How to identify risky IT projects and avoid them turning into black swans

Magne Jørgensen

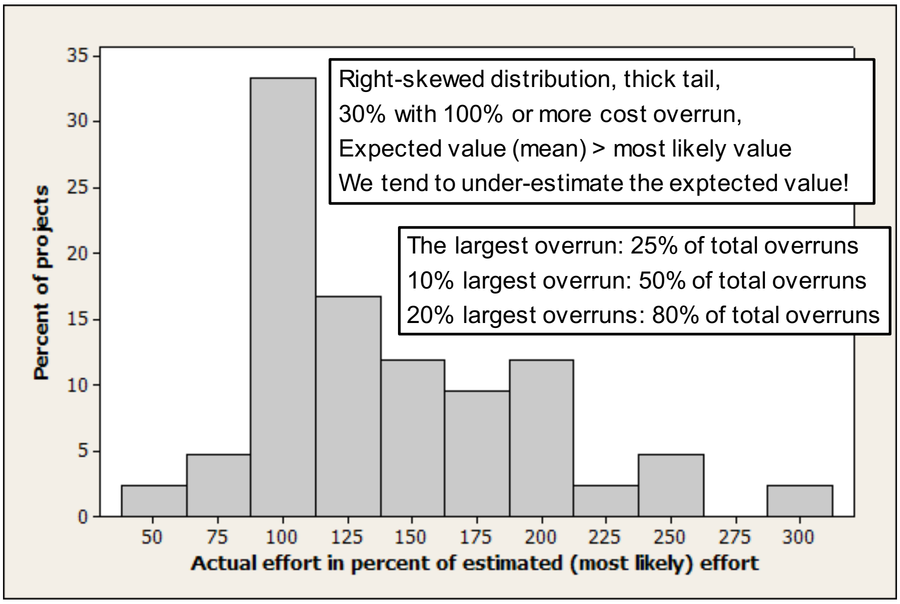
Simula Research Laboratory

Universitetet i Oslo

Scienta

Black Swan (Taleb): An outlier event with severe consequences. Here used to denote a software project with large budget overrun or other severe problems.

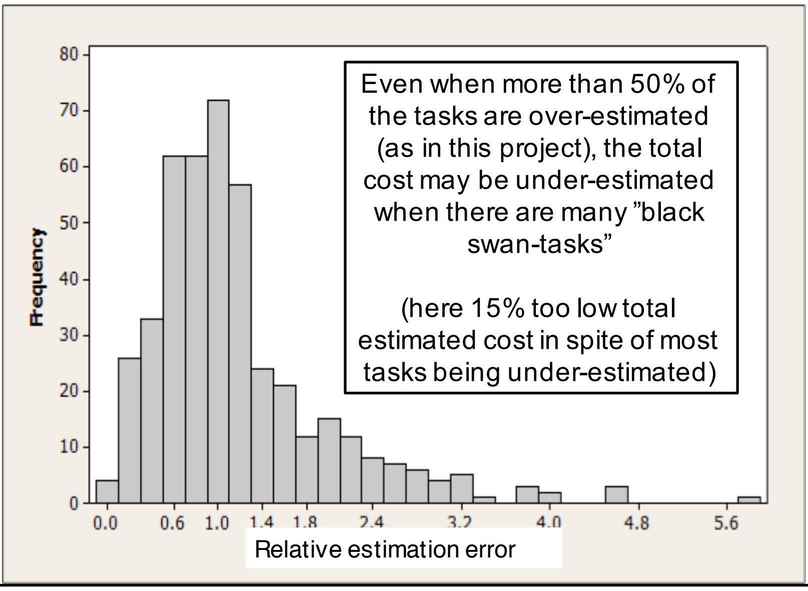
A typical distribution of cost overrun in software projects



Right-skewed distribution, thick tail,
30% with 100% or more cost overrun,
Expected value (mean) > most likely value
We tend to under-estimate the exptected value!

The largest overrun: 25% of total overruns
10% largest overrun: 50% of total overruns
20% largest overruns: 80% of total overruns

Black swans (right-skewed, fat tail) are also relevant at task level



Even when more than 50% of the tasks are over-estimated (as in this project), the total cost may be under-estimated when there are many ”black swan-tasks”

(here 15% too low total estimated cost in spite of most tasks being under-estimated)

Higher Rate of Black Swans in IT Compared to Other Project Types

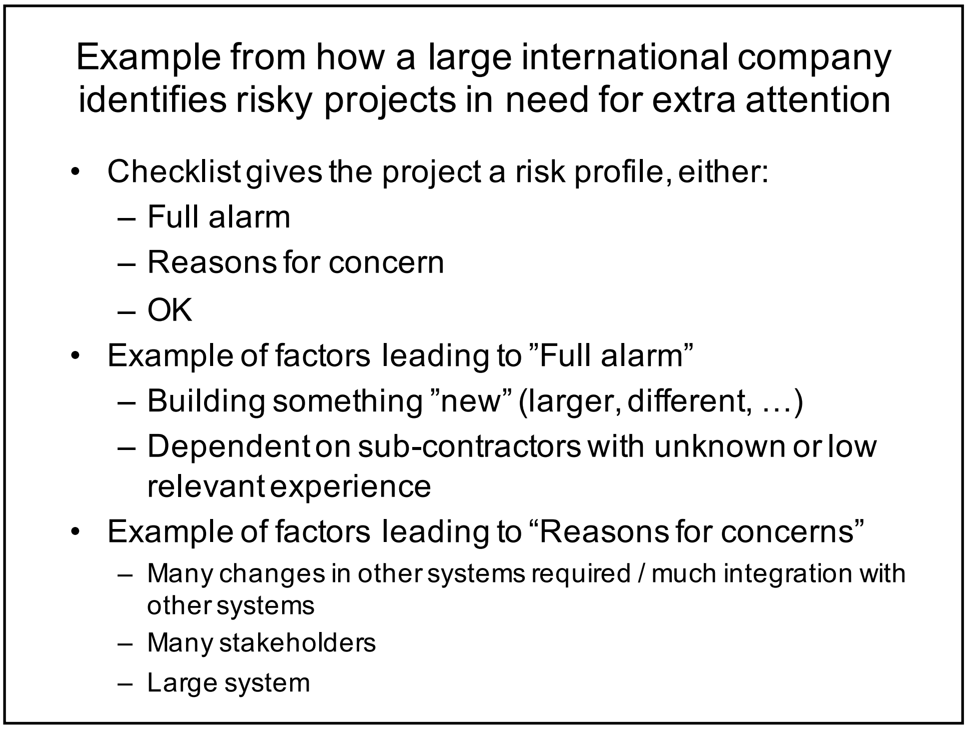
	Roads	Links	Energy	Rail	Dams	IT	Olympics
Cost overrun	20%	34%	36%	45%	90%	107%	219%
Frequency of cost overrun	9 of 10	9 of 10	6 of 10	9 of 10	7 of 10	5 of 10	10 of 10
Schedule overrun	38%	23%	38%	45%	44%	37%	0%
Benefits shortfall	10%	n/a	n/a	-51%	-11%	-29%	n/a
Cost Black Swans	5%	9%	7%	6%	10%	18%	6%
Ø duration (years)	5.5	8.0	5.3	7.8	8.2	3.3	7.0

It is actually even worse than the data on completed projects suggests:

10-20% of all IT-projects are never completed or completed with no or little client benefits.

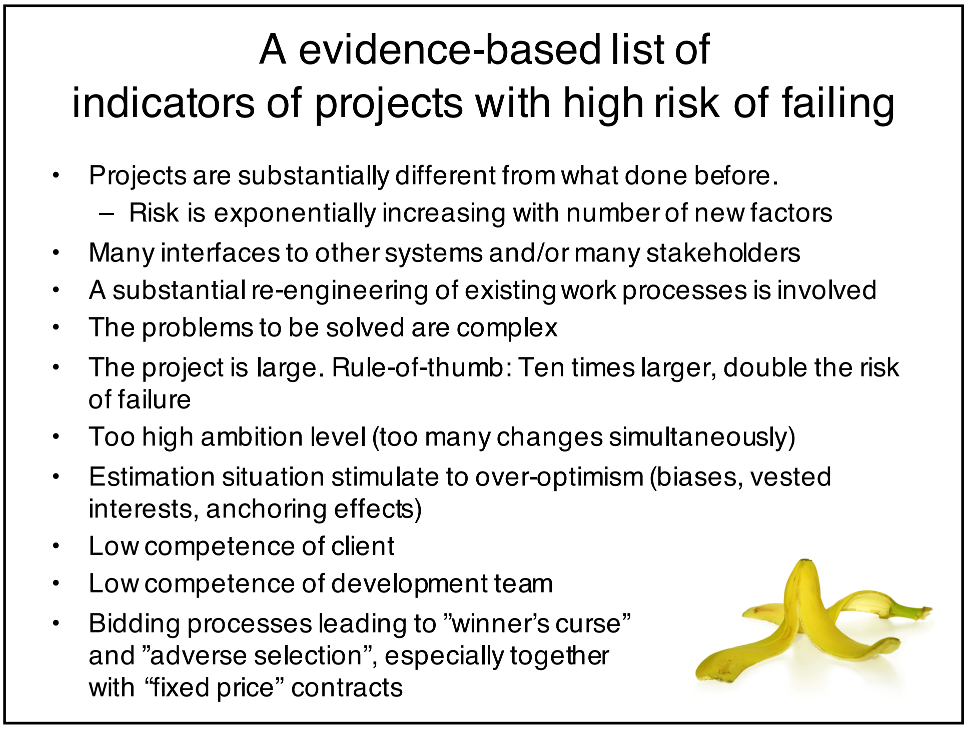
Consequences

- A few Black Swan projects (large cost overruns) will dominate the total project portfolio performance
  – It is the expected value (mean), not the median or the most likely cost overruns that matters for the total performance
  – In addition to budget problems, a few Black Swan projects may have a strong negative effect on market reputation, client relationship, profit ….
- Management focus should be strong on identifying and managing (learn to reduce risks and how to react on crisis) projects with high risk of becoming Black Swans.



Example from how a large international company identifies risky projects in need for extra attention

- Checklist gives the project a risk profile, either:
  - Full alarm
  - Reasons for concern
  - OK
- Example of factors leading to ”Full alarm”
  - Building something ”new” (larger, different, …)
  - Dependent on sub-contractors with unknown or low relevant experience
- Example of factors leading to “Reasons for concerns”
  - Many changes in other systems required / much integration with other systems
  - Many stakeholders
  - Large system



A evidence-based list of indicators of projects with high risk of failing

- Projects are substantially different from what done before.
  - Risk is exponentially increasing with number of new factors
- Many interfaces to other systems and/or many stakeholders
- A substantial re-engineering of existing work processes is involved
- The problems to be solved are complex
- The project is large. Rule-of-thumb: Ten times larger, double the risk of failure
- Too high ambition level (too many changes simultaneously)
- Estimation situation stimulate to over-optimism (biases, vested interests, anchoring effects)
- Low competence of client
- Low competence of development team
- Bidding processes leading to ”winner’s curse” and ”adverse selection”, especially together with “fixed price” contracts

Failure factors from a study of >500.000 small projects

Predictor variable	Coefficient	p-value	Odds ratio	95% confidence interval	
				Lower	Upper
Constant	-2.90	0.00			
SatisfactionScoreProviderCat=Low	0.35	0.00	1.42	1.39	1.45
SatisfactionScoreProviderCat=No Scores	0.91	0.00	2.49	2.33	2.67
FailureRateProviderCat=Low	-0.66	0.00	0.52	0.51	0.53
FailRateProviderCat=No Projects	-0.34	0.00	0.71	0.67	0.76
SkillTestPassRateProviderCat=Low	0.07	0.00	1.07	1.02	1.12
SkillTestPassRateProviderCat=No Tests	0.58	0.00	1.79	1.74	1.85
SatisfactionScoreClientCat=Low	0.18	0.00	1.20	1.17	1.23
SatisfactionScoreClientCat=No Scores	0.25	0.00	1.28	1.23	1.33
FailureRateClientCat=Low	-0.64	0.00	0.53	0.52	0.54
FailureRateClientCat=No Projects	-0.63	0.00	0.53	0.51	0.56
PreviousCollaboration=Yes	-1.74	0.00	0.17	0.17	0.18
FocusLowPriceCat=Low	-0.19	0.00	0.83	0.81	0.85
FocusLowPriceCat=Medium	-0.08	0.00	0.92	0.89	0.95
FailureRateProviderRegionCat=High	0.27	0.00	1.31	1.28	1.33
FailureRateClientRegionCat=High	0.42	0.00	1.53	1.48	1.58
GeographicalDistance=Neighbor	-0.07	0.02	0.93	0.90	0.97
GeographicalDistance=Offshore	0.02	0.10	1.02	1.00	1.05
logProjectSize	0.71	0.00	2.03	1.99	2.06

Regional differences in failure rate

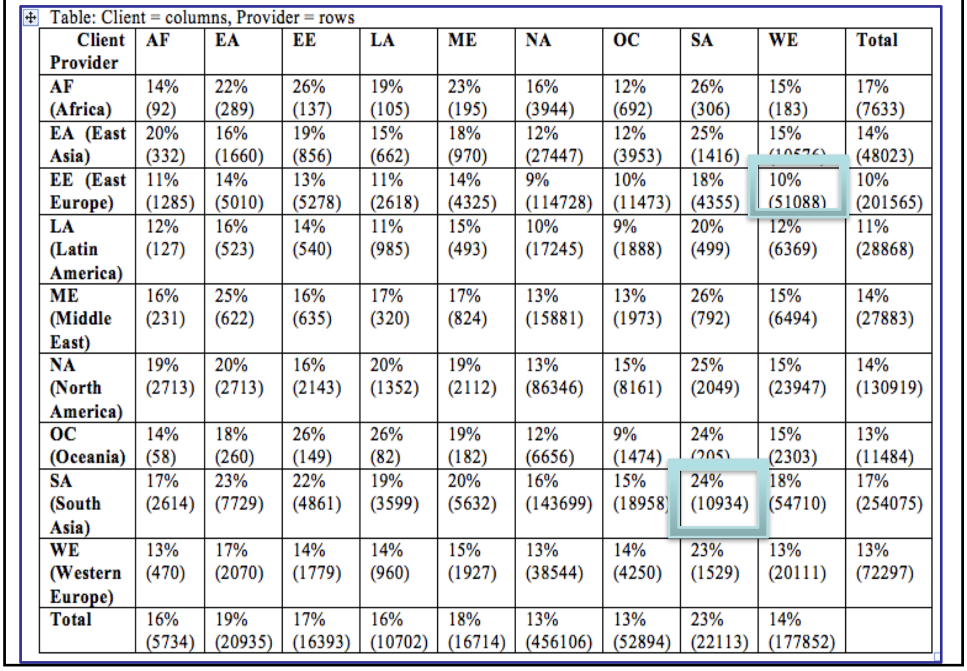


Table: Client = columns, Provider = rows

Client Provider	AF	EA	EE	LA	ME	NA	OC	SA	WE	Total
AF (Africa)	14% (92)	22% (289)	26% (137)	19% (105)	23% (195)	16% (3944)	12% (692)	26% (306)	15% (183)	17% (7633)
EA (East Asia)	20% (332)	16% (1660)	19% (856)	15% (662)	18% (970)	12% (27447)	12% (3953)	25% (1416)	15% (10576)	14% (48023)
EE (East Europe)	11% (1285)	14% (5010)	13% (5278)	11% (2618)	14% (4325)	9% (114728)	10% (11473)	18% (4355)	10% (51088)	10% (201565)
LA (Latin America)	12% (127)	16% (523)	14% (540)	11% (985)	15% (493)	10% (17245)	9% (1888)	20% (499)	12% (6369)	11% (28868)
ME (Middle East)	16% (231)	25% (622)	16% (635)	17% (320)	17% (824)	13% (15881)	13% (1973)	26% (792)	15% (6494)	14% (27883)
NA (North America)	19% (2713)	20% (2713)	16% (2143)	20% (1352)	19% (2112)	13% (86346)	15% (8161)	25% (2049)	15% (23947)	14% (130919)
OC (Oceania)	14% (58)	18% (260)	26% (149)	26% (82)	19% (182)	12% (6656)	9% (1474)	24% (205)	15% (2303)	13% (11484)
SA (South Asia)	17% (2614)	23% (7729)	22% (4861)	19% (3599)	20% (5632)	16% (143699)	15% (18958)	24% (10934)	18% (54710)	17% (254075)
WE (Western Europe)	13% (470)	17% (2070)	14% (1779)	14% (960)	15% (1927)	13% (38544)	14% (4250)	23% (1529)	13% (20111)	13% (72297)
Total	16% (5734)	19% (20935)	17% (16393)	16% (10702)	18% (16714)	13% (456106)	13% (52894)	23% (22113)	14% (177852)	

Reasons for too low cost estimates

- Judgment biases
  - Wishful thinking
  - Illusion of control
  - Anchoring effects
  - Sequence effects +++
- Ignorance of an uncertain and right-skewed world
  - Ignorance of unknown unknowns
  - Ignorance of ”Black Swan” (outlier)-effects
- Selection bias
  - Winner’s curse
- Deception (”strategic” estimation)

JUDGMENT BIASES

Awareness of judgment biases helps ...

McKinsey-research results from 2010:

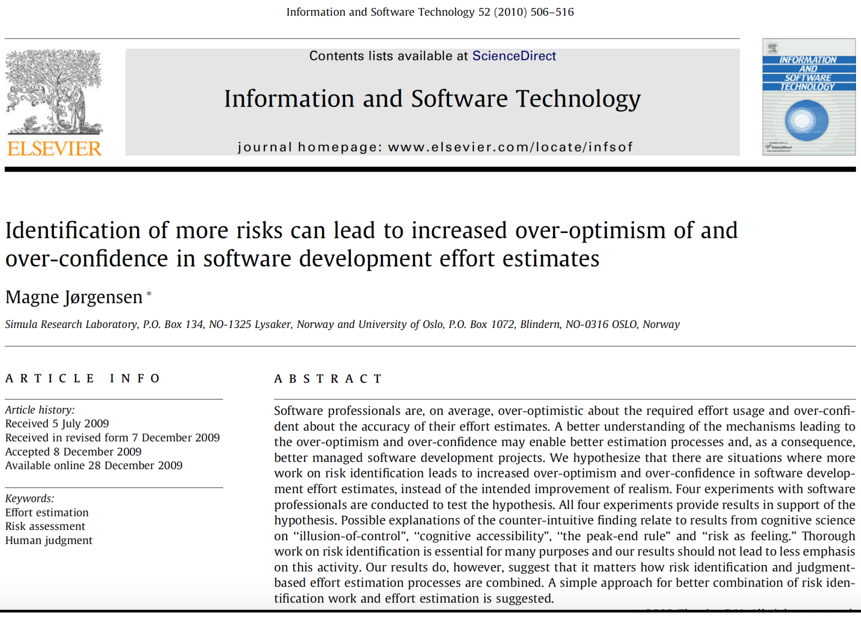
•Study of 1048 strategic decisions in several companies.

•The 25% companies best at avoiding and reducing decision biases (= better use of evidence) had a profit (Return on Investment) seven times better than the 25% worst.

•Avoiding and reducing decision biases was six times more important for the profit than the amount of or level of detail of analysis preceding the decision.

Lovallo, Dan, and Olivier Sibony. "The case for behavioral strategy." *McKinsey Quarterly* 2 (2010): 30-43.www.edpiccolino.com/workspace/articles/mckinsey-the-case-for-behavioral-strategy.pdf)

Illusion of control



Information and Software Technology 52 (2010) 506–516

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates

Magne Jørgensen *

Simula Research Laboratory, P.O. Box 134, NO-1325 Lysaker, Norway and University of Oslo, P.O. Box 1072, Blindern, NO-0316 OSLO, Norway

ARTICLE INFO

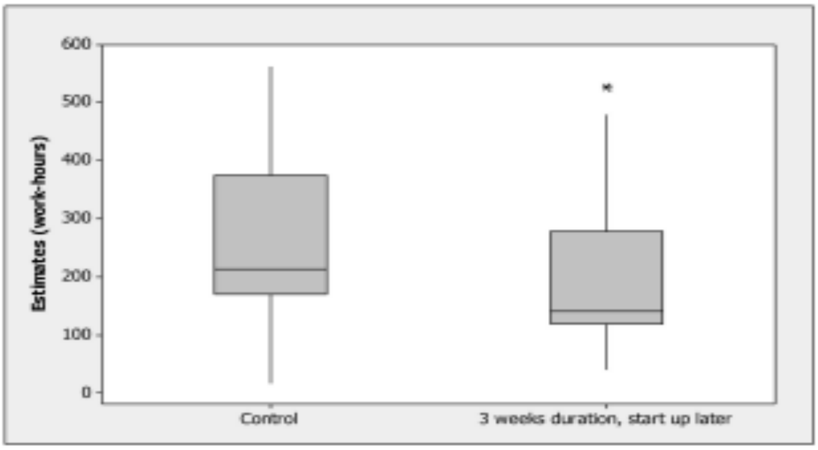
Article history:
Received 5 July 2009
Received in revised form 7 December 2009
Accepted 8 December 2009
Available online 28 December 2009

Keywords:
Effort estimation
Risk assessment
Human judgment

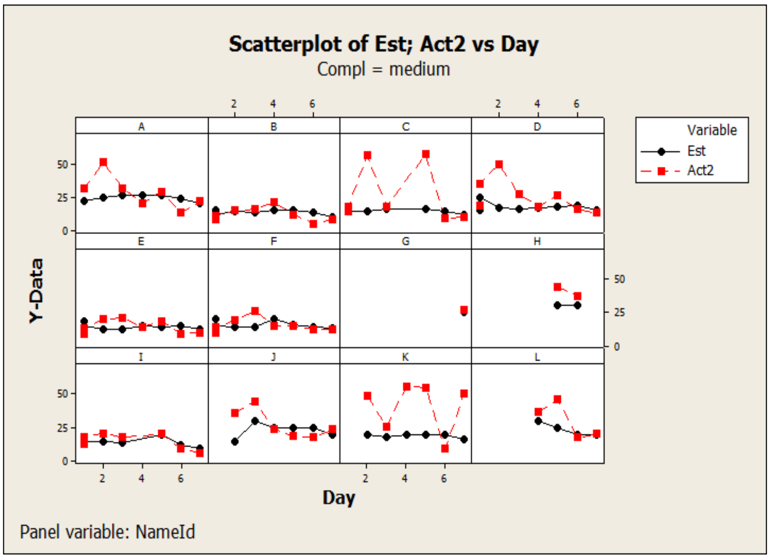
ABSTRACT

Software professionals are, on average, over-optimistic about the required effort usage and over-confident about the accuracy of their effort estimates. A better understanding of the mechanisms leading to the over-optimism and over-confidence may enable better estimation processes and, as a consequence, better managed software development projects. We hypothesize that there are situations where more work on risk identification leads to increased over-optimism and over-confidence in software development effort estimates, instead of the intended improvement of realism. Four experiments with software professionals are conducted to test the hypothesis. All four experiments provide results in support of the hypothesis. Possible explanations of the counter-intuitive finding relate to results from cognitive science on "illusion-of-control", "cognitive accessibility", "the peak-end rule" and "risk as feeling." Thorough work on risk identification is essential for many purposes and our results should not lead to less emphasis on this activity. Our results do, however, suggest that it matters how risk identification and judgment-based effort estimation processes are combined. A simple approach for better combination of risk identification work and effort estimation is suggested.

Wishful thinking



Estimated time = ideal time



Anchoring effects

Experiment:

- HIGH (LOW) group: *"The customer has indicated that he believes that **1000 (50)** work-hours is a reasonable effort estimate for the specified system. However, the customer knows very little about the implications of his specification on the development effort and you shall not let the customer's expectations impact your estimate. Your task is to provide a realistic effort estimate of a system that meets the requirements specification and has a sufficient quality."*

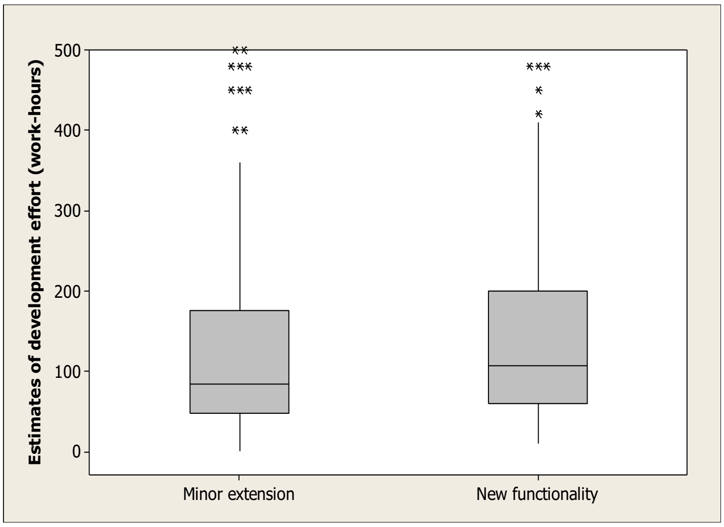
- Participants: Experienced software developers.

- All (HIGH, LOW, CONTROL) received the same requirement specification.

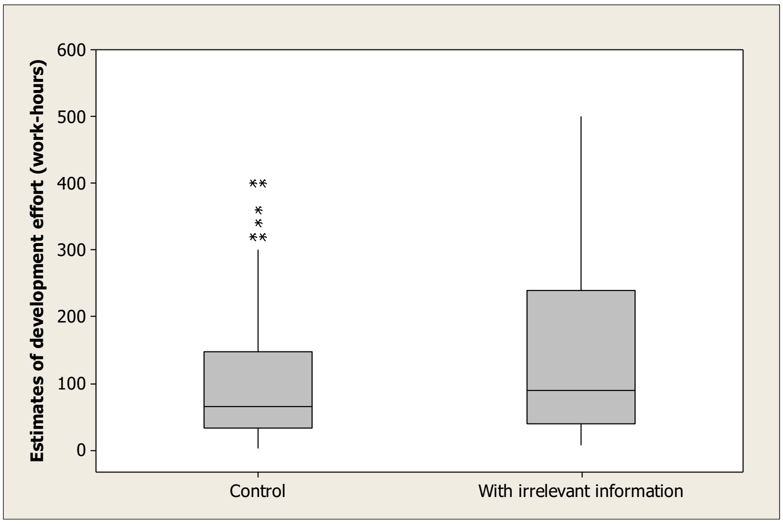
Anchoring effects

- Results:
  - – HIGH group average: 555 work-hours
  - – CONTROL group (no anchor) average: 456 work-hours
  - – LOW group average: 99 work-hours!!!
- None of the developers thought they were much affected by the client expectation.
- Learning about and attention to the anchoring effect helps, but do not remove the effect. Avoidance is the only really effective strategy!

Framing



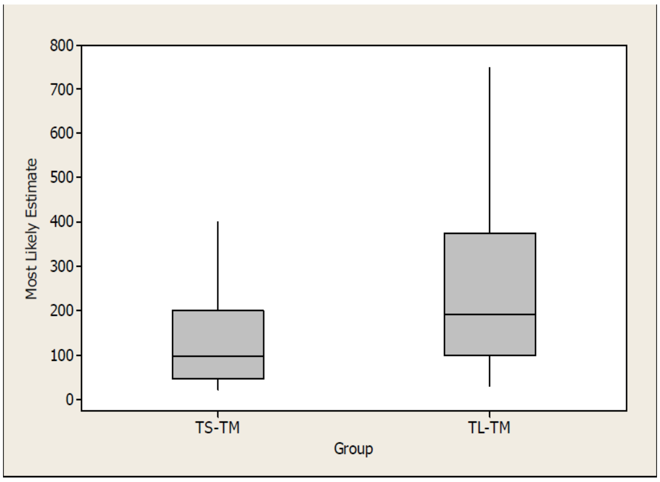
Irrelevant information



Unitosity (Do you think work-hours or workdays gives higher estimates?)

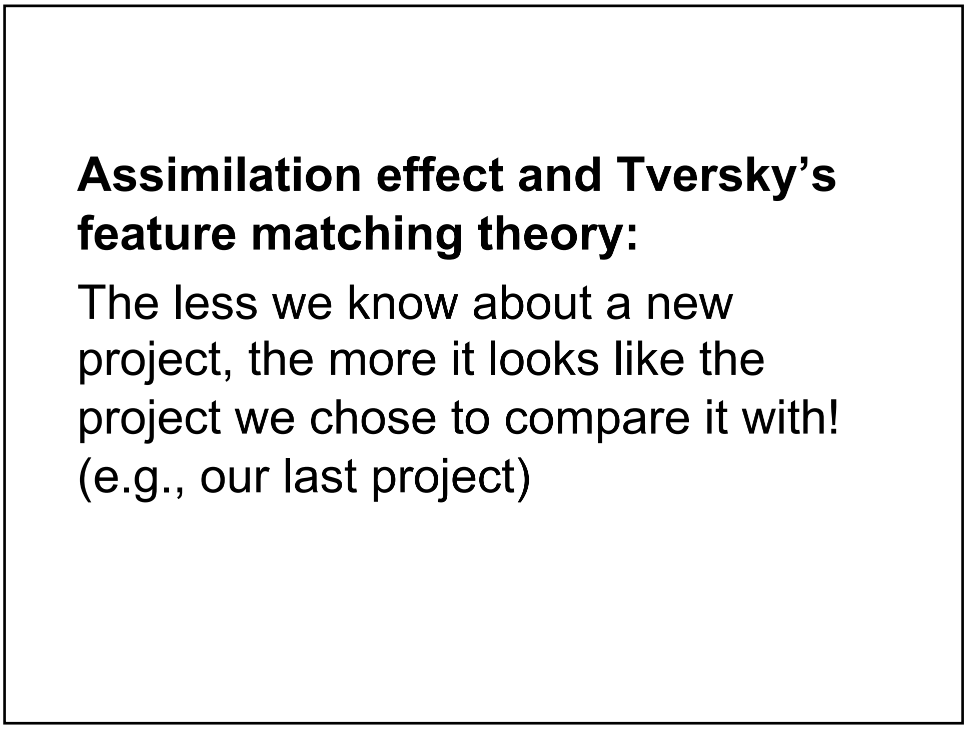
	Most likely effort (mean)
Estimates in workdays, *n*=29	177 work-hours (std. dev. 254)
Estimates in work-hours, *n*=19	72 work-hours (std. dev. 57)
Difference between Groups	105 work-hours (59% decrease)
T-test of difference (*p*-value)	0.02[1]

Sequence effect (assimilation effect)



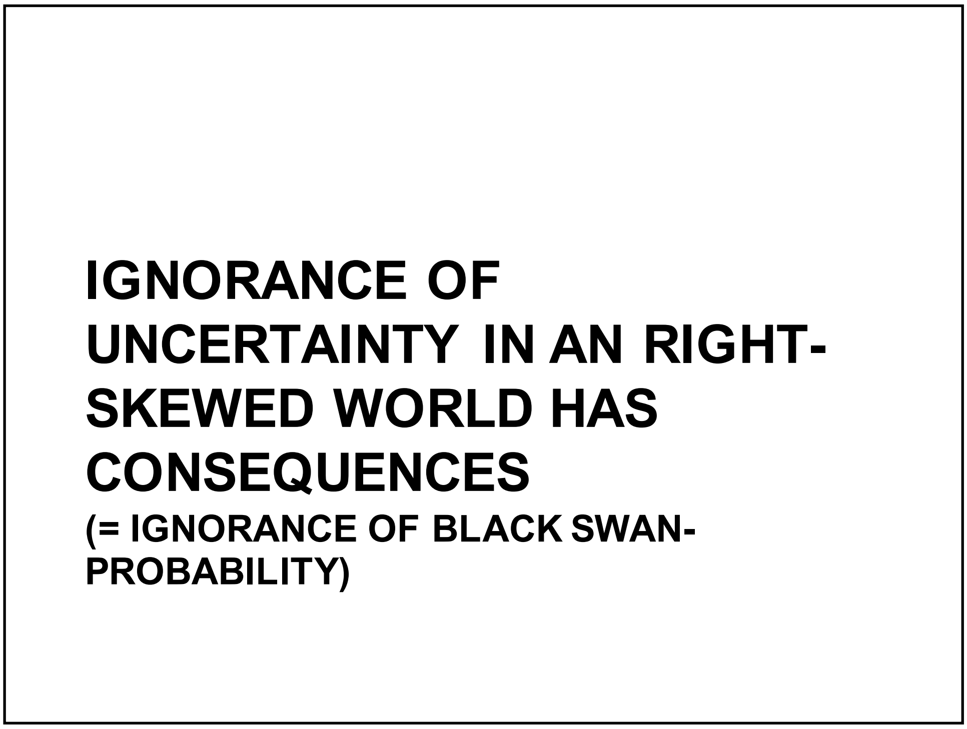
TS-TM: Estimation of small, then medium large task

TL-TM: Estimation of large, then medium large task



Assimilation effect and Tversky’s feature matching theory:

The less we know about a new project, the more it looks like the project we chose to compare it with! (e.g., our last project)



IGNORANCE OF UNCERTAINTY IN AN RIGHT-SKEWED WORLD HAS CONSEQUENCES

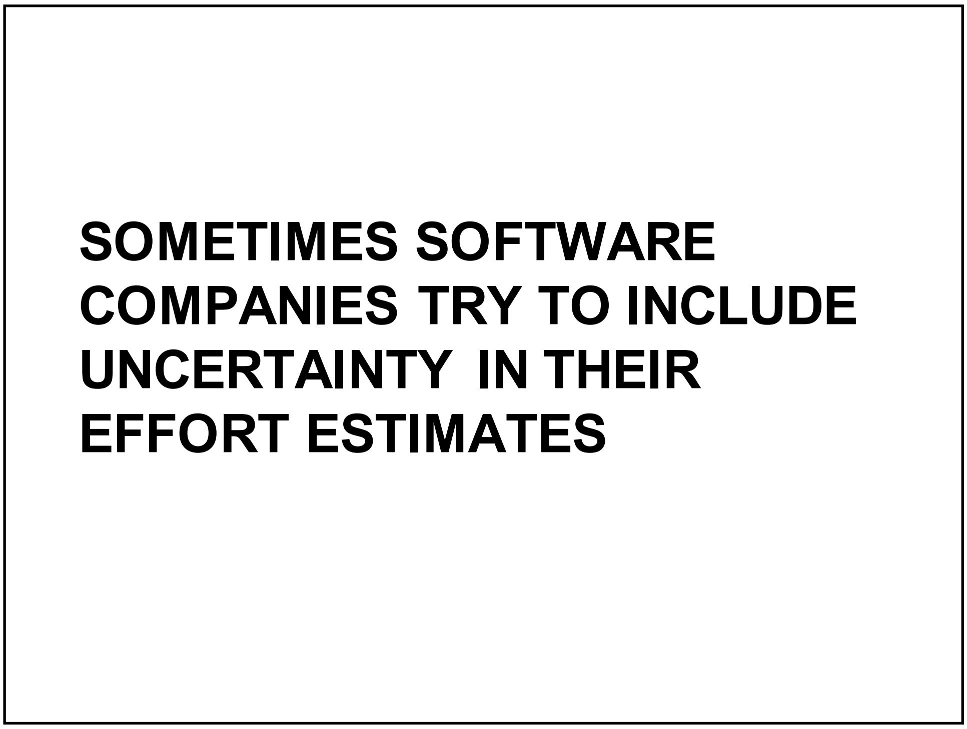
(= IGNORANCE OF BLACK SWAN-PROBABILITY)

What is an estimate? A survey.

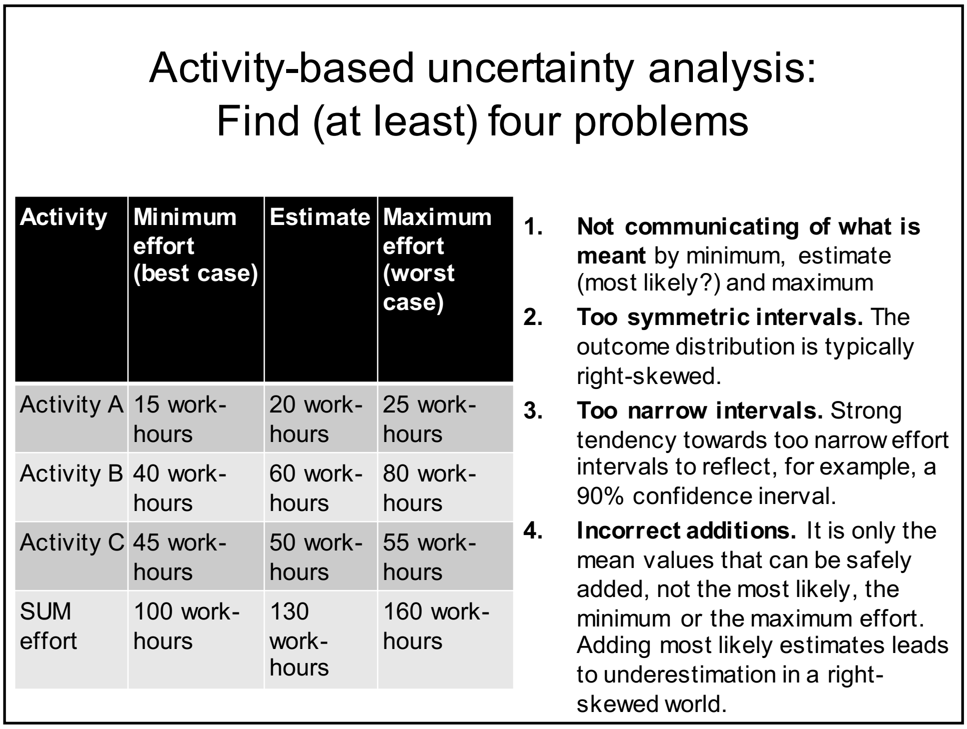
"You have just estimated the number of work-hours you think you need to develop and test four different software systems. Please select the description below that you think is closest to what you meant by your effort estimate in the previous four estimation tasks:

- *Number of work-hours I will use given that I experience almost no problems.*
- *Number of work-hours I will use given that I experience no major problems.*
- *Number of work-hours I most likely will use.*
- *Number of work-hours where it is about just as likely that I will use more as it is that I will use less effort than estimated.*
- *Number of work-hours where it is unlikely that I will use more effort than estimated.*
- *Number of work-hours based on my expert judgment/feeling of how many work-hours I will use. I find it difficult to decide about the exact meaning of the estimate.*
- *None of the above descriptions is close to what I typically mean by an effort estimate."*

Interpretation of "effort estimate" (as claimed in hindsight)	**Frequency of interpretation**
Ideal effort	37%
Most likely effort	27%
Median effort (p50)	5%
Risk averse effort	9%
Don't know/gut feeling/other	22%



SOMETIMES SOFTWARE COMPANIES TRY TO INCLUDE UNCERTAINTY IN THEIR EFFORT ESTIMATES

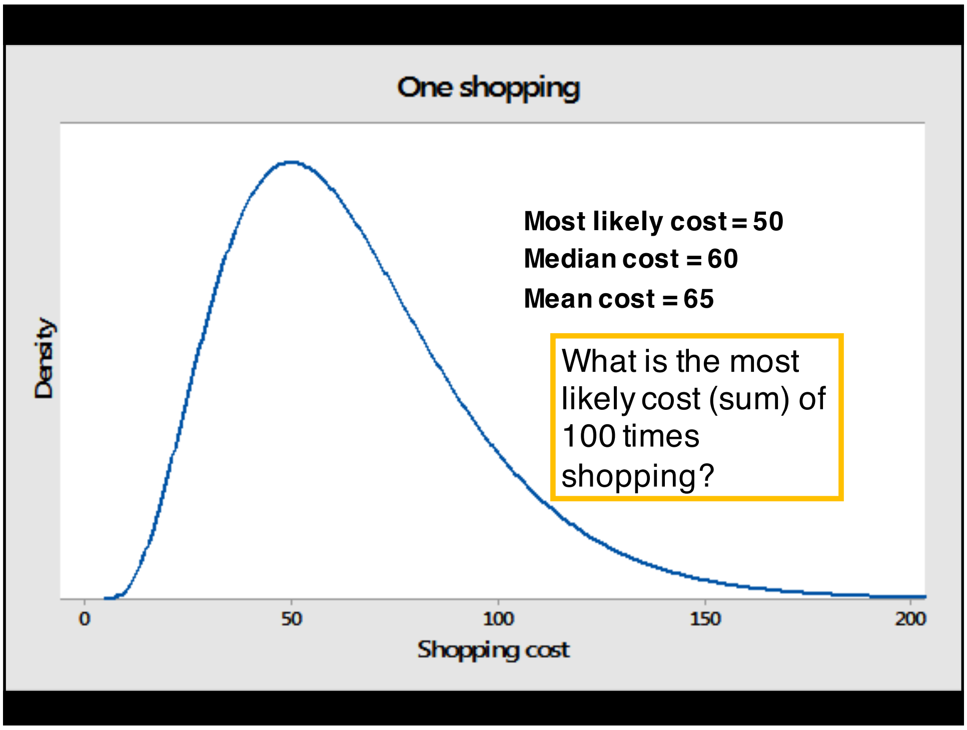


Activity-based uncertainty analysis: Find (at least) four problems

Activity	Minimum effort (best case)	Estimate	Maximum effort (worst case)
Activity A	15 work-hours	20 work-hours	25 work-hours
Activity B	40 work-hours	60 work-hours	80 work-hours
Activity C	45 work-hours	50 work-hours	55 work-hours
SUM effort	100 work-hours	130 work-hours	160 work-hours

1. **Not communicating of what is meant** by minimum, estimate (most likely?) and maximum
2. **Too symmetric intervals.** The outcome distribution is typically right-skewed.
3. **Too narrow intervals.** Strong tendency towards too narrow effort intervals to reflect, for example, a 90% confidence inerval.
4. **Incorrect additions.** It is only the mean values that can be safely added, not the most likely, the minimum or the maximum effort. Adding most likely estimates leads to underestimation in a right-skewed world.

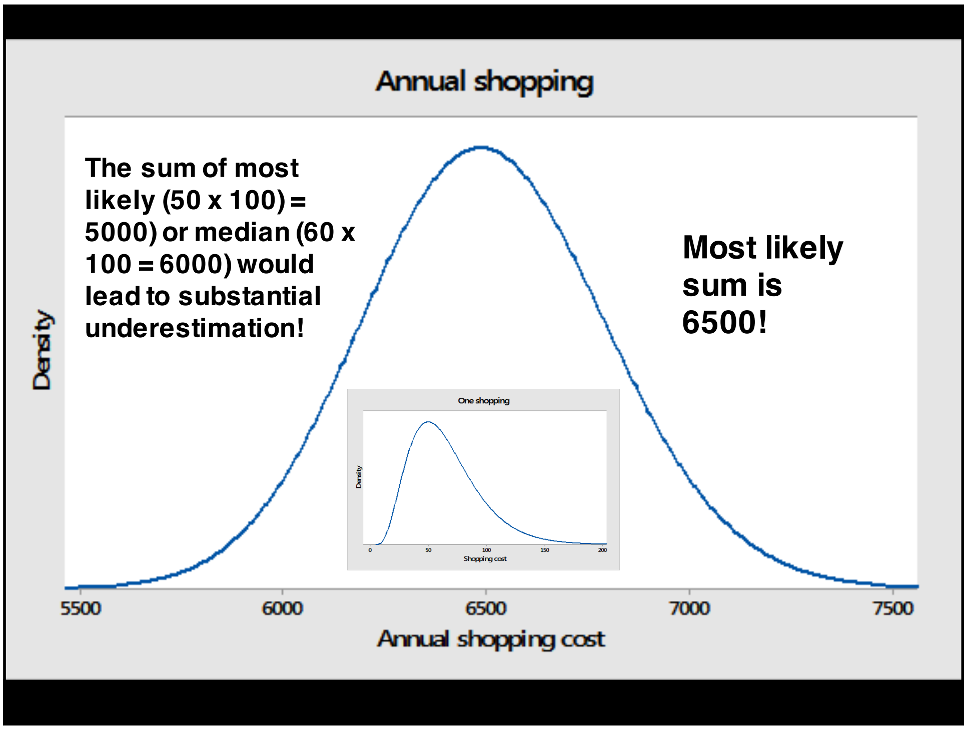
ADDING ESTIMATES IN A RIGHT-SKEWED WORLD



One shopping

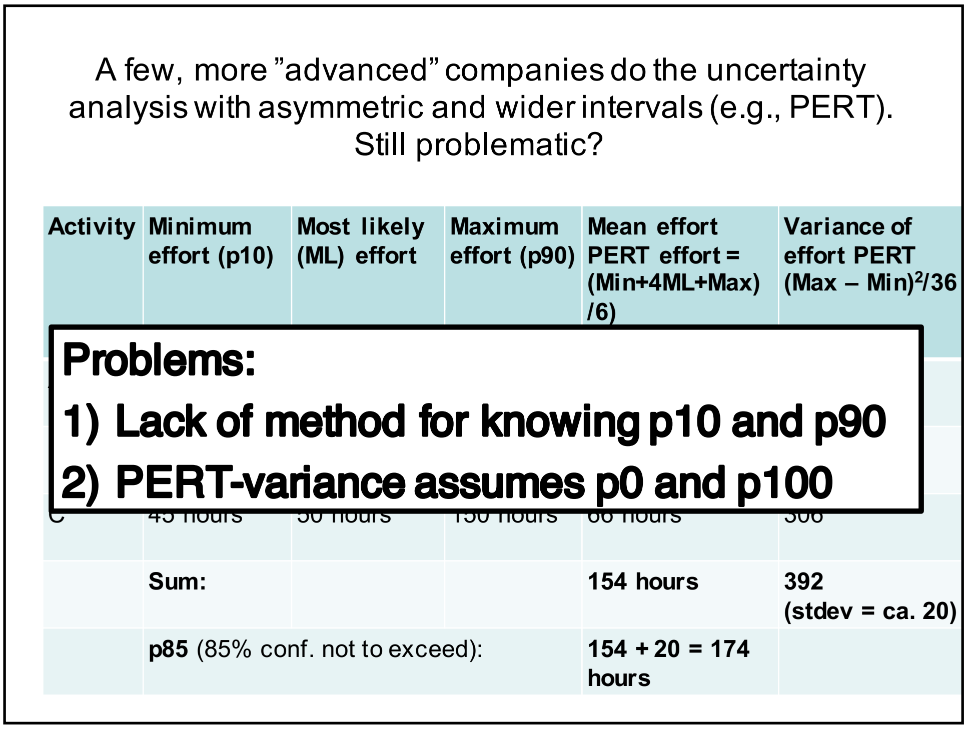
Most likely cost = 50
Median cost = 60
Mean cost = 65

What is the most likely cost (sum) of 100 times shopping?



The sum of most likely (50 x 100) = 5000) or median (60 x 100 = 6000) would lead to substantial underestimation!

Most likely sum is 6500!



A few, more ”advanced” companies do the uncertainty analysis with asymmetric and wider intervals (e.g., PERT). Still problematic?

Activity	Minimum effort (p10)	Most likely (ML) effort	Maximum effort (p90)	Mean effort PERT effort = (Min+4ML+Max) /6)	Variance of effort PERT (Max – Min)^2/36
C	45 hours	50 hours	150 hours	66 hours	306
	Sum:			**154 hours**	**392 (stdev = ca. 20)**
	p85 (85% conf. not to exceed):			**154 + 20 = 174 hours**	

Problems:

1) **Lack of method for knowing p10 and p90**
2) **PERT-variance assumes p0 and p100**

Cost uncertainty in software projects

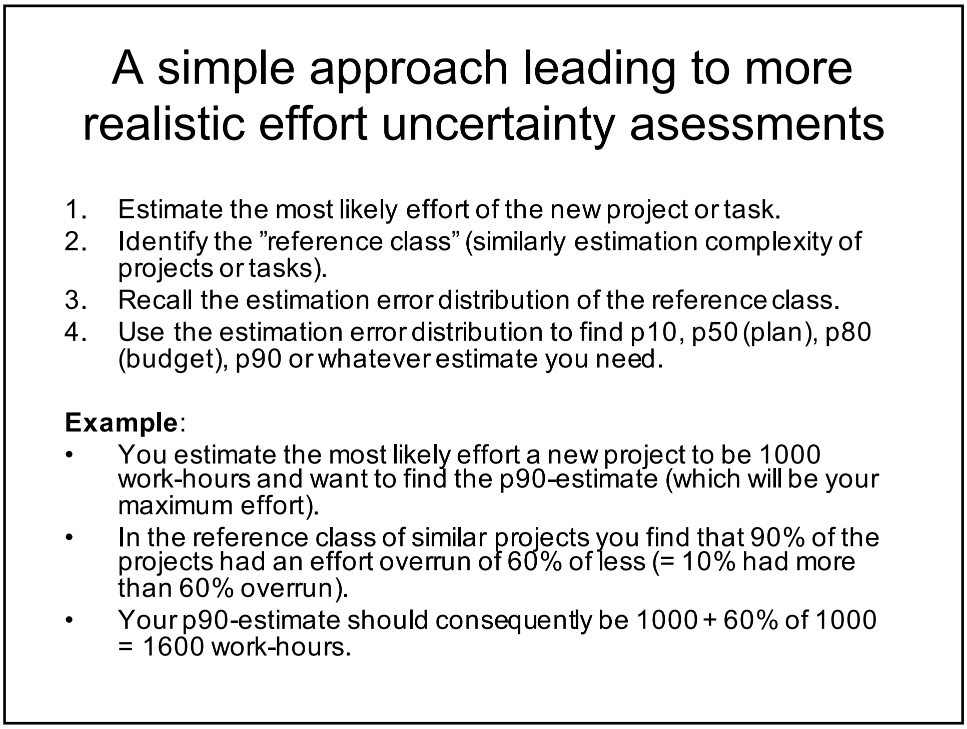
Types of project cost uncertainties:

- •Activity uncertainty
- •Known risks (manageable)
- •Unforeseen events and activities
- •Risk of total failure (chaos, re-definition)

My studies find that most software projects' cost estimation work:

- •Include no or insufficient analysis of activity uncertainty
- •Analyse known risks, but have no proper transfer of risk analysis to cost estimates
- •Include no analysis of unknown risks
- •Ignore the risk of total failure

What to do? A long way to go …

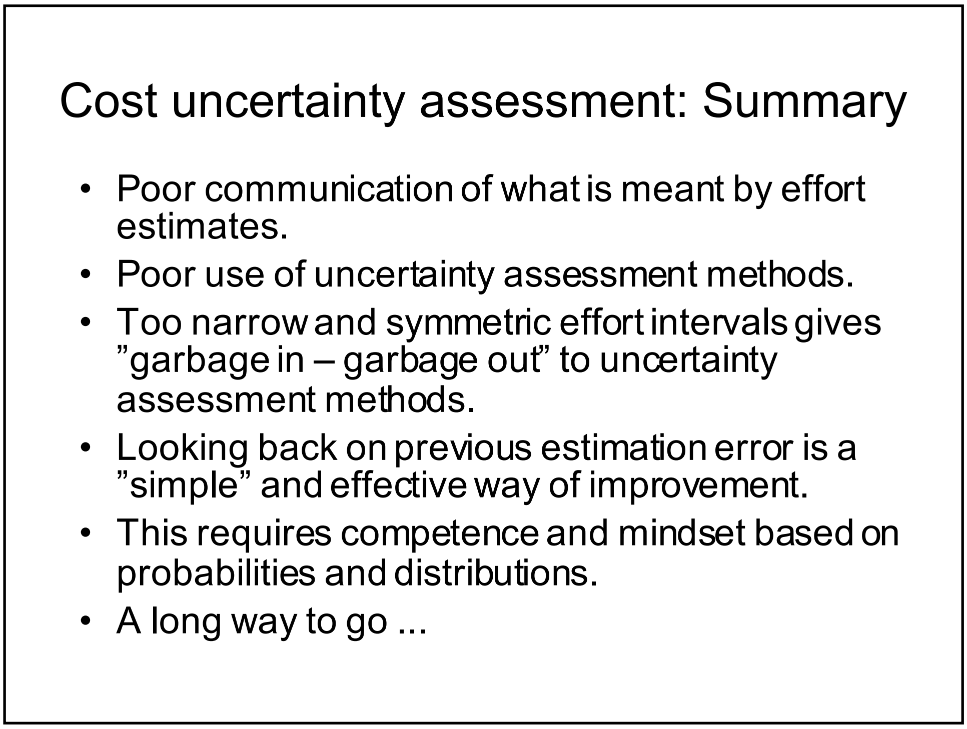


A simple approach leading to more realistic effort uncertainty asessments

1. Estimate the most likely effort of the new project or task.
2. Identify the ”reference class” (similarly estimation complexity of projects or tasks).
3. Recall the estimation error distribution of the reference class.
4. Use the estimation error distribution to find p10, p50 (plan), p80 (budget), p90 or whatever estimate you need.

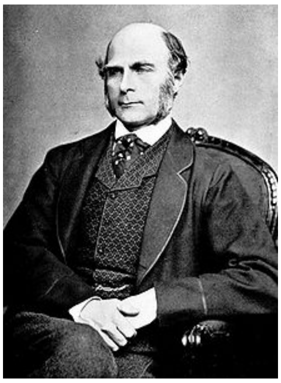
Example:

- You estimate the most likely effort a new project to be 1000 work-hours and want to find the p90-estimate (which will be your maximum effort).
- In the reference class of similar projects you find that 90% of the projects had an effort overrun of 60% of less (= 10% had more than 60% overrun).
- Your p90-estimate should consequently be 1000 + 60% of 1000 = 1600 work-hours.



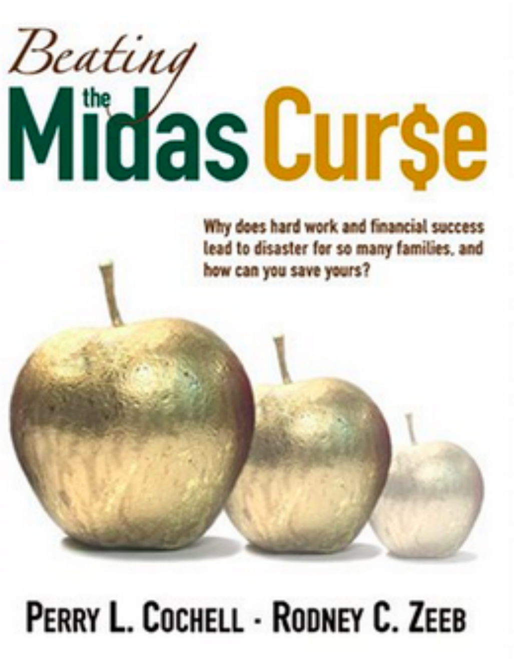
Cost uncertainty assessment: Summary

- Poor communication of what is meant by effort estimates.
- Poor use of uncertainty assessment methods.
- Too narrow and symmetric effort intervals gives ”garbage in – garbage out” to uncertainty assessment methods.
- Looking back on previous estimation error is a ”simple” and effective way of improvement.
- This requires competence and mindset based on probabilities and distributions.
- A long way to go ...

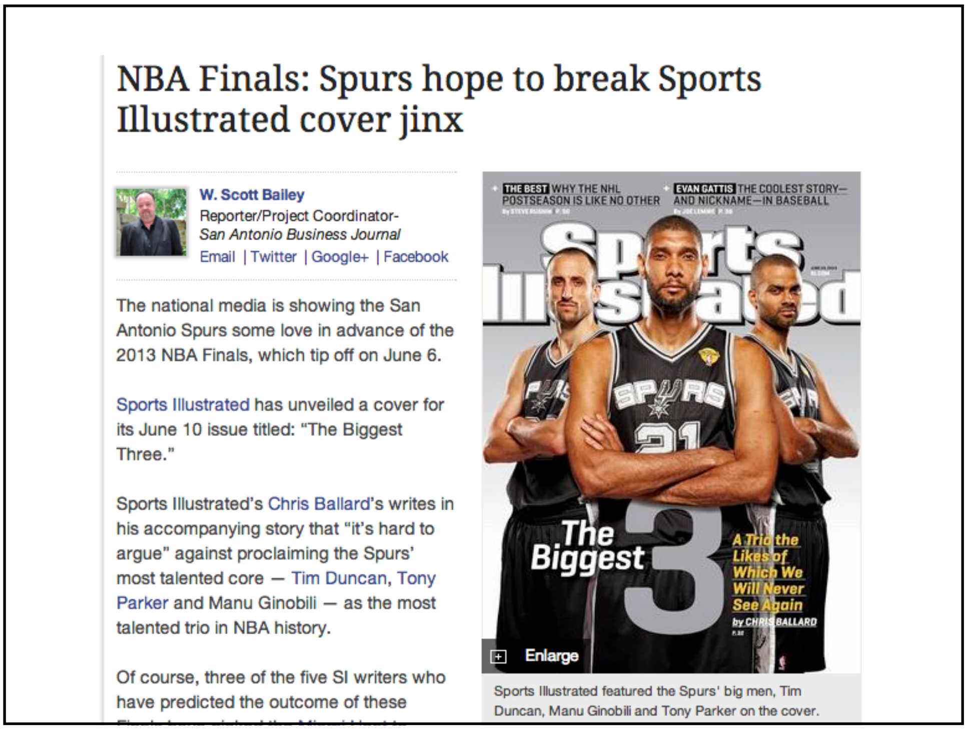


Selection bias

(winner's curse, optimizer's curse, regression towards the mean)


Beating the Midas Cur$e
Why does hard work and financial success lead to disaster for so many families, and how can you save yours?
PERRY L. COCHELL · RODNEY C. ZEEB

.... six out of ten affluent (rich) families will lose the family fortune by the end of the second generation.



NBA Finals: Spurs hope to break Sports Illustrated cover jinx

W. Scott Bailey
Reporter/Project Coordinator-
San Antonio Business Journal
Email | Twitter | Google+ | Facebook

The national media is showing the San Antonio Spurs some love in advance of the 2013 NBA Finals, which tip off on June 6.

Sports Illustrated has unveiled a cover for its June 10 issue titled: "The Biggest Three."

Sports Illustrated's Chris Ballard's writes in his accompanying story that "it's hard to argue" against proclaiming the Spurs' most talented core — Tim Duncan, Tony Parker and Manu Ginobili — as the most talented trio in NBA history.

Of course, three of the five SI writers who have predicted the outcome of these

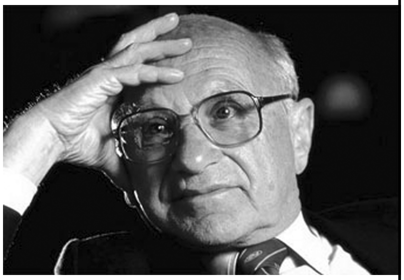
THE BEST WHY THE NHL POSTSEASON IS LIKE NO OTHER
EVAN GATTIS THE COOLEST STORY— AND NICKNAME—IN BASEBALL
Sports Illustrated
SPURS
21
The Biggest 3
A Trio the Likes of Which We Will Never See Again
by CHRIS BALLARD
Enlarge

Sports Illustrated featured the Spurs' big men, Tim Duncan, Manu Ginobili and Tony Parker on the cover.

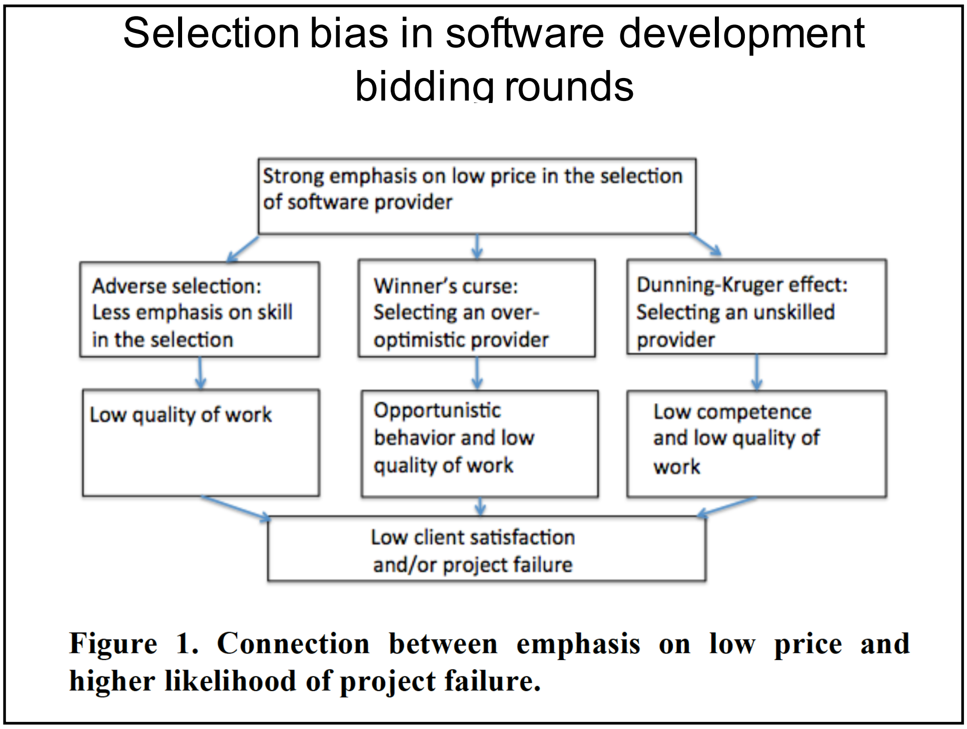
Analyses of non-random samples (self-selected, the best 20% on a test, the projects with highest cost overrun, the developers with lowest estimates, etc.), will easily be misleading.

The more extreme the sampling, the stronger the effect of regression effects.

"I suspect that the regression fallacy is the most common fallacy in the statistical analysis of economic data"



Milton Friedman (Nobel prize winner in economy)



Selection bias in software development bidding rounds

Strong emphasis on low price in the selection of software provider

Adverse selection: Less emphasis on skill in the selection

Winner's curse: Selecting an over-optimistic provider

Dunning-Kruger effect: Selecting an unskilled provider

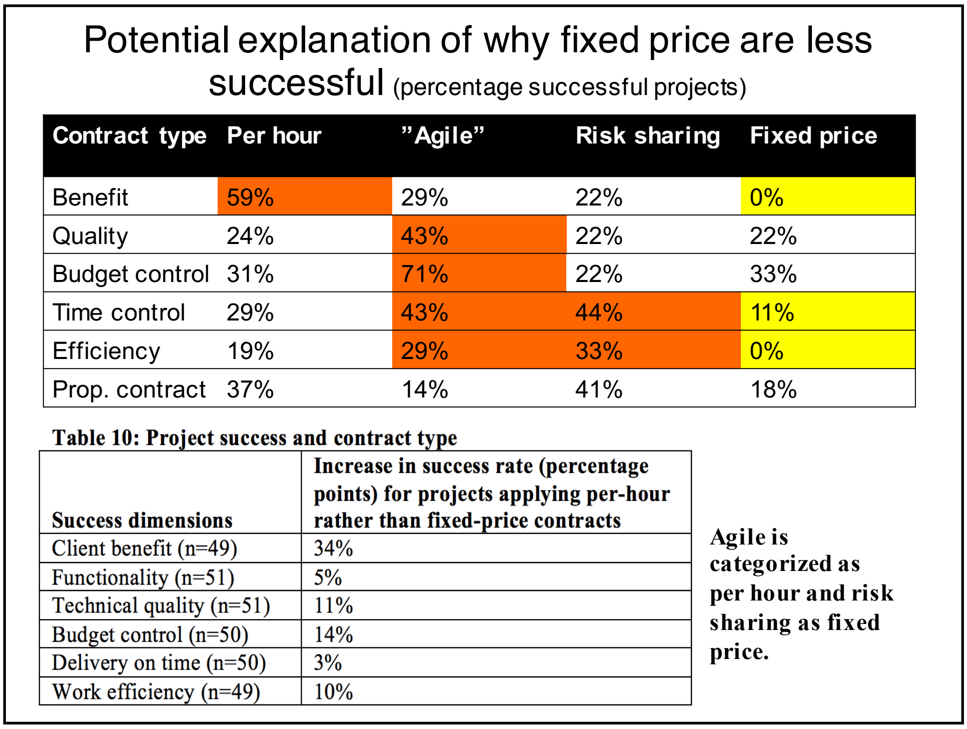
Low quality of work

Opportunistic behavior and low quality of work

Low competence and low quality of work

Low client satisfaction and/or project failure

Figure 1. Connection between emphasis on low price and higher likelihood of project failure.



Potential explanation of why fixed price are less successful (percentage successful projects)

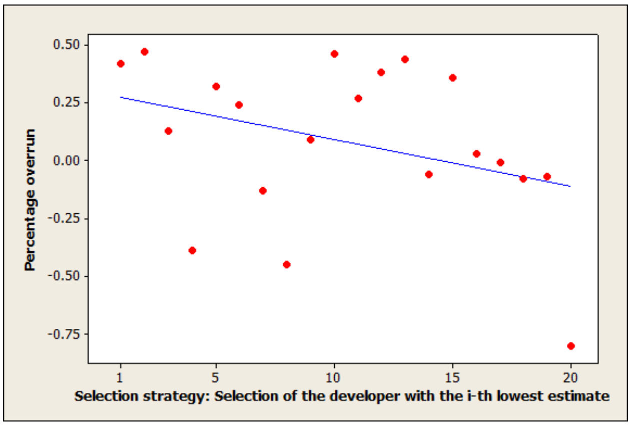
Contract type	Per hour	"Agile"	Risk sharing	Fixed price
Benefit	59%	29%	22%	0%
Quality	24%	43%	22%	22%
Budget control	31%	71%	22%	33%
Time control	29%	43%	44%	11%
Efficiency	19%	29%	33%	0%
Prop. contract	37%	14%	41%	18%

Table 10: Project success and contract type

Success dimensions	**Increase in success rate (percentage points) for projects applying per-hour rather than fixed-price contracts**
Client benefit (n=49)	34%
Functionality (n=51)	5%
Technical quality (n=51)	11%
Budget control (n=50)	14%
Delivery on time (n=50)	3%
Work efficiency (n=49)	10%

Agile is categorized as per hour and risk sharing as fixed price.

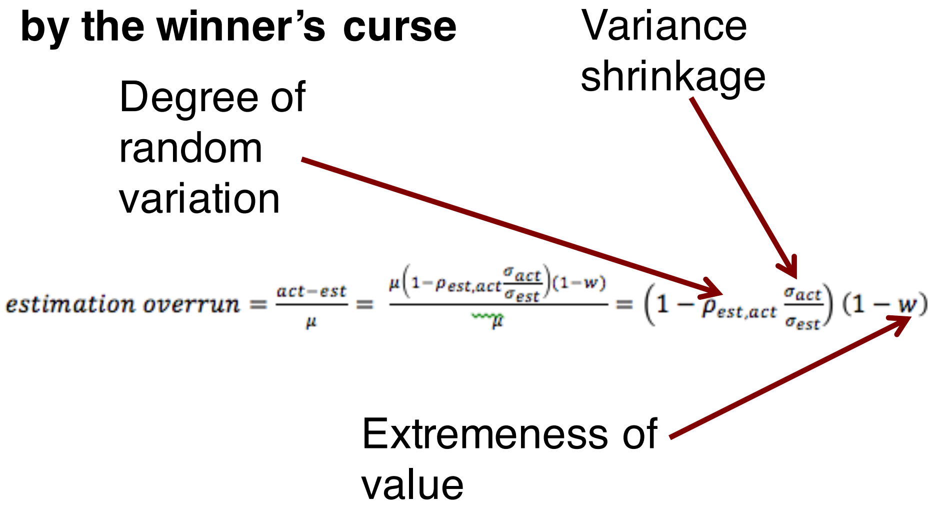
The lower the selected effort estimate, the higher the risk of effort overrun (the winner’s curse)



Study:
20 developers estimating and completing the same five tasks

M. Jørgensen. The Influence of Selection Bias on Effort Overruns in Software Development Projects, Information and Software Technology 55(9):1640-1650, 2013.

The size of expected cost overrun caused by the winner’s curse



Variance shrinkage

Degree of random variation

$$estimation\ overrun = \frac{act-est}{\mu} = \frac{\mu\left(1-\rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w)}{\mu} = \left(1-\rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w)$$

Extremeness of value

M. Jørgensen. The Influence of Selection Bias on Effort Overruns in Software Development Projects, Information and Software Technology 55(9):1640-1650, 2013.

M. Jørgensen, A Strong Focus on Low Price When Selecting Software Providers Increases the Likelihood of Failure in Software Outsourcing Projects, EASE, Porto de Galinhas, 2013

Selection of provider with bid less than 25% of the average bid, led to a 9% increase in project failure **for the same level of provider competence.**

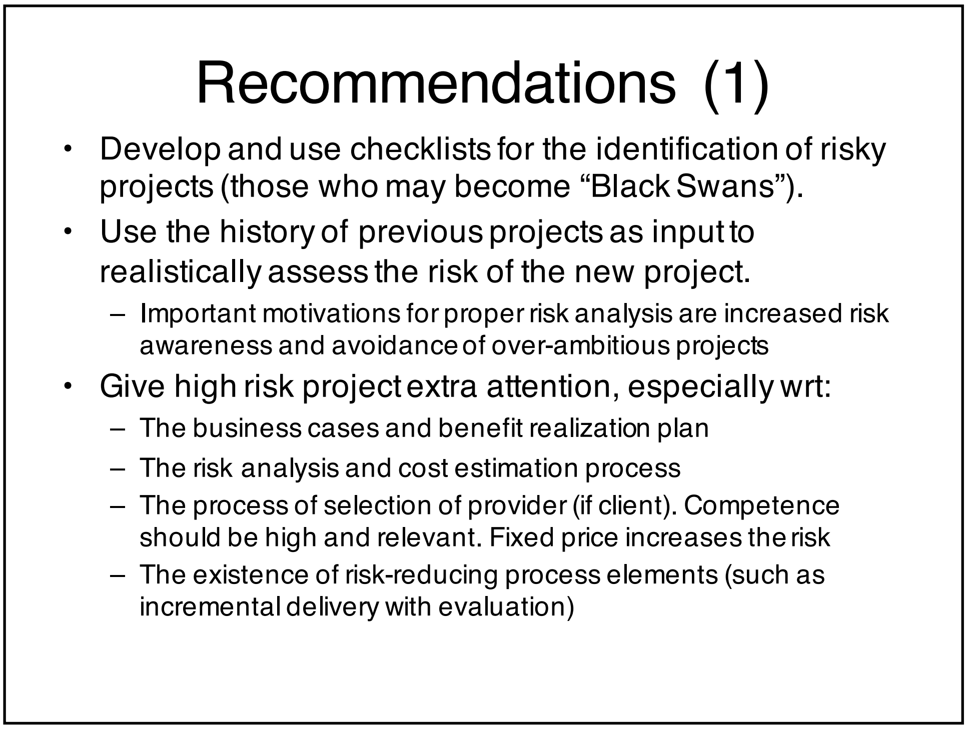
Analysis of nearly 800.000 small projects at VWorker.com

Winner's curse leads to client's curse

Failure = Cancelled or completed with client satisfaction of "poor" or worse.

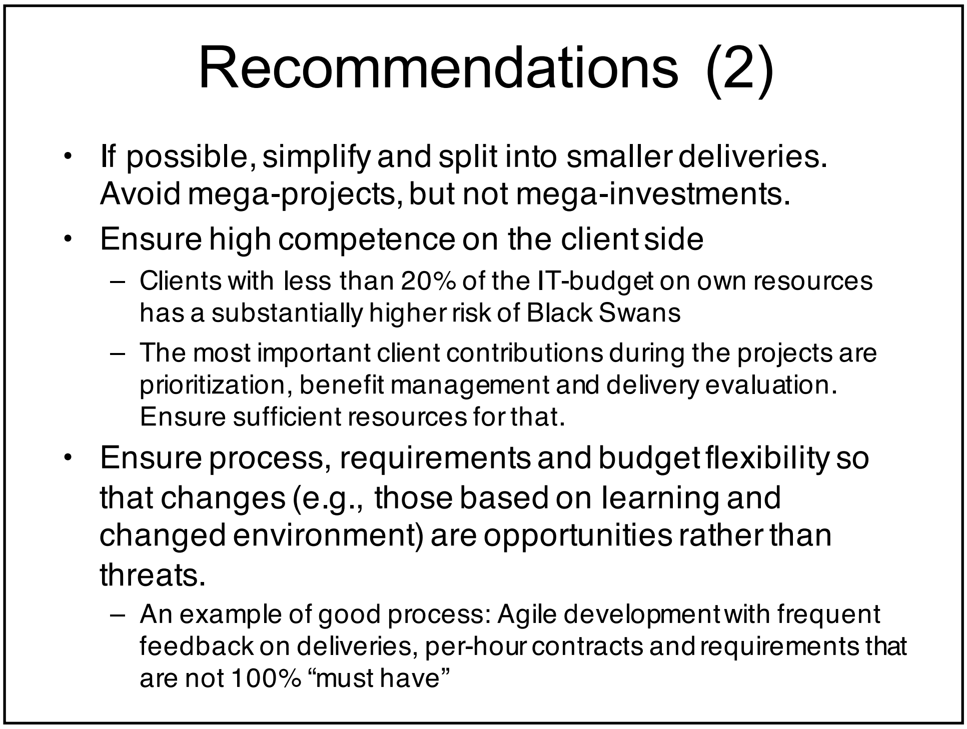
Competence = Average client satisfaction + failure frequency.

WHAT TO DO?



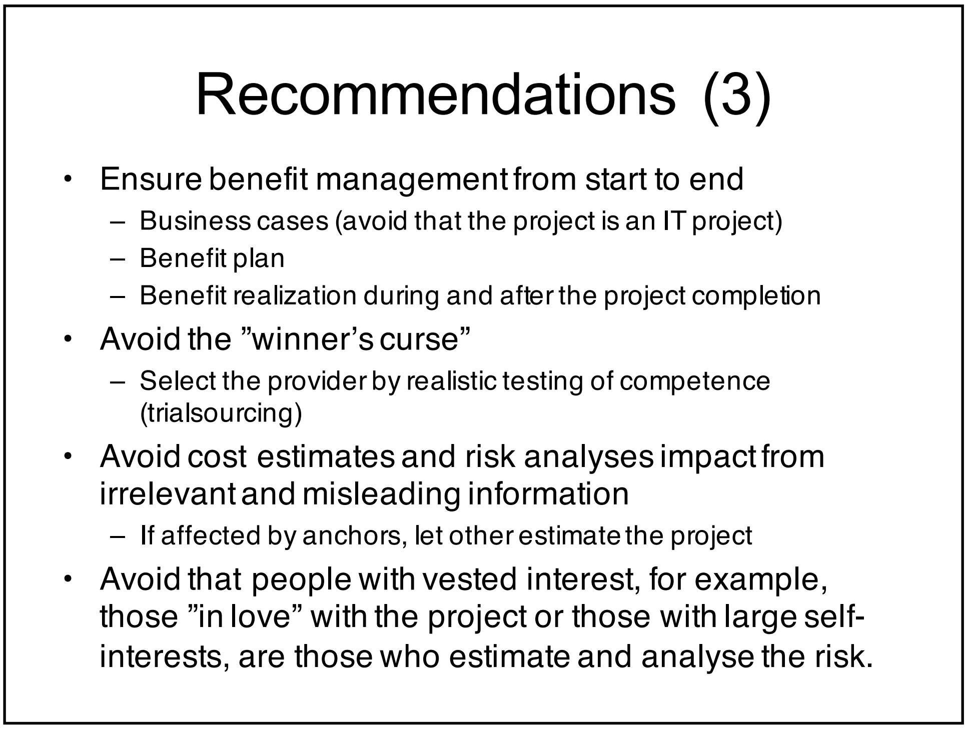
Recommendations (1)

- Develop and use checklists for the identification of risky projects (those who may become “Black Swans”).
- Use the history of previous projects as input to realistically assess the risk of the new project.
  - Important motivations for proper risk analysis are increased risk awareness and avoidance of over-ambitious projects
- Give high risk project extra attention, especially wrt:
  - The business cases and benefit realization plan
  - The risk analysis and cost estimation process
  - The process of selection of provider (if client). Competence should be high and relevant. Fixed price increases the risk
  - The existence of risk-reducing process elements (such as incremental delivery with evaluation)



Recommendations (2)

- If possible, simplify and split into smaller deliveries. Avoid mega-projects, but not mega-investments.
- Ensure high competence on the client side
  - Clients with less than 20% of the IT-budget on own resources has a substantially higher risk of Black Swans
  - The most important client contributions during the projects are prioritization, benefit management and delivery evaluation. Ensure sufficient resources for that.
- Ensure process, requirements and budget flexibility so that changes (e.g., those based on learning and changed environment) are opportunities rather than threats.
  - An example of good process: Agile development with frequent feedback on deliveries, per-hour contracts and requirements that are not 100% “must have”



Recommendations (3)

- Ensure benefit management from start to end
  - Business cases (avoid that the project is an IT project)
  - Benefit plan
  - Benefit realization during and after the project completion
- Avoid the ”winner’s curse”
  - Select the provider by realistic testing of competence (trialsourcing)
- Avoid cost estimates and risk analyses impact from irrelevant and misleading information
  - If affected by anchors, let other estimate the project
- Avoid that people with vested interest, for example, those ”in love” with the project or those with large self-interests, are those who estimate and analyse the risk.