# Avoid software project horror stories

# Check the reality value of the estimate first!

**Harold van Heeringen**
**ICEAA 2014**

nesma READY TO CONNECT SOFTWARE METRICS

## Harold van Heeringen

**NESMA board member**
Nederlandse Software Metrieken Association
www.nesma.org

**President ISBSG**
International Software Benchmarking Standards Group
www.isbsg.org

**IAC member COSMIC**
Common Software Measurement International Consortium
www.cosmicon.com

**Senior Software Cost Engineer,**
Sogeti Nederland B.V.
www.sogeti.com

nesma READY TO CONNECT SOFTWARE METRICS

## Questions for our session today

Software project horror stories – what are they?

The software industry – when will it mature?

Experts – what's wrong with them?

Why should we do parametric software estimates?

How can we assess the reality value of an estimate?

What should we do to avoid horror stories, or at least decrease the risk of these to happen?

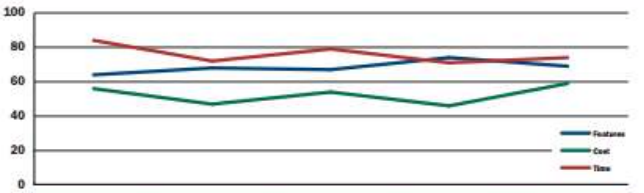
nesma READY TO CONNECT SOFTWARE METRICS

## Software industry

| | 2004 | 2006 | 2008 | 2010 | 2012 |
|---|---|---|---|---|---|
| Successful | 29% | 35% | 32% | 37% | 39% |
| Failed | 18% | 19% | 24% | 21% | 18% |
| Challenged | 53% | 46% | 44% | 42% | 43% |

Project resolution results from CHAOS research for years 2004 to 2012.

Time and cost overruns, plus percentage of features delivered from CHAOS research for the years 2004 to 2012.




| | 2004 | 2006 | 2008 | 2010 | 2012 |
|---|---|---|---|---|---|
| TIME | 84% | 72% | 79% | 71% | 74% |
| COST | 56% | 47% | 54% | 46% | 59% |
| FEATURES | 64% | 68% | 67% | 74% | 69% |

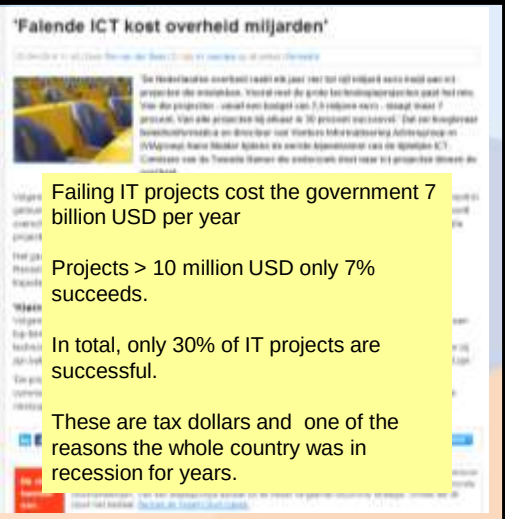
nesma READY TO CONNECT SOFTWARE METRICS

## Impact

Deliver too late: losing business.

Fail/stop: loss of time, money, business and still no solution for the problem that needed to be solved.

Waste of resources that could have been deployed successfully otherwise.




Failing IT projects cost the government 7 billion USD per year

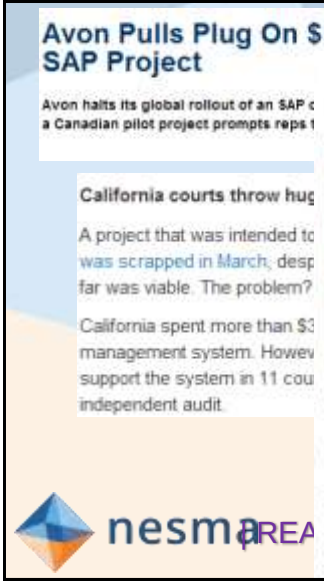
Projects > 10 million USD only 7% succeeds.

In total, only 30% of IT projects are successful.

These are tax dollars and one of the reasons the whole country was in recession for years.

nesma READY TO CONNECT SOFTWARE METRICS

## Horror stories





nesma REA

## Two main reasons

**Unstable user requirements**

- Starting the development too early in the project
- Not enough time spent on requirements analysis
- Users not involved or not involved enough

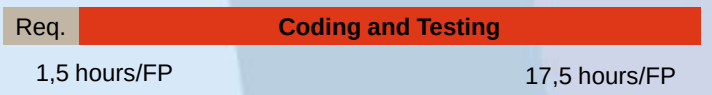
**Unrealistic project expectations**

- Usually: only expert estimates (optimistic)
- Pressure to lower cost and deliver faster
- End date is not estimated, but a given
- Duration is an important cost driver!

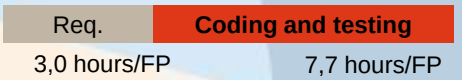
nesma READY TO CONNECT SOFTWARE METRICS

## Requirements

**Worst in class software development organizations spend 7,5% of the project budget on requirements**




**Best in class software development organizations spend 28% of the project budget on requirements**




**More effort spent on requirements increases project success!**

nesma READY TO CONNECT SOFTWARE METRICS

## Unrealistic expectations

Software project industry: low maturity
- Low estimation maturity;
- No or little formal estimation processes;
- No or little use of historical data;
- Customers choose suppliers based on price, not reality.

No reality check before finalizing an estimate! So, many unrealistically estimated projects actually start!

Results:
- Many failing projects
- Low customer satisfaction rates

nesma READY TO CONNECT SOFTWARE METRICS

## Why do we need realistic estimates?

A realistic estimate is one of the most important **conditions** for a successful project.

The estimate is the **basis** for:
- Business case;
- Planning;
- Proposal (outsourcing: fixed price / date);
- Financial result of the project… and the organization;
- Claiming and releasing of resources;
- Alignment between IT and business / customer;
- Progress reports / dashboards;
- The feeling of the team and the stakeholder.

Without a realistic estimate, the project is **likely to fail**!

nesma READY TO CONNECT SOFTWARE METRICS

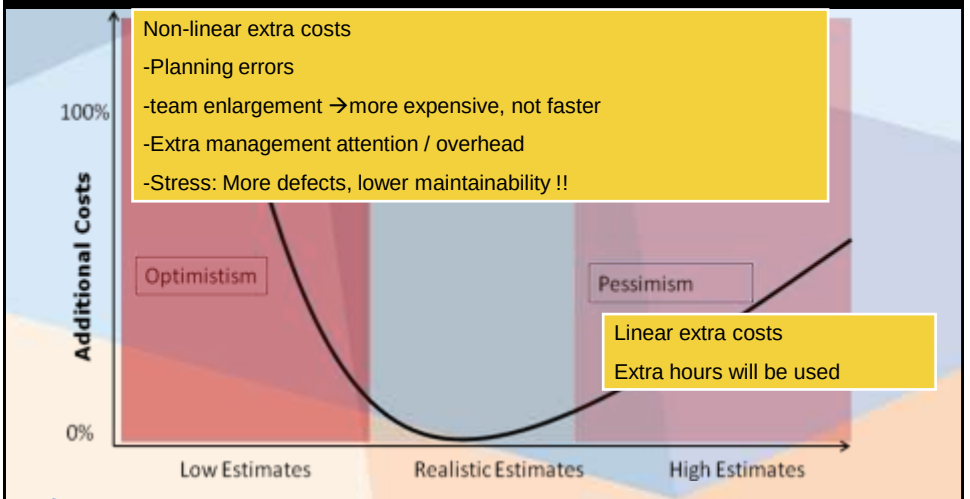
## Software estimation is hard

It's hard to accurately estimate software projects:

- Software is hard to measure, because intangible;
- Technical environments change all the time;
- Software companies are not mature enough to measure performance and store the metrics of completed projects;
- Software companies don't use data of completed projects in new estimates;
- The estimate often has to be finalized before the requirements are fully known;
- It's hard to estimate whether requirements will change, how much they will change, to factor this in the estimate and to explain this.
- It's hard to estimate which technical challenged have to be solved during the project;
- …

nesma READY TO CONNECT SOFTWARE METRICS

## The effect of underestimation




nesma READY TO CONNECT SOFTWARE METRICS

## Two types of project estimation

Two types of software estimation
- Expert estimate (Bottom-up / WBS)
- Parametric (Top-down / methodical)

Expert Estimate
- Technical specialists
- Bottom-up, effort estimate for activities identified (WBS)

Parametric estimate
- Based on size and historical data
- Use of Parametric models

nesma READY TO CONNECT SOFTWARE METRICS

## Expert estimates

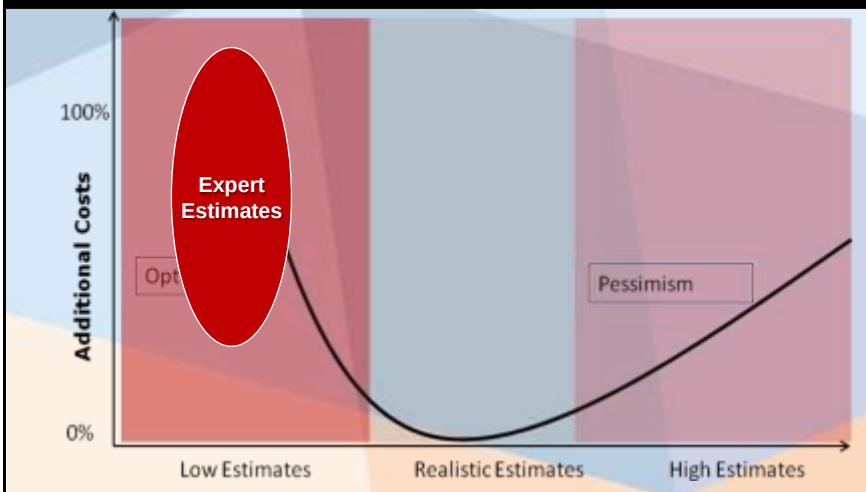
**Bottom-up , assign effort hours to work items, based on knowledge and experience**

**Result: expert estimates are optimistic, on average 30% underestimation.**

**Disadvantages:**
- Forgotten activities (testscript reviews. …);
- No good foundation of the estimate, very subjective;
- The expert is not going to de all the work (who will ?);
- How expert is the expert? (projects are unique);
- Experts don't take into account duration, team size, etc.;
- Experts don't assess the reality value, no real use of history.

nesma READY TO CONNECT SOFTWARE METRICS

## The effect of underestimation




nesma READY TO CONNECT SOFTWARE METRICS

## Parametric estimates

**Top-down, based on size, historical data and parametric models**

**Advantages:**
- Objective, repeatable, verifiable, risk;
- Scenario analysis: duration, team size, % confidence.

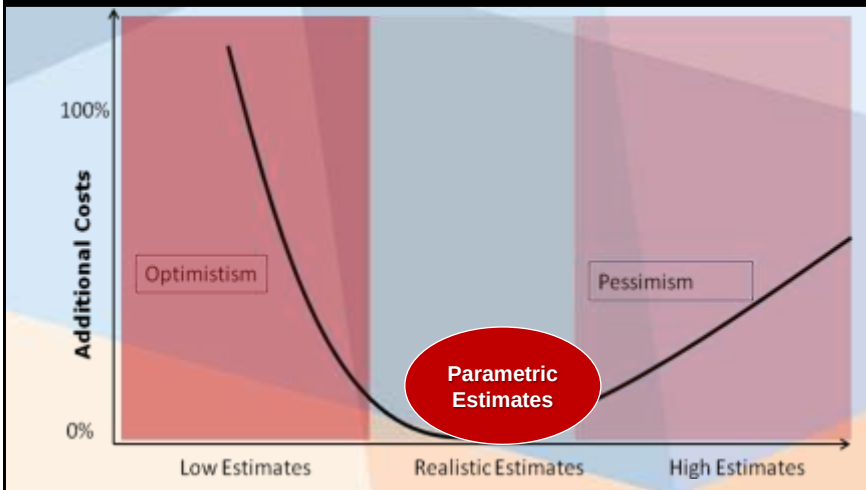
Are these really **dis**advantages?

**Disadvantages:**
- Requires a certain maturity level of an organization;
- Measurement and analysis of completed projects;
- Investment in expertise and tooling;
- Documentation requirements – must be possible to measure;
- Knowledge about estimation parameters is necessary.

**Result: realistic estimate, scenario's, risk profile.**

nesma READY TO CONNECT SOFTWARE METRICS

## Parametric estimates




nesma READY TO CONNECT SOFTWARE METRICS

## Estimates in practice

**Many projects start with optimistic estimates**

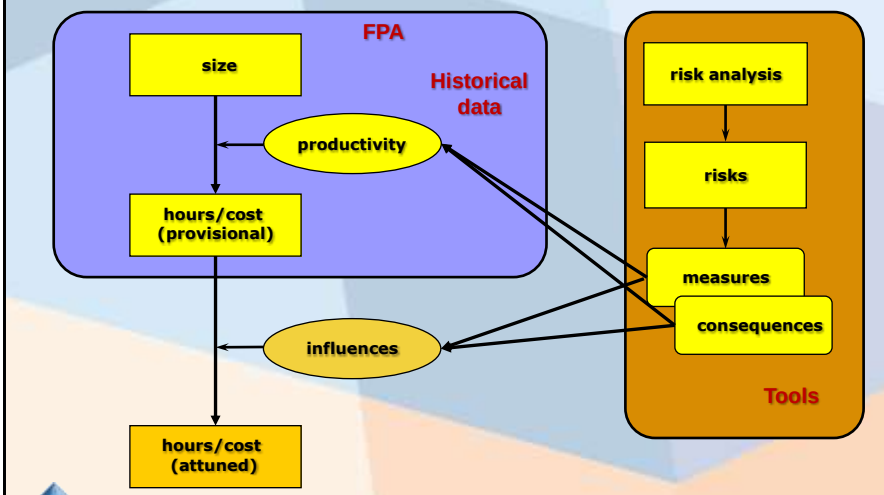
- Organizations that use only expert estimates;
- Outsourcing: suppliers don't use parametric estimation;
- Outsourcing: customer selects lowest price;
- Business / Customer: pressure to deliver faster and cheaper;
- Final estimates have to be made based on incomplete requirements;

**Software horror stories are a common phenomenon**

- But many can be avoided by using parametric estimates
- Or at least perform a **reality check** of the estimate!!

nesma READY TO CONNECT SOFTWARE METRICS

## Basic estimation model




nesma READY TO CONNECT SOFTWARE METRICS

## Size

Most important input parameter is **size.**

The two most used size units for software are:

Technical size – slocs

- Has to be guessed / not possible to measure upfront
- More slocs is good / bad? Who needs slocs?
- No standard;
- Slocs cannot really be compared between technical environments.

Functional size – function points

- Can be measured upfront (fairly accurately);
- More function points means more functionality;
- ISO/IEC standard;
- Independent of technical environment.

nesma READY TO CONNECT SOFTWARE METRICS

## Estimate breakdown

All effort estimates can be broken down to these components:

- Size (Unit of Measure)
- Productivity (effort hours per UoM)
- Adjustments (informed decisions about the specific project)

When this is done, it becomes possible to perform a reality check based on historical data.

- Company data
- Industry data

nesma READY TO CONNECT SOFTWARE METRICS

## ISBSG

**International Software Benchmarking Standards Group**

- Independent and not-for-profit;
- Full Members are non-profit organizations;
- Grows and exploits two repositories of software data (in .xls):
  - New developments and enhancements (> 6000 projects);
  - Maintenance and support (> 1200 applications).

**Everybody can submit project data**

DCQ on the site / on request (.xls)
Anonymous
Free benchmark report in return

nesma READY TO CONNECT SOFTWARE METRICS

## ISBSG industry data

Mission: "**To improve the management of IT resources** by both business and government, through the provision and exploitation of public repositories of software engineering knowledge that are standardized, verified, recent and representative of current technologies".

**All ISBSG data is**
- validated and rated in accordance with its quality guidelines
- current
- representative of the industry
- independent and trusted
- captured from a range of organization sizes and industries

nesma READY TO CONNECT SOFTWARE METRICS

## Performing a reality check

**Reality check of an estimate**

1. Break down the estimate
   - Size
   - Productivity
   - Adjustments
2. Select a relevant peer group in the historical database
3. Analyze the productivity of this peer group
4. Define the 'reality zone'
5. Assess whether the estimate is in the reality zone
   - If yes, the estimate is probably realistic
   - If no, the estimators have to explain why

nesma READY TO CONNECT SOFTWARE METRICS

## Example

**Project X**
Technical environment: Java
Effort estimate (by team of experts): 2000 hours
No risk adjustments made

**Breakdown:**
1. Functional size: 411 function points (IFPUG)
2. Productivity: 2000/411 = 4.9 hours/FP
3. Adjustments: 0 hours

nesma READY TO CONNECT SOFTWARE METRICS

## Select relevant peer group

ISBSG 'New Developments & Enhancements'
Select a relevant 'peer group'
- Data Quality A or B
- Count approach: IFPUG 4.x or NESMA
- Primary Programming Language = 'Java'
- 300 FP < Project Size < 500 FP

ISBSG
D&E Corporate Release 17 April 2013

| ISBSG Project ID | Application Type | Development Type | Primary Programming Language | Count Approach | Functional Size | Relative Size |
|---|---|---|---|---|---|---|
| 1475 | Management or performance | New Development | Java | IFPUG 4+ | 430 M2 | |
| 14816 | Follow up of car failure | Enhancement | Java | IFPUG 4+ | 316 M2 | |
| 14879 | IT management | New Development | Java | IFPUG 4+ | 427 M2 | |
| 15130 | Financial transaction process | New Development | Java | IFPUG 4+ | 435 M2 | |

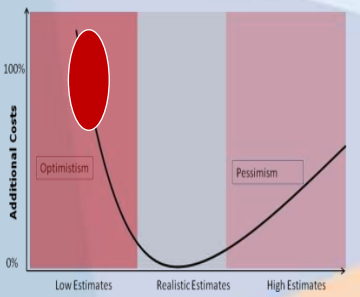
nesma READY TO CONNECT SOFTWARE METRICS

## Data analysis

488 projects selected
Productivity Analysis:

| Productivity | h/FP |
|---|---|
| Minimum | 0.1 |
| Percentile 10 | 4.5 |
| Percentile25 | 6.7 |
| Median | 9.8 |
| Percentile 75 | 15.4 |
| Percentile 90 | 21.6 |
| Maximum | 78.3 |
| Average | 14.2 |

Estimate project X: 4.9 h/FP
**Not realistic !!**




Reality zone: Percentile 25 – Percentile 75

nesma READY TO CONNECT SOFTWARE METRICS

## Conclusions & recommendation

- Accurate software estimation is hard;
- Optimistic estimates result in failing projects;
- Most organizations only use expert estimates;
- Expert estimates usually result in optimistic estimates;
- Performing reality check using historical data is easy.

Strong recommendation:

Always perform a **reality check** to decrease the risk of horror stories !!

nesma READY TO CONNECT SOFTWARE METRICS


October 6-8, 2014
nesma
2014.iwsm-mensura.org
IWSM MENSURA 2014 - ROTTERDAM