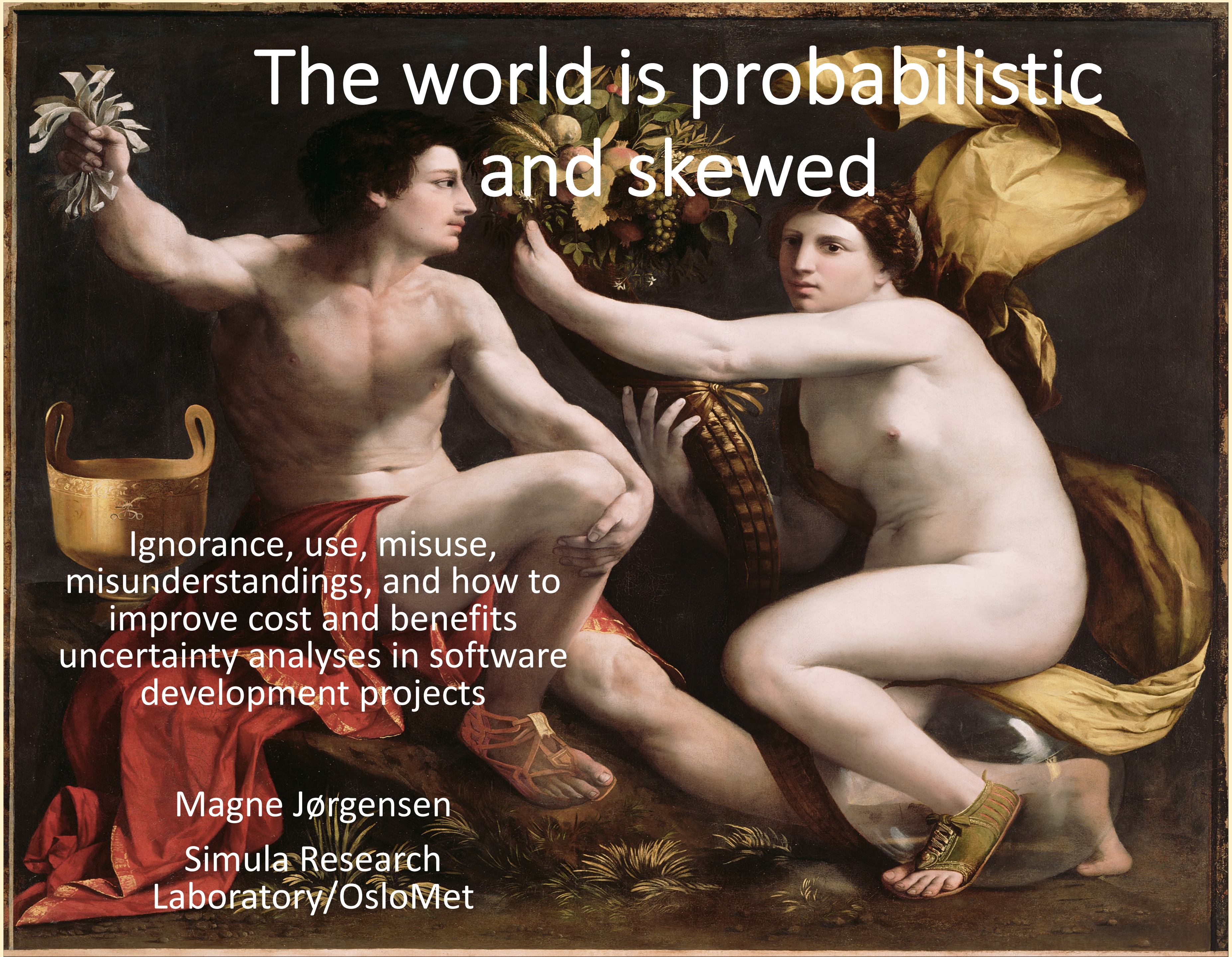


The world is probabilistic and skewed

Ignorance, use, misuse, misunderstandings, and how to improve cost and benefits uncertainty analyses in software development projects

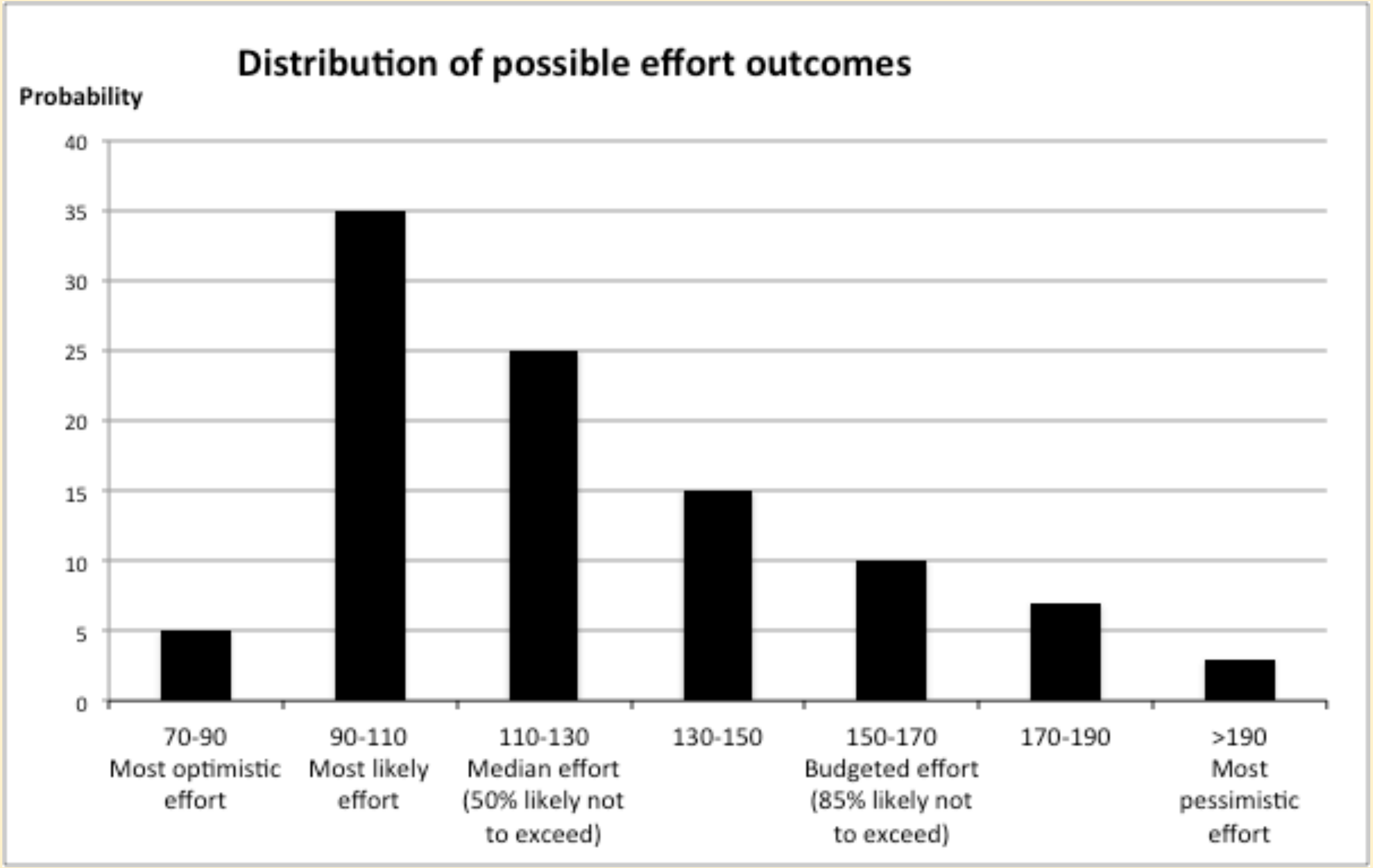
Magne Jørgensen

Simula Research Laboratory/OsloMet

Does the software industry know and communicate what they mean with an effort estimate?

(Do **you** know what an estimate is?)

An estimate is an estimate is an estimate?



Estimate =
Best case?
Most likely?
Median?
With contingency added?
Something else?

A proper communication of what we mean with an estimate requires a probabilistic understanding!

It's fine to give a single point estimate, as long as we tell where at the distribution we are, e.g., that we communicate a most likely or p50-estimate (median estimate).

A survey among software professionals

“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 no or almost no major problems.* [Ideal effort]
- *Number of work-hours I most likely will use.* [Most likely effort]
- *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.* [Median effort/p50]
- *Number of work-hours where it is unlikely that I will use more effort than estimated.* [Risk averse effort/budgeted effort/…]
- *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.* [Don’t know/gut feeling]
- *None of the above descriptions is close to what I typically mean by an effort estimate.*

Results (replicated in other surveys)

Interpretation (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%

Similar problems (probably worse) with estimates of benefits ...

I analysed more than 100 cost-benefit plans of Norwegian IT projects. **None** were explicit about what they meant with their estimated benefits or estimated profit.

Sometimes software companies try to include uncertainty in their estimates.

Some provide and add uncertainty as shown below
Exercise: Find (at least) five problems

Activity	Minimum effort (best case, optimistic)	Estimate	Maximum effort (worst case, pessimistic)
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. (For benefits, which may be left-skewed (?), this may lead to over-estimation.)
5. **No dependencies**. Most projects have dependencies between activities, e.g., testing is 40% of development. Not including this, leads to even more underestimation.

A brief side-track on adding estimates in a right-skewed world

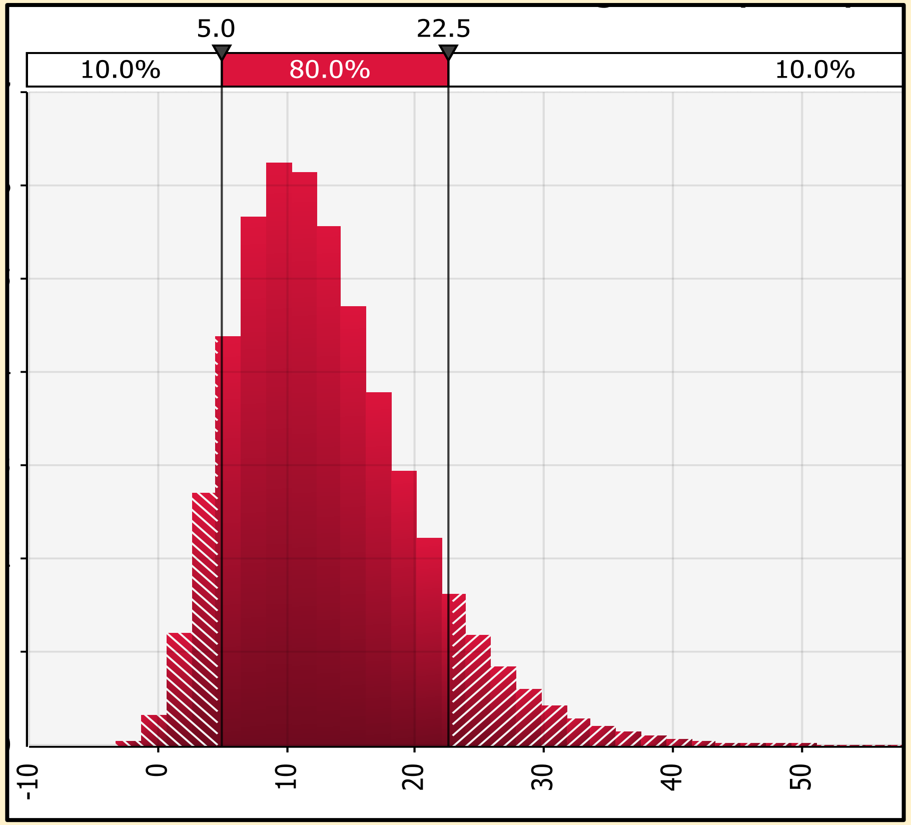
Assume project X

- Ten user stories, where all have the same (right-skewed) effort outcome distribution
  - Minimum (p10): 5 hour
  - **Most likely: 10 hours**
  - Maximum (p90): 22.5 hours

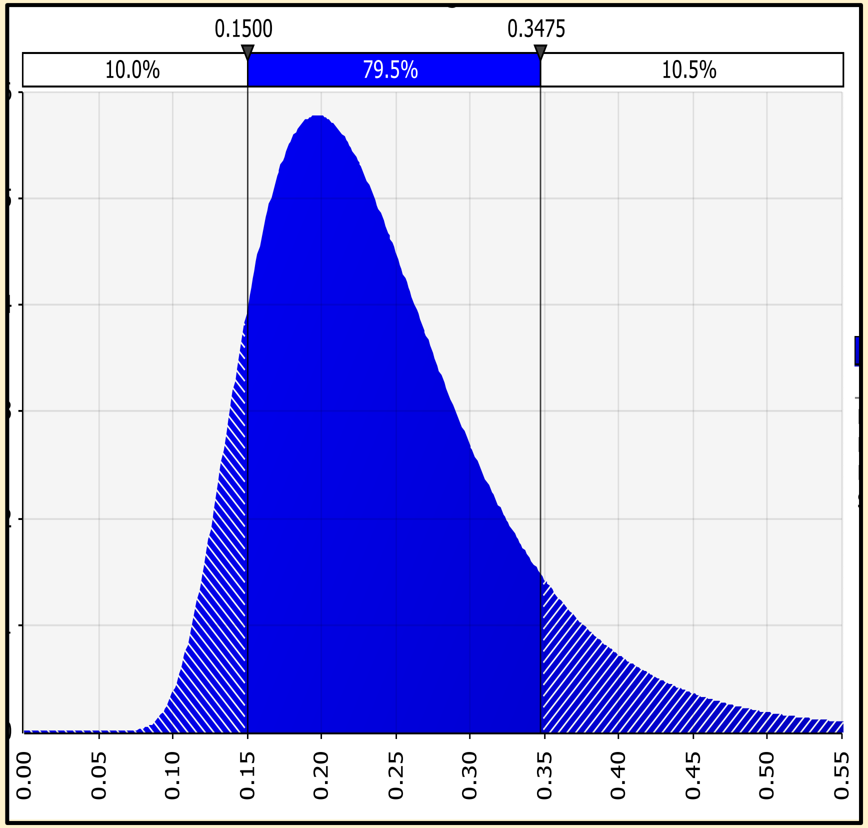
- Add-on activities (dependencies): 5 activities calculated as proportions of the sum of the ten user stories (administration, system test, ...). All of them have the same right-skewed effort outcome distribution
  - Minimum (p10): 15% of the effort on the user stories
  - **Most likely: 20% of the effort on the user stories**
  - Maximum (p90): 35% of the effort on the user stories

The effort distributions (log-normal, right-skewed)

A user story



An add-on activity

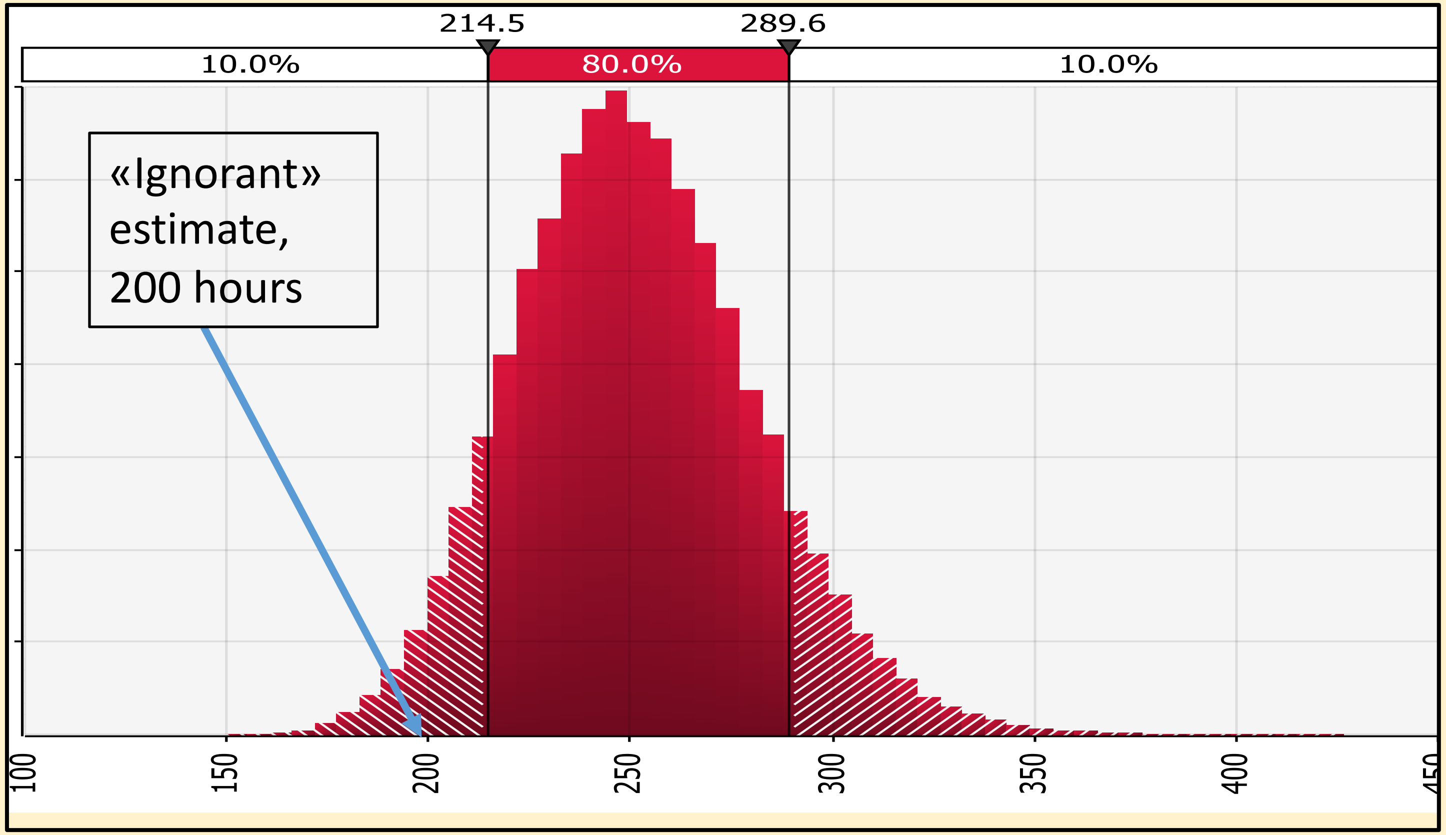


«Ignorant» adding of activities

- SUM User stories
  - 10 x most likely (10 hours) = 100 hours
- SUM Add-on activities
  - 5 x 0.2 x SUM User stories = 100 hours
- SUM TOTAL
  - 100 tv + 100 tv = 200 hours
- Gives a STRONG underestimation!
- **Very unlikely to use 200 hours or less!**

The sum of the most likely effort is NOT the most likely sum

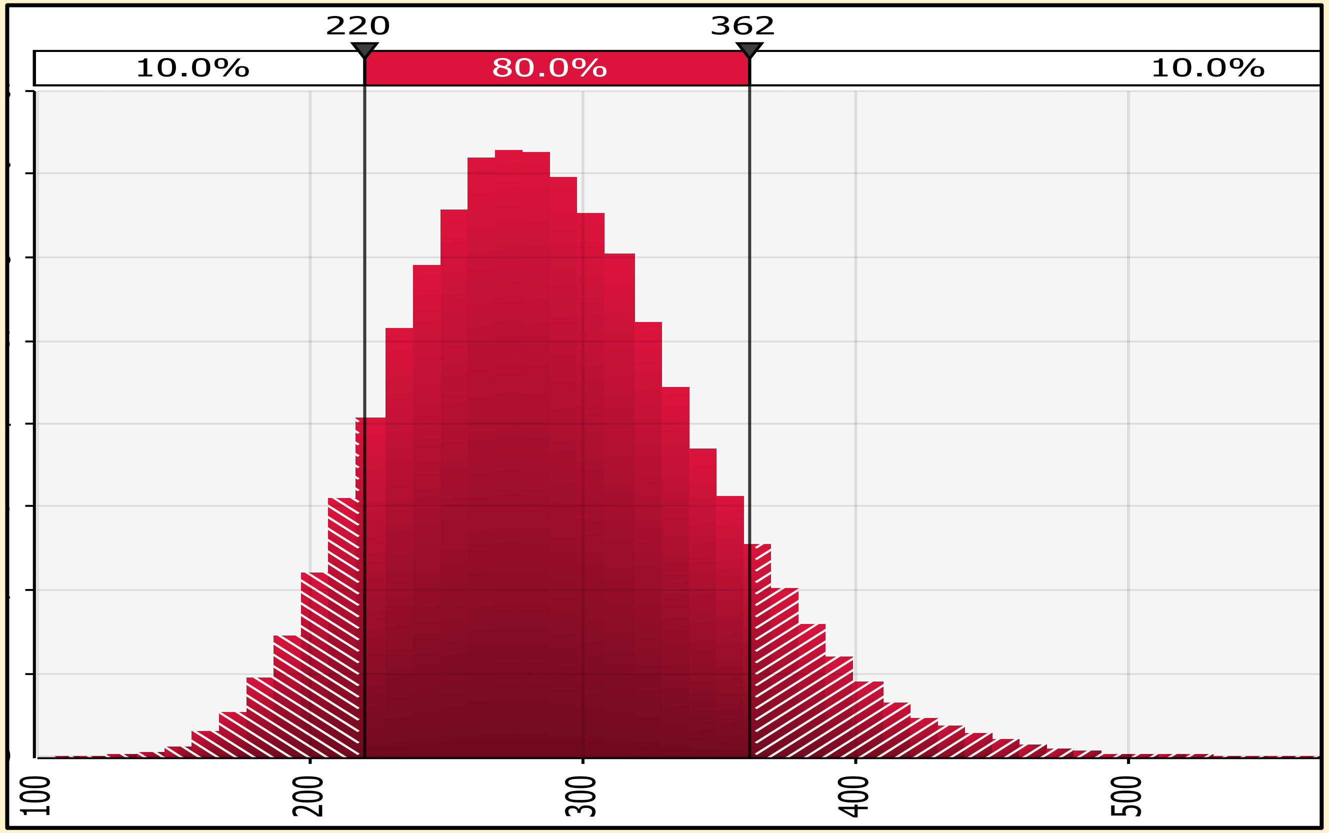
Sum of the effort distributions



P10 = 215
Most likely = 246
P50 (median) = 250
Expected value (mean) = 251
P85 = 282
P90 = 290

Still not optimal, not right-skewed.
Central limit theorem + no dependencies → symmetric

Sum of effort distributions with dependencies (add-on dep.)



P10 = 220 (up from 215)
Most likely = 265 (up from 246)
P50 (median) = 284 (up from 250)
Expected value (mean) = 288 (up from 251)
P85 = 345 (up from 282)
P90 = 361 (up frok 290)

Large increase in most likely estimates, nd it is right-skewed.
Median estimate (50% likely) is 284 (vs. “ignorant“ of 200 and “no dependencies“ of 250)

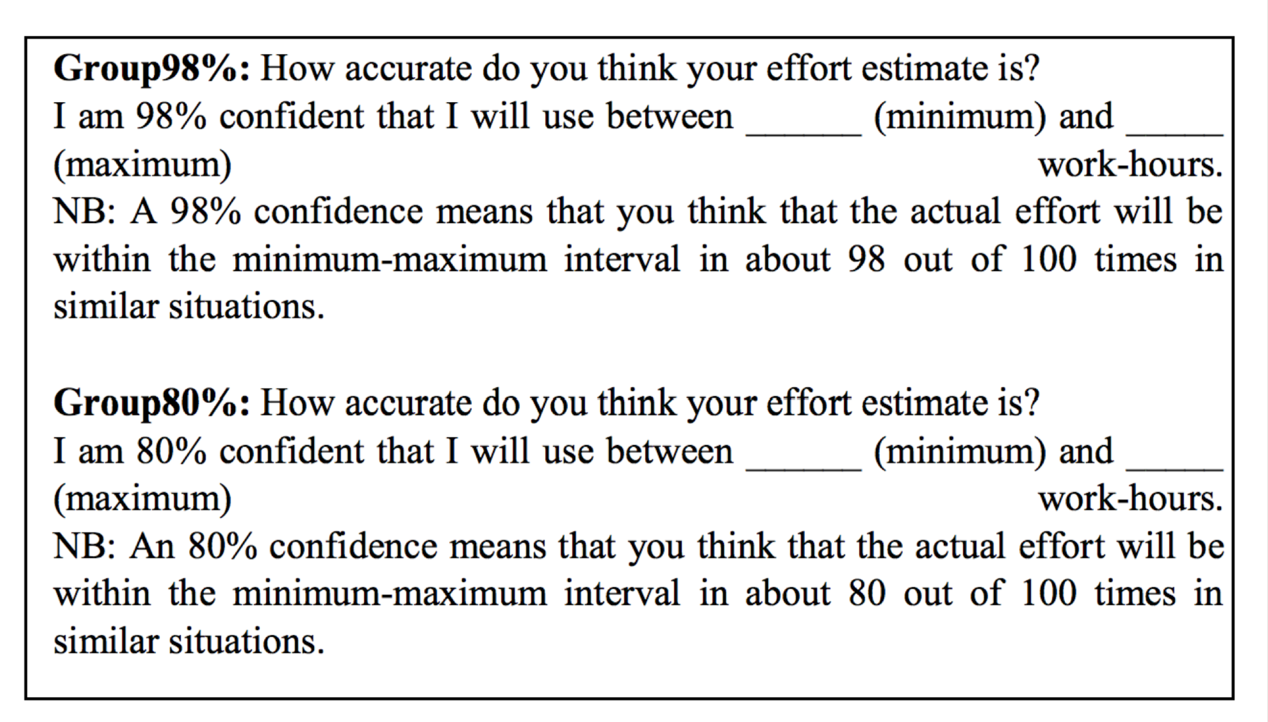
The most ”advanced” companies do it with asymmetric and wider intervals, and the use of the ”PERT”-formula. Still problematic?

Activity	**Minimum effort (p10)**	**Most likely (ML) effort**	**Maximum effort (p90)**	**Mean effort** **PERT effort = (Min+4ML+Max)/6)**	**Variance of effort** **PERT variance = (Max – Min)^2/36**
Activity A	15 work-hours	20 work-hours	40 work-hours	23 work-hours	17
Activity B	50 work-hours	60 work-hours	100 work-hours	65 work-hours	69
Activity C	45 work-hours	50 work-hours	150 work-hours	66 work-hours	306
Sum			Expected value =	**154 work-hours**	**392 (stdev = 20)**
Uncertainty	p85 (85% conf. not to exceed) equals ca. exp. value + stdev			**154 + 20 = 174 wh**	

- The assumption of the PERT-formula is the unrealistic assumption that min=p0 and max=p100. Does not affect mean effort much, but the variance get much too small. Should divide variance (assuming p10 as min and p90 as max) by approx. 2.65^2= 7.0 instead of 36! PERT gives much too narrow intervals.
- No support for knowing what a p10 and p90 estimate should be (No diff betwen 75%, 80%, 90% and 98% confidence intervals.)

Experiment: Are software developers and managers able to give min-max with X% confidence?

- Participants: 62 professional software developers (from Ukraine)
- Estimated the same task.
- Asked for min-max intervals



Group98%: How accurate do you think your effort estimate is?
I am 98% confident that I will use between ______ (minimum) and _____ (maximum) work-hours.
NB: A 98% confidence means that you think that the actual effort will be within the minimum-maximum interval in about 98 out of 100 times in similar situations.

Group80%: How accurate do you think your effort estimate is?
I am 80% confident that I will use between ______ (minimum) and _____ (maximum) work-hours.
NB: An 80% confidence means that you think that the actual effort will be within the minimum-maximum interval in about 80 out of 100 times in similar situations.

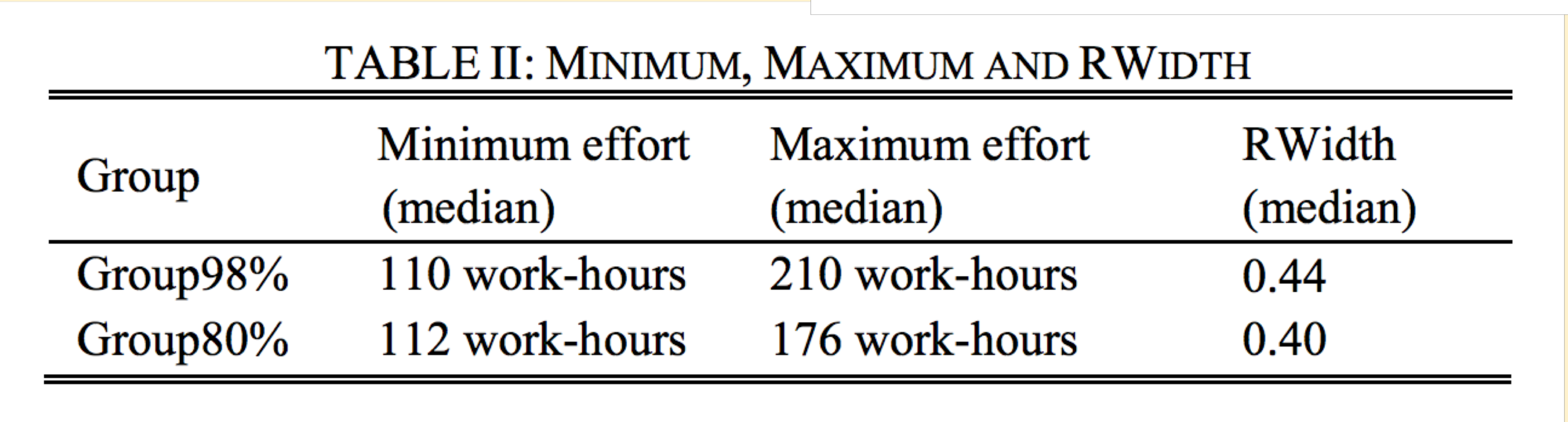


TABLE II: MINIMUM, MAXIMUM AND RWIDTH

Group	Minimum effort (median)	Maximum effort (median)	RWidth (median)
Group98%	110 work-hours	210 work-hours	0.44
Group80%	112 work-hours	176 work-hours	0.40

Rwidth =
(Max – Min)/Most likely

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/know 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.

We have evaluated and implemented this approach in real-world contexts

Experiment

- Nineteen estimation teams of software professionals within one company.
- Estimation of the most likely effort of a project, which had just started.
- Estimation of the uncertainty in terms of 90%-confidence intervals (p5 and p95).
- Two groups:
  - Group A: Uncertainty assessment ”as usual”. Give 90% prediction intervals. No support for minimum and maximum judgements.
  - Group B: Create the error distribution of the reference class. Provide minimum and maximum effort.
- **Results**: The teams in Group B had much more realistic views of the real uncertainty of the project. Especially for the minimum effort, understanding that the world is right-skewed.
- Two replications in real-world contexts (controlled field experiments) confirm the results of improved realism using this method.

So what …

- Poor communication of what is meant by effort and benefits estimates is typical in software estimation contexts.
- Poor use of uncertainty assessment methods, if used at all, is even more common.
- Too narrow and too symmetric effort intervals gives ”garbage in – garbage out” even when using proper uncertainty assessment methods.
- Looking back on previous estimation error is a ”simple” and effective way of getting realistic effort prediction intervals.
- This requires compentence and mindsets based on probabilities and distributions.
- A long way to go before the IT industry are able to identify the real uncertainty of software projects...