulfilling prophecies)
Expert agreement: Relatively low	Negative
Predictability: Frequently low	Negative
Errors expected: Some errors expected	Positive, but may require a probabilistic mindset
Repetitive task: Medium/low	Negative
Feedback: Typically late (if any) and hard to related to the estimate	Negative
Analysis: Partly subjective	Negative
Decomposition of task: To some extent	Positive
Decisions aids: To some extent	Positive (if used)



CONCLUSION:

We cannot expect good expert performance in many software cost estimation contexts!

Does this mean that we should use estimation models instead?



International Journal of Forecasting

Volume 23, Issue 3, July–September 2007, Pages 449–462

Forecasting of software development work effort: Evidence on expert judgement and formal models

Magne Jørgensen

Simula Research Laboratory, P.O. Box 134, NO-1325 Lysaker, Norway

http://dx.doi.org/10.1016/j.ijforecast.2007.05.008, How to Cite or Link Using DOI

Permissions & Reprints

Abstract

The review presented in this paper examines the evidence on the use of expert judgement, formal models, and a combination of these two approaches when estimating (forecasting) software development work effort. Sixteen relevant studies were identified and reviewed. The review found that the average accuracy of expert judgement-based effort estimates was higher than the average accuracy of the models in ten of the sixteen studies. Two indicators of higher accuracy of judgement-based effort estimates were estimation models not calibrated to the organization using the model, and important contextual information possessed by the experts not included in the formal estimation models. Four of the reviewed studies evaluated effort estimates based on a combination of expert judgement and models. The mean estimation accuracy of the combination-based methods was similar to the best of that of the other estimation methods.

Keywords

Models and experts have different strengths and weaknesses

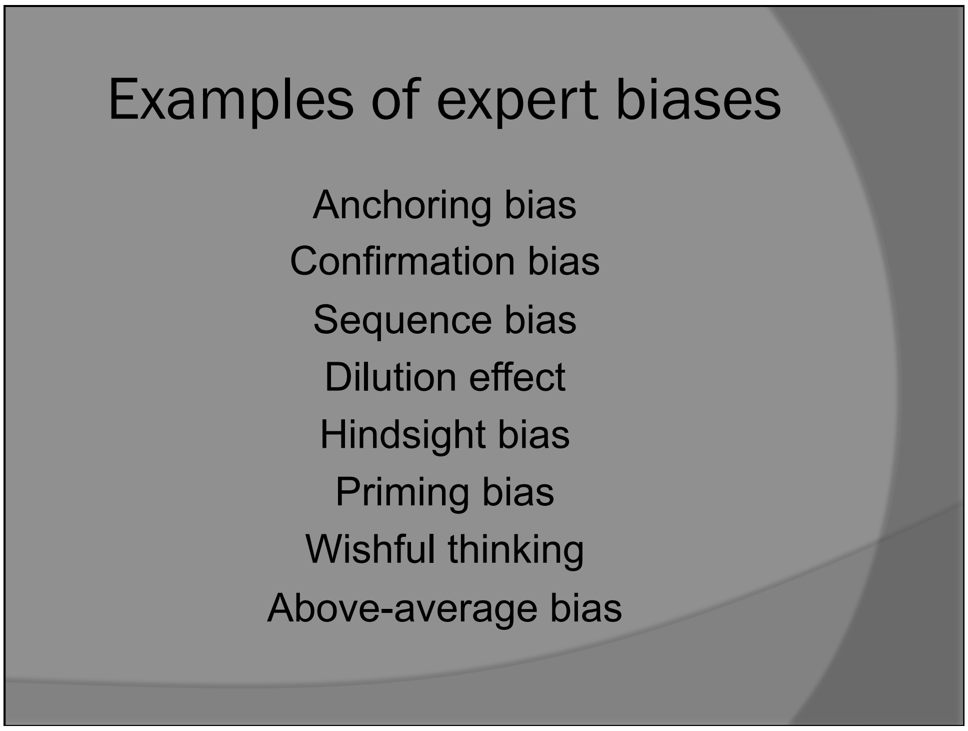
Models give better opportunities to understand and improve the processes. Expert judgment is to a large extent unconscious.

We tend to believe more in outcome of expert judgment, in spite of believing less in the process.

Model output is typically more consistent.
Experts can be incredible inconsistent (both between experts and within same expert)

Expert judgment is typically more “flexible”, “effort-less” and can include very specific information. Models are typically based on stable relationships and a stable set of important indicators.

Easy to mislead (bias) the experts.



Examples of expert biases

Anchoring bias
Confirmation bias
Sequence bias
Dilution effect
Hindsight bias
Priming bias
Wishful thinking
Above-average bias

Let us have a look at your answers on the four questions.

A simple report can be found at:

goo.gl/wgfqqN

Anchoring bias - study

Companies instructed to support us with estimates for five projects: “The independent effort estimates will be applied to evaluate the realism of other software providers’ effort estimates of the projects.”

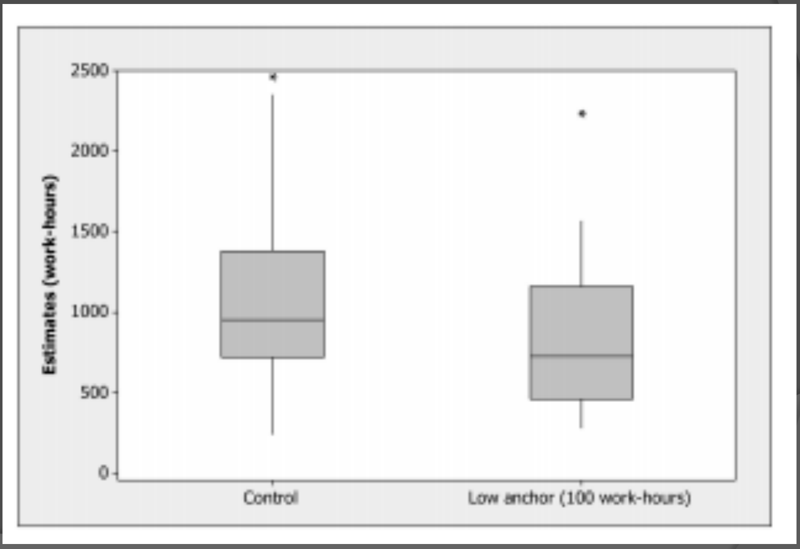
46 offshoring companies accepted the paid work and estimated the required effort of the same five projects

Project 4: “The preliminary budget of the new system is $10 000 [corresponding to about 100 work-hours with typical pricing in the country in which it will be built]. The preliminary budget is not built on any knowledge about the actual cost of developing the new system, and will, if needed, be extended to cover the expenses necessary to build a quality system with the desired functionality.”

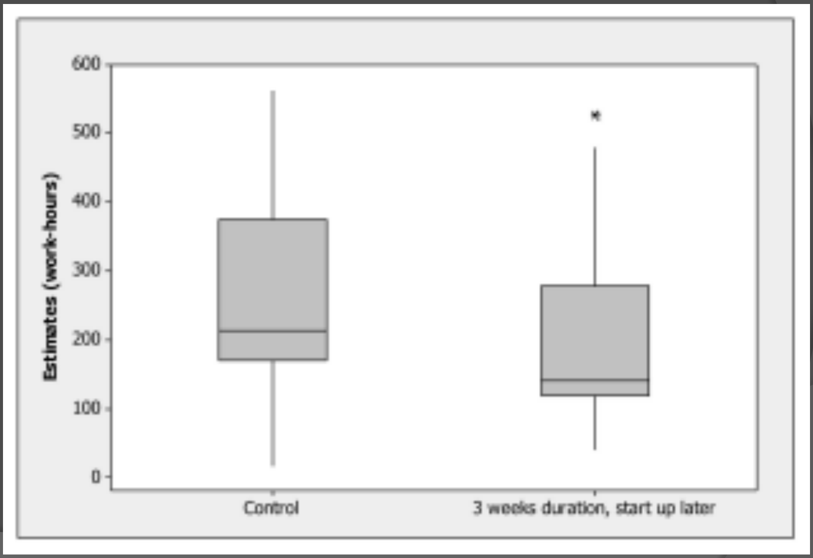
Project 5: “[the client] expects that the system development starts February 3, 2008 and can be launched on February 23, 2008. This three-week period should include all development and testing.”

M. Jørgensen and S. Grimstad. The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment, IEEE Transactions on Software Engineering, 37(5):695-707, 2011.

Anchoring in preliminary (low) budget

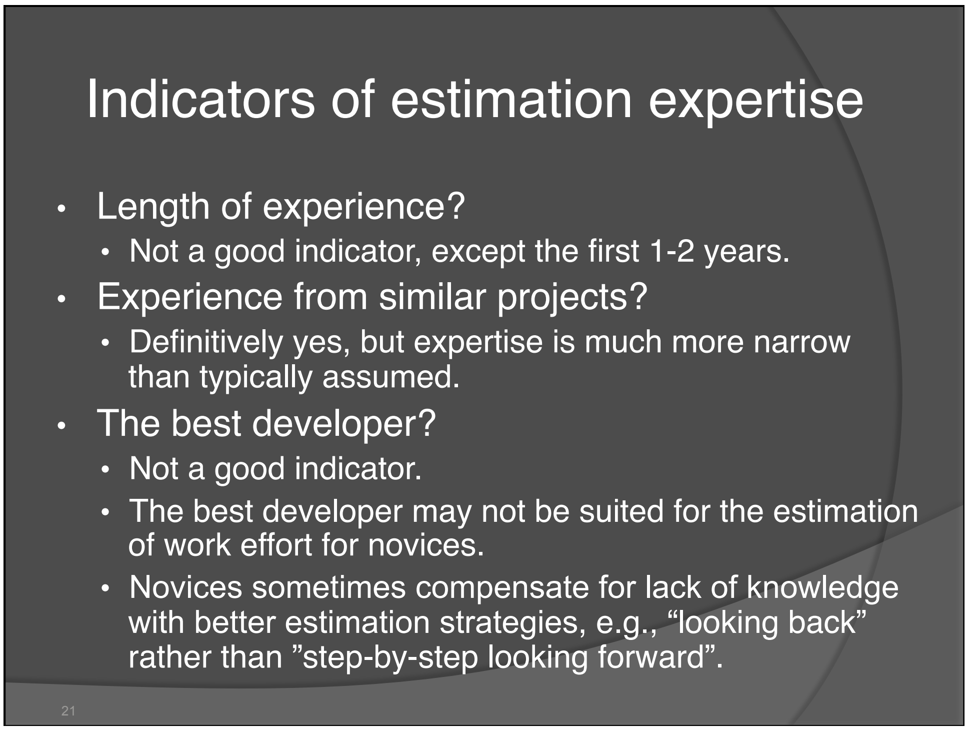


Anchoring in short time period



Example: Indicators of expertise

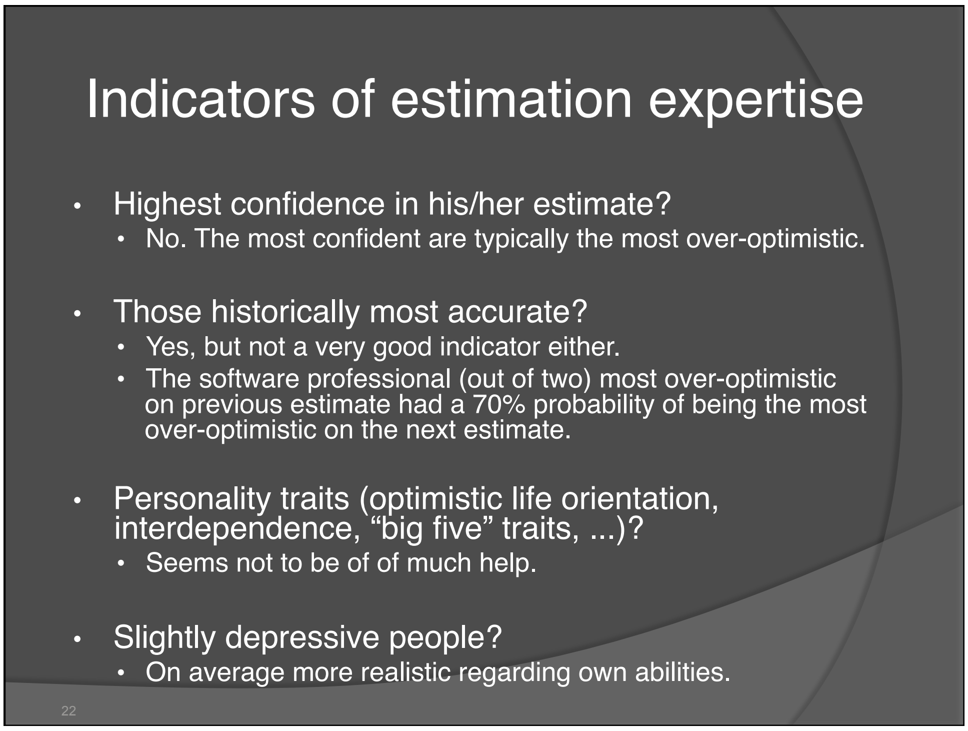
Estimation of software development effort



Indicators of estimation expertise

- Length of experience?
  - Not a good indicator, except the first 1-2 years.
- Experience from similar projects?
  - Definitively yes, but expertise is much more narrow than typically assumed.
- The best developer?
  - Not a good indicator.
  - The best developer may not be suited for the estimation of work effort for novices.
  - Novices sometimes compensate for lack of knowledge with better estimation strategies, e.g., “looking back” rather than ”step-by-step looking forward”.

21



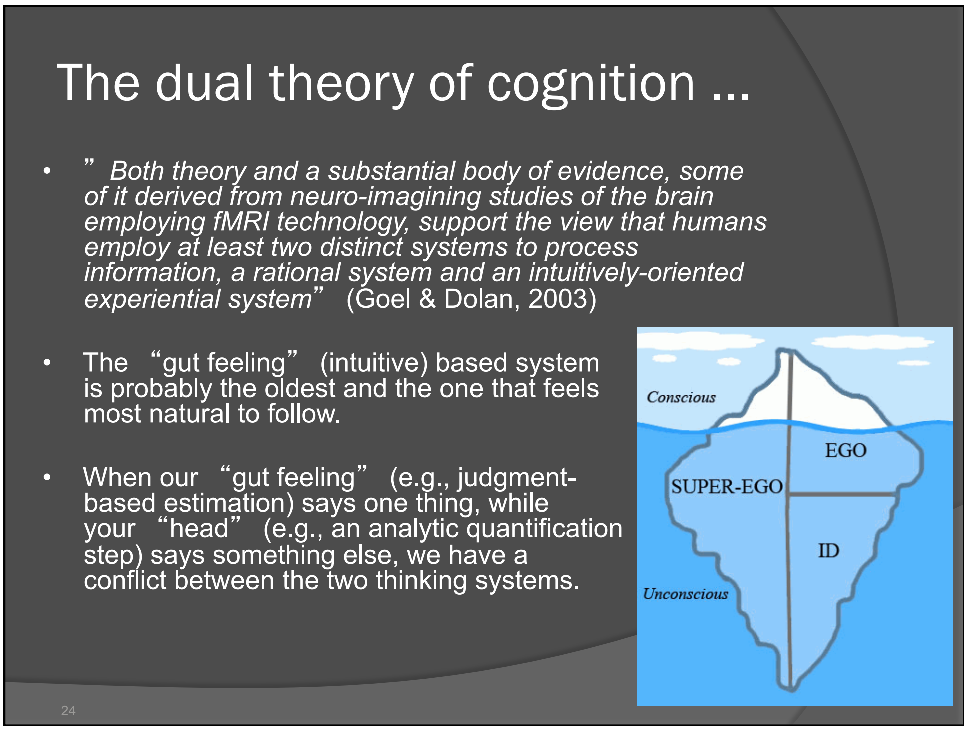
Indicators of estimation expertise

- Highest confidence in his/her estimate?
  - No. The most confident are typically the most over-optimistic.
- Those historically most accurate?
  - Yes, but not a very good indicator either.
  - The software professional (out of two) most over-optimistic on previous estimate had a 70% probability of being the most over-optimistic on the next estimate.
- Personality traits (optimistic life orientation, interdependence, “big five” traits, ...)?
  - Seems not to be of of much help.
- Slightly depressive people?
  - On average more realistic regarding own abilities.

22

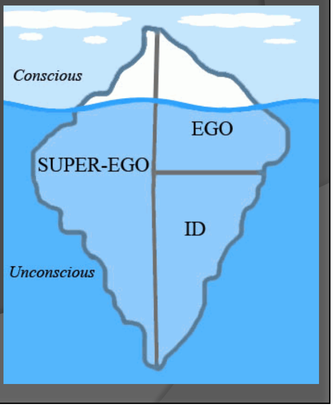
How to study expert judgment

The unconscious part of expert judgment

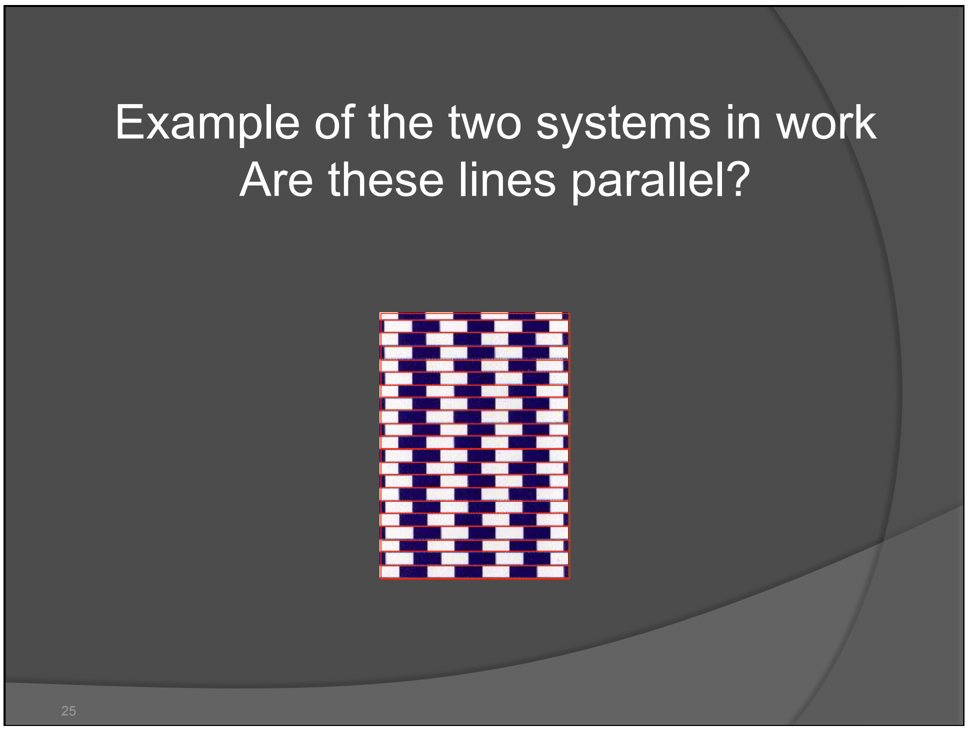


The dual theory of cognition …

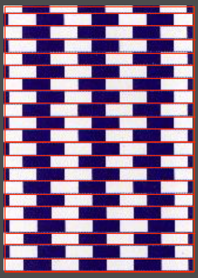
- ” *Both theory and a substantial body of evidence, some of it derived from neuro-imagining studies of the brain employing fMRI technology, support the view that humans employ at least two distinct systems to process information, a rational system and an intuitively-oriented experiential system*” (Goel & Dolan, 2003)
- The “gut feeling” (intuitive) based system is probably the oldest and the one that feels most natural to follow.
- When our “gut feeling” (e.g., judgment-based estimation) says one thing, while your “head” (e.g., an analytic quantification step) says something else, we have a conflict between the two thinking systems.



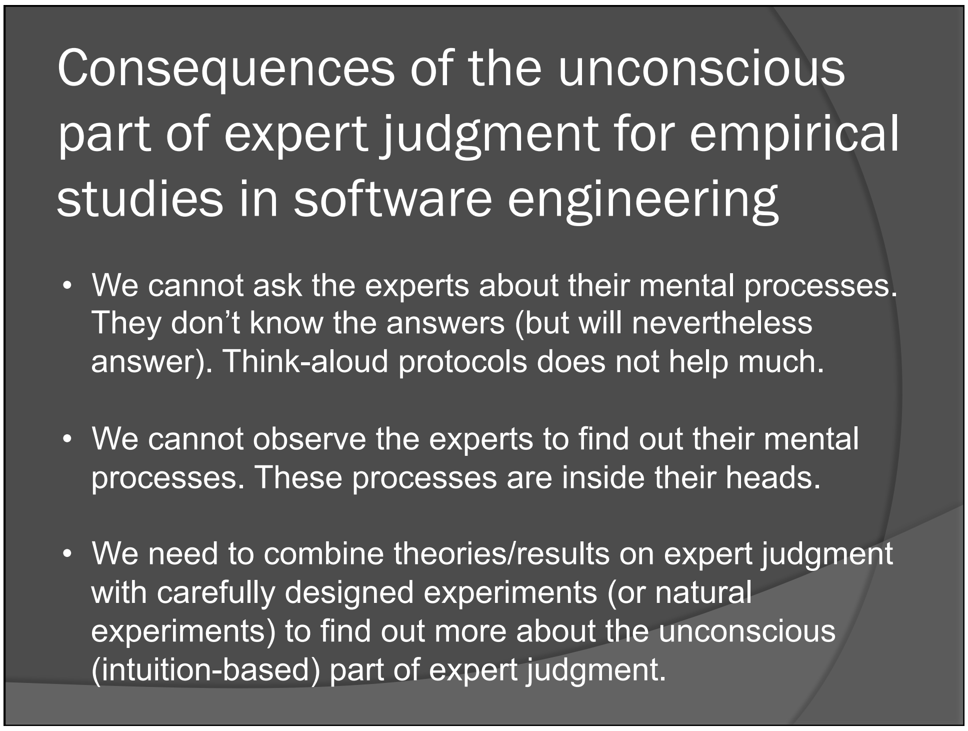
24



Example of the two systems in work
Are these lines parallel?



25

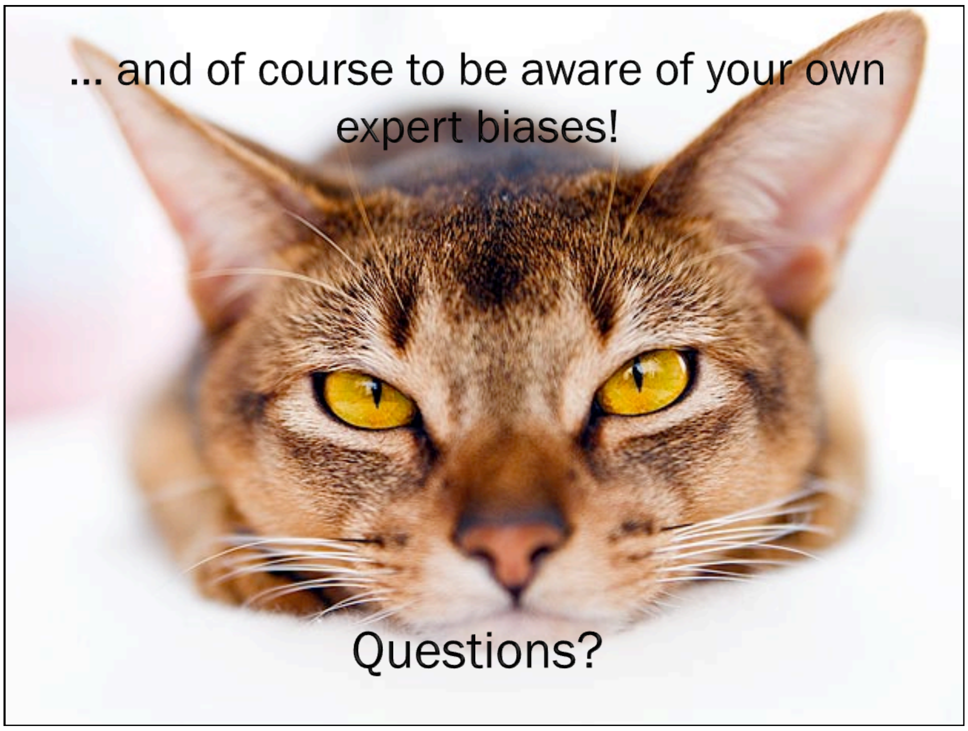


Consequences of the unconscious part of expert judgment for empirical studies in software engineering

- We cannot ask the experts about their mental processes. They don’t know the answers (but will nevertheless answer). Think-aloud protocols does not help much.
- We cannot observe the experts to find out their mental processes. These processes are inside their heads.
- We need to combine theories/results on expert judgment with carefully designed experiments (or natural experiments) to find out more about the unconscious (intuition-based) part of expert judgment.

What I would like you to remember

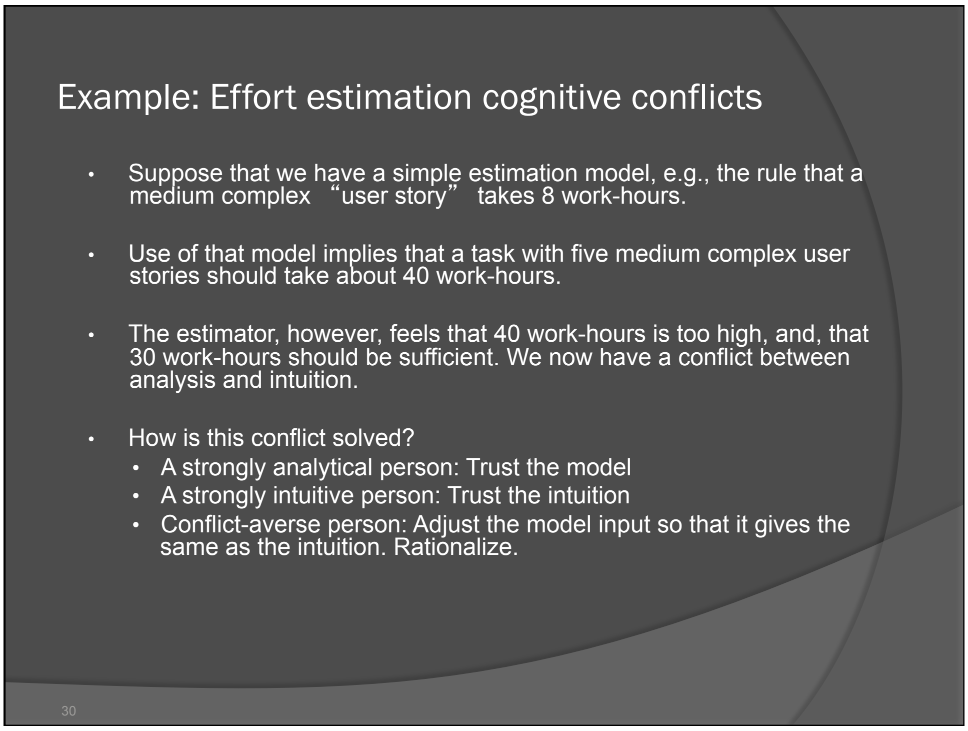
- Expert judgment is essential in most software engineering activities
- We should know about experts limitations and biases in important software engineering contexts
- We should try to understand how to build expert performance in software engineering
- We should try to understand when and how expert judgment should be supported or replaced with models/ explicit methods/tools
- To do this we need to be updated on results from psychology and use proper methods to study expert judgment



... and of course to be aware of your own expert biases!

Questions?

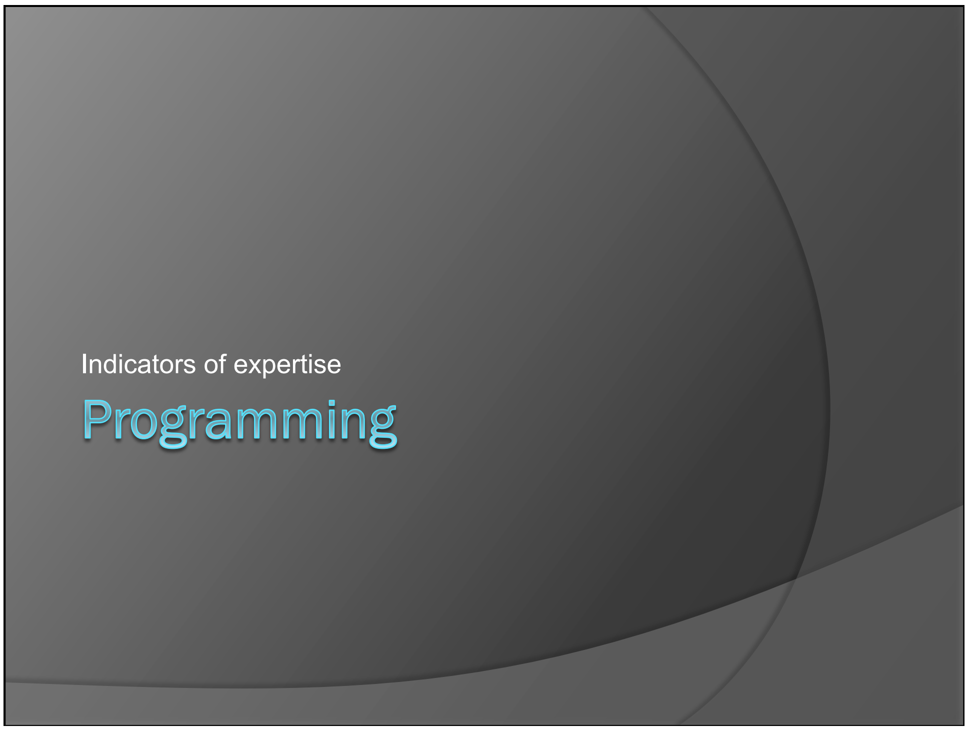
Extra material …



Example: Effort estimation cognitive conflicts

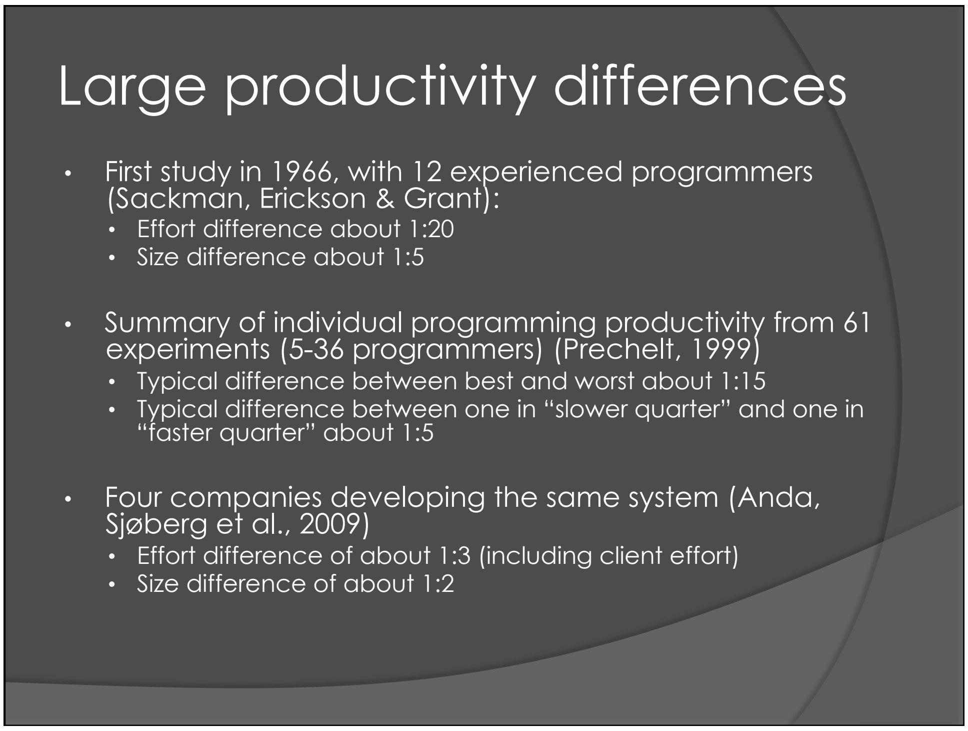
- Suppose that we have a simple estimation model, e.g., the rule that a medium complex “user story” takes 8 work-hours.
- Use of that model implies that a task with five medium complex user stories should take about 40 work-hours.
- The estimator, however, feels that 40 work-hours is too high, and, that 30 work-hours should be sufficient. We now have a conflict between analysis and intuition.
- How is this conflict solved?
  - A strongly analytical person: Trust the model
  - A strongly intuitive person: Trust the intuition
  - Conflict-averse person: Adjust the model input so that it gives the same as the intuition. Rationalize.

30



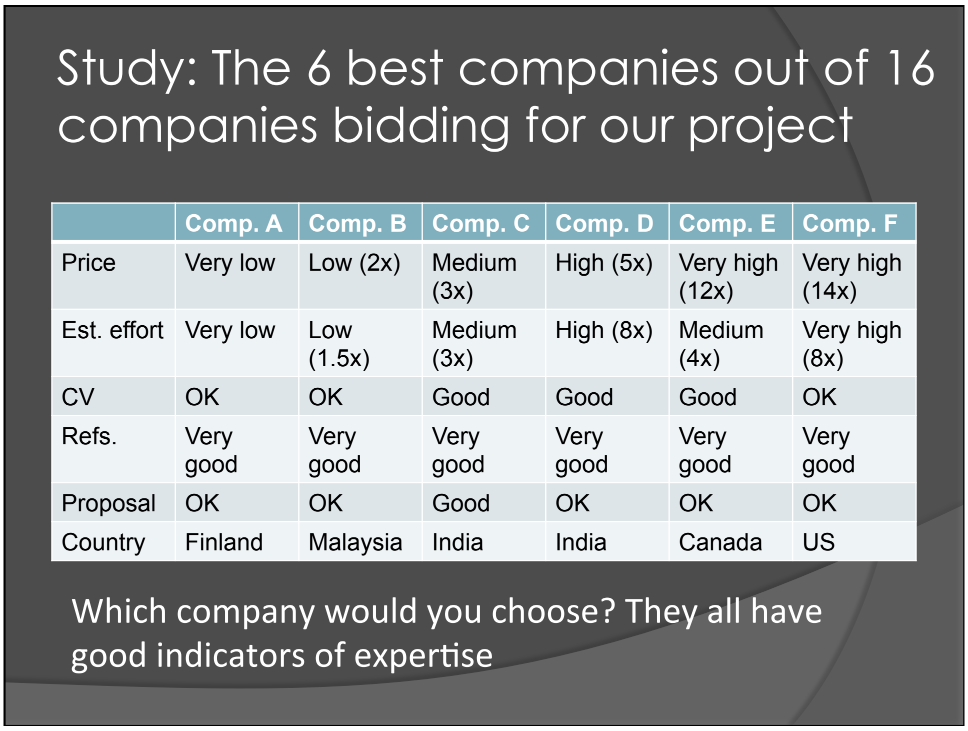
Indicators of expertise

Programming



Large productivity differences

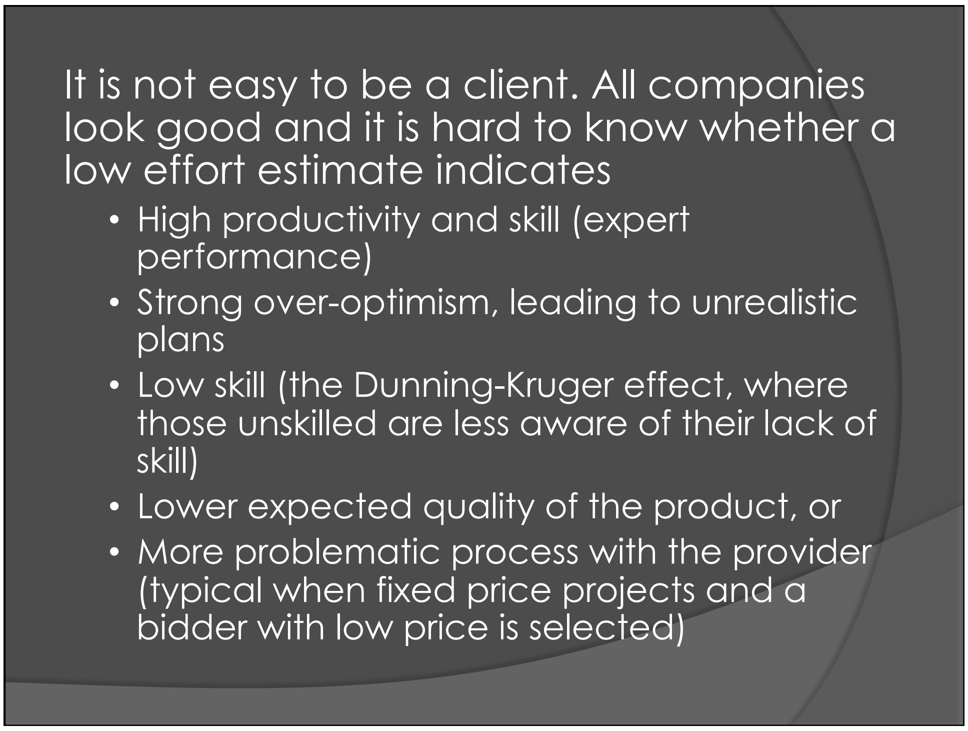
- First study in 1966, with 12 experienced programmers (Sackman, Erickson & Grant):
  - Effort difference about 1:20
  - Size difference about 1:5
- Summary of individual programming productivity from 61 experiments (5-36 programmers) (Prechelt, 1999)
  - Typical difference between best and worst about 1:15
  - Typical difference between one in "slower quarter" and one in "faster quarter" about 1:5
- Four companies developing the same system (Anda, Sjøberg et al., 2009)
  - Effort difference of about 1:3 (including client effort)
  - Size difference of about 1:2



Study: The 6 best companies out of 16 companies bidding for our project

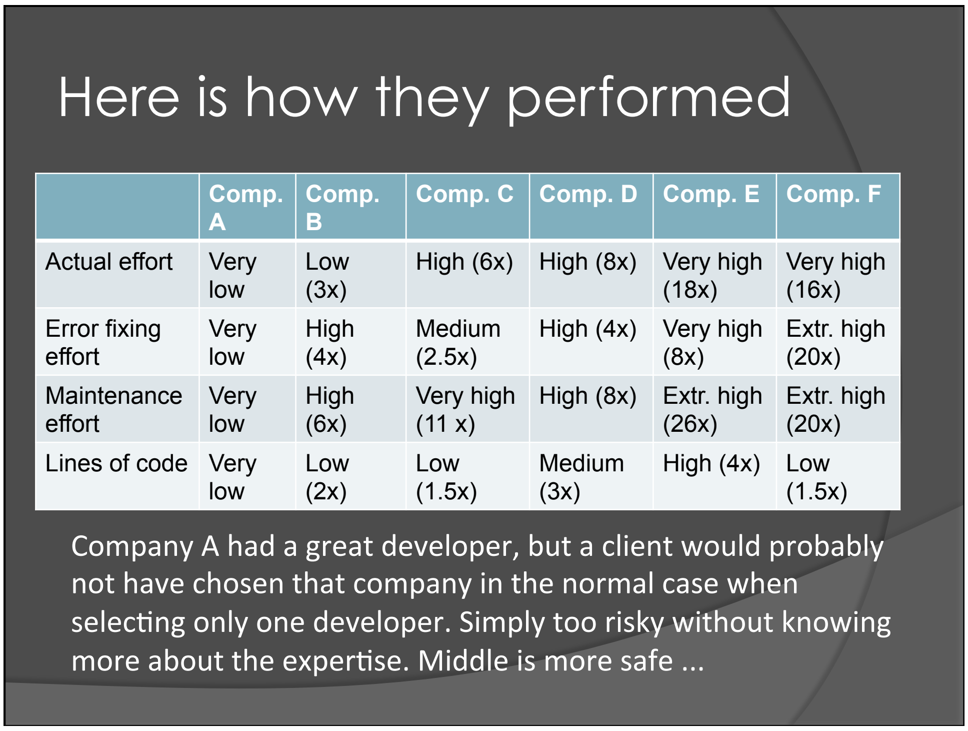
	Comp. A	Comp. B	Comp. C	Comp. D	Comp. E	Comp. F
Price	Very low	Low (2x)	Medium (3x)	High (5x)	Very high (12x)	Very high (14x)
Est. effort	Very low	Low (1.5x)	Medium (3x)	High (8x)	Medium (4x)	Very high (8x)
CV	OK	OK	Good	Good	Good	OK
Refs.	Very good	Very good	Very good	Very good	Very good	Very good
Proposal	OK	OK	Good	OK	OK	OK
Country	Finland	Malaysia	India	India	Canada	US

Which company would you choose? They all have good indicators of expertise



It is not easy to be a client. All companies look good and it is hard to know whether a low effort estimate indicates

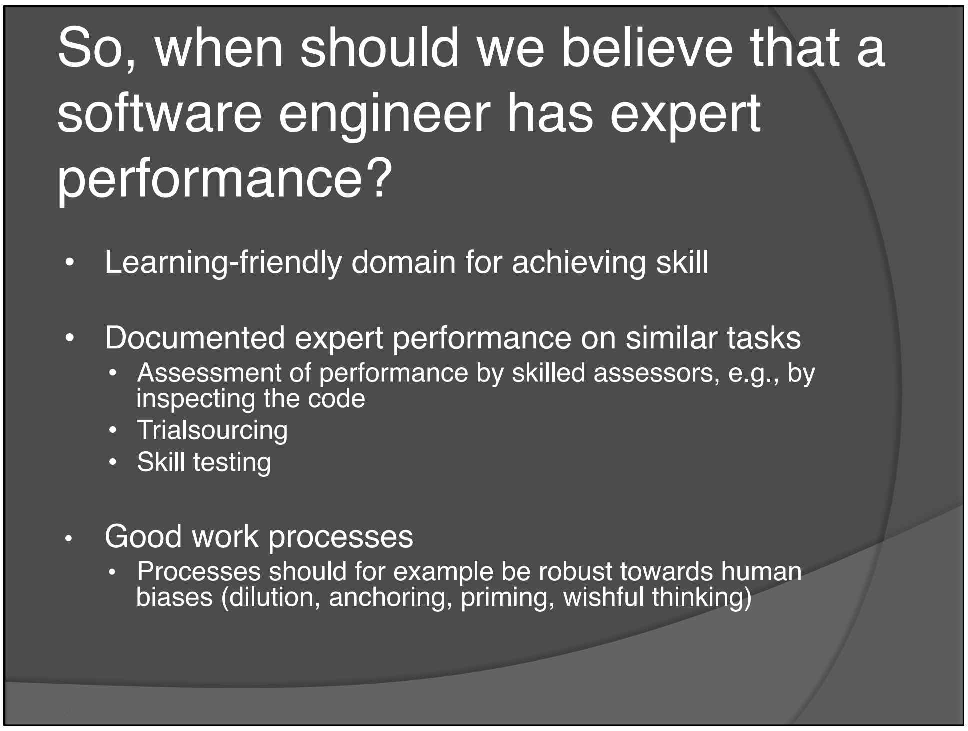
- High productivity and skill (expert performance)
- Strong over-optimism, leading to unrealistic plans
- Low skill (the Dunning-Kruger effect, where those unskilled are less aware of their lack of skill)
- Lower expected quality of the product, or
- More problematic process with the provider (typical when fixed price projects and a bidder with low price is selected)



Here is how they performed

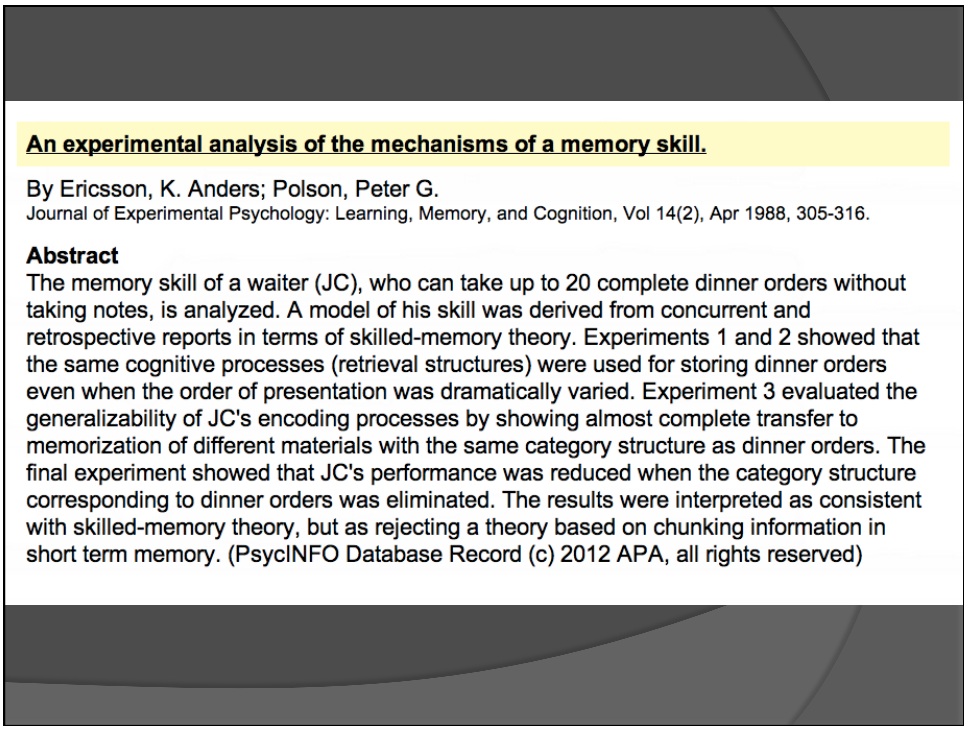
	Comp. A	Comp. B	Comp. C	Comp. D	Comp. E	Comp. F
Actual effort	Very low	Low (3x)	High (6x)	High (8x)	Very high (18x)	Very high (16x)
Error fixing effort	Very low	High (4x)	Medium (2.5x)	High (4x)	Very high (8x)	Extr. high (20x)
Maintenance effort	Very low	High (6x)	Very high (11 x)	High (8x)	Extr. high (26x)	Extr. high (20x)
Lines of code	Very low	Low (2x)	Low (1.5x)	Medium (3x)	High (4x)	Low (1.5x)

Company A had a great developer, but a client would probably not have chosen that company in the normal case when selecting only one developer. Simply too risky without knowing more about the expertise. Middle is more safe ...



So, when should we believe that a software engineer has expert performance?

- Learning-friendly domain for achieving skill
- Documented expert performance on similar tasks
  - Assessment of performance by skilled assessors, e.g., by inspecting the code
  - Trialsourcing
  - Skill testing
- Good work processes
  - Processes should for example be robust towards human biases (dilution, anchoring, priming, wishful thinking)



An experimental analysis of the mechanisms of a memory skill.

By Ericsson, K. Anders; Polson, Peter G.
Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol 14(2), Apr 1988, 305-316.

Abstract
The memory skill of a waiter (JC), who can take up to 20 complete dinner orders without taking notes, is analyzed. A model of his skill was derived from concurrent and retrospective reports in terms of skilled-memory theory. Experiments 1 and 2 showed that the same cognitive processes (retrieval structures) were used for storing dinner orders even when the order of presentation was dramatically varied. Experiment 3 evaluated the generalizability of JC's encoding processes by showing almost complete transfer to memorization of different materials with the same category structure as dinner orders. The final experiment showed that JC's performance was reduced when the category structure corresponding to dinner orders was eliminated. The results were interpreted as consistent with skilled-memory theory, but as rejecting a theory based on chunking information in short term memory. (PsycINFO Database Record (c) 2012 APA, all rights reserved)

Who is an expert?

High performance?

Certified?

Self-proclaimed, expert-like behaviour?

Other criteria include long experience, knowledgeable and, in domain where evaluation is difficult, those with judgments consistent with those of other expert or those good at discrimination between stimuli and consistent in judgment (CWS-test)