Sources of Error in Software Cost Estimation

Seminar on Software Cost Estimation

Silvio Meier

Presentation Schedule

- Accuracy of historical cost data
- Correcting historical cost data
- Judging the accuracy of software cost estimations
- Classes of software estimation errors
- Conclusions
- Discussion

Accuracy of Historical Costs (1)

- Why do we need historical costs from software projects?
  - Development of software cost estimation tools / methods
  - Comparing estimation against its real (historical) costs
- Problem: Historical cost data is often biased
  - Tools predict often 50 – 100% higher costs than tracked costs of a project → Why?
  - What is inaccurate? Is it the historical data or the cost tracking tools?
    - Tendency to omit data
  - Sources of inaccuracy: Conventional cost tracking tools not optimized for software cost tracking → omitting of activities and task, as example: Early requirement phase, unpaid overtime, etc.

Accuracy of Historical Costs (2)

- Other omissions beside omitting of activities:
  - Include all classes of workers
  - Unpaid overtime
- According to [Jones98], the most common omissions in cost tracking systems
  - Cost tracking initialized after first project activities
  - Non-programming personnel work
  - Project management work
  - Technical work performed by users
- Other reasons for biases
  - Project funding low → tendency to charge time to other projects
  - ..

→ Results of overall tracked project costs are too low!

Correcting Historical Cost Data (1)

- Historical cost data for calibration and accuracy check
  → For adjustment or avoiding biases:
  - Excluding incomplete projects from the estimation portfolio
  - Correcting missing data based on interviews with the project members
  - Building activity-based cost estimated tools
- Interviews on several projects found resulted in:
  - 5 – 10 % of the costs is spent before cost tracking is set up
  - 15 – 30 % of the work is done by workers which are not in the cost tracking system
  - Management effort is between 10 and 20 % and is sometimes not tracked
  - User participation in technical work is about 5 – 20%
  - Unpaid overtime of exempt professionals is about 5 – 15%

Correcting Historical Cost Data (2)

- The values of omitted costs vary from industry to industry:

Software Subindustry	Percentage of missing data	Most common omissions
Military software	10	Unpaid overtime
Contracted or outsourced software	10	Unpaid overtime
Systems software	12	Unpaid overtime and documentation
Commercial software	15	Unpaid overtime, user activities, noncoding tasks, ….
End-user software	75	Everything but coding

Correcting Historical Cost Data (3)

- More general suggestions to avoid biases in cost data:
  - Try to make data granular
  - Break down of the project into activities and tasks → Example
    - Needed at least for the fine tuning of cost estimation tools
    - Resulting activities, task and sub-tasks can result in more than 1000 elements
  - Only the data from 5 of 25 activities can directly be used for estimation → other data has to be corrected.
  - Activity based cost estimation is more and more common

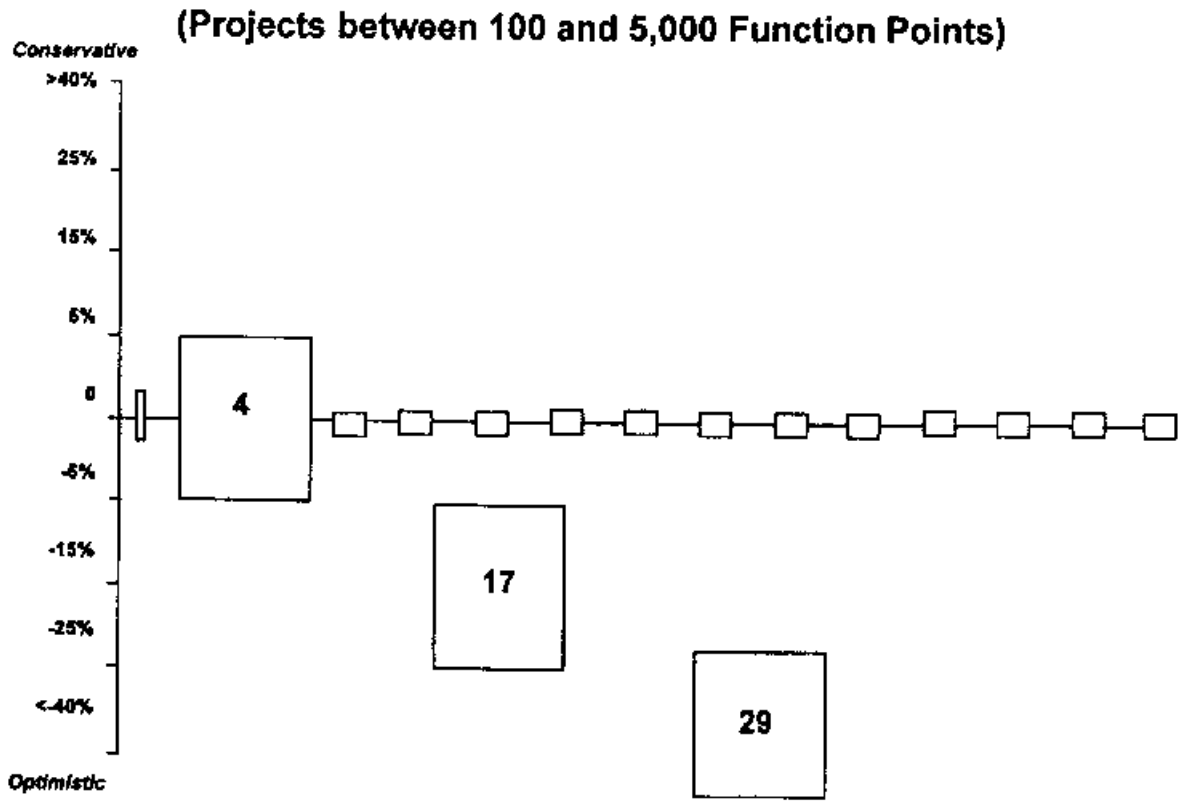
Judging Accuracy of Software Cost Estimations (1)

- Three general questions related to the accuracy of software estimation:
  - Estimation results of tools compared to historical cost data
  - Estimation results of tools compared to competing products
  - Accuracy of manual estimation vs. tool estimation
- First issue: As discussed before: Tools are often more accurate corresponding than historical cost data
- Second issue: See [Jones98], there are discussed several studies about comparison of cost estimation tools.

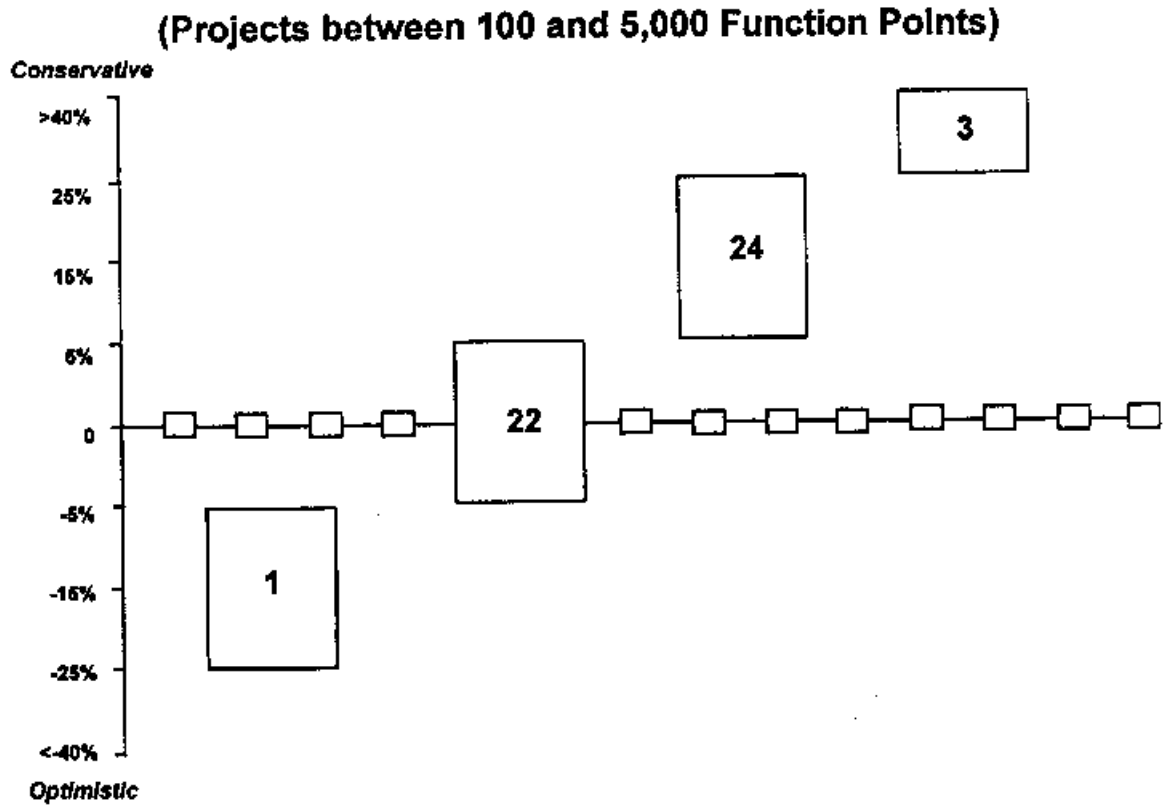
Judging Accuracy of Software Cost Estimations (2)

- Third issue:
  - Empirical results show, that manual estimates are often too optimistic
  - [Jones98] mentions a study where 50 manual estimated software projects were compared to 50 tool estimated software project.
  - Only 4 manual estimated projects were in the range of –5% to +5% deviation of the real costs
  - Whereas 22 of 50 projects estimated by a tool were in the +/- 5% deviation range
  - Tool based estimates were more often even conservative

Judging Accuracy of Software Cost Estimations (3)



Judging Accuracy of Software Cost Estimations (4)



Judging Accuracy of Software Cost Estimations (5)

- When the responsible people were asked about the too optimistic estimation, they responded the following answers:
  - I could not get approval for an accurate estimate, so I had to change it
  - The project doubled in size after the requirements
  - Debugging and testing took longer than we thought
  - The new case tools we were using didn’t work right and slowed us down
  - We didn’t have any estimating tools available at the same time the estimate was needed
  - I lost some of my developers and had to find replacements

Classes of Software Estimation Errors

- [Jones98] distinguishes 12 classes of errors in software cost estimation:
  - Metrics errors
  - Scaling errors
  - Executive and client errors
  - Sizing errors
  - Activity-selection errors
  - Assignment-scope errors
  - Creeping user requirements errors
  - Critical path errors
  - Staff build-up errors
  - Technology adjustment errors
  - Special or unique situations

Metrics Errors

- Mainly occurring with LOC
  - How to count
  - Only half of the software project are related to code
- Latter problem is worse!
  - Wrong productivity assumptions
    - Granularity
    - Problem cross-language measurement
- Function points are less error-prone
- Especially when doing manual estimations
- Impact of the error: deviation can exceed 100 %

Scaling Errors (1)

- Problems when using data from small projects for estimating large projects:
  - Large projects need more activities (out of 25) than small ones
    - Small projects 10 – 15 activities
    - Large civilian projects at least 20 activities
    - Large military projects all 25
  - Large projects have other cost profiles than small ones:
    - Other activities than coding are becoming important for large projects
    - Example: following table with project size
- Impact: Up to 1000 % deviation

Scaling Errors (2)

Size, Function Points	Size, KLOC	Coding %	Paperwork %	Defect Removal %	Manage-ment and Support %
1	0.1	70	5	15	10
10	1.0	65	7	17	11
100	10.0	54	15	20	11
1000	100.0	30	26	30	14
10000	1000.0	18	31	35	16

Executive and Client Errors

- Influence of management and clients on software cost estimation
  - Management reject accurate estimates
  - Subjective opinion of management not objective estimates
- Analyses mentioned in [Jones98] → late projects cause by executive and client errors.
- Impact
  - Schedule deviation about 50 %
  - Costs deviation about 100%
- Often related to the critical path error

Sizing Errors

- Errors in predicting sizes for
  - Internal deliverables (pages of specifications, pages of plans, etc.)
  - External deliverables (quantity of source code, number of screens, etc.)
- More common for manual estimates → with tools quite accurate
- Impact of error (manual estimation):
  - Stable requirements, experienced project manager: +/- 15 %
  - Stable requirements, inexperienced project manager up to 100% and more

Activity-Selection Errors

- Error: Omitting necessary work (Phase, activity, task, sub task)
- Tool support
  - Customizing a specific chart of account for the project
  - Templates from former projects
- Different project types/sizes contain different activities → [Jones98] and [Jones02]
- Impact of the error:
  - Varies widely
  - Up to 1000% deviation

Assigment-Scope Errors (1)

- Assignment scope: Quantity of work handled by the staff
  - Workload to high → Error
  - Today supported by tools, using natural or synthetic metrics
  - Error more common for manual estimates
  - Also depending on experience of staff
- How to determine workload
  - Determining job profiles
  - Determining work load per job profile
- Several job profiles with assignment scope identified
- Some job profiles are difficult to find a assignment scope

Assigment-Scope Errors (2)

- Unsolved problems
  - Firms do not use categorization for job profiles
  - Difficult to do research
- Assignment scope is more and more important:
  - Downsizing, layoffs of personnel
  - Business process engineering
  - Shortages of software personnel (e.g. year-2000)
  - More and more outsourcing arrangements
- Impact: Range of uncertainty can reach up to 100 %

Production-Rate Errors

- Production-rate
  - Amount of work that can be completed by one person within a standard period of time
  - Metrics: Natural or synthetic, synthetic metrics are to prefer, → example Table
  - Synthetic metrics are comparable and additive over different activities
  - Often supported by tools, available from historical data
- Impact of the error
  - Range of uncertainty is linear to the range between the true rate and the anticipated rate
  - Not easy to determine

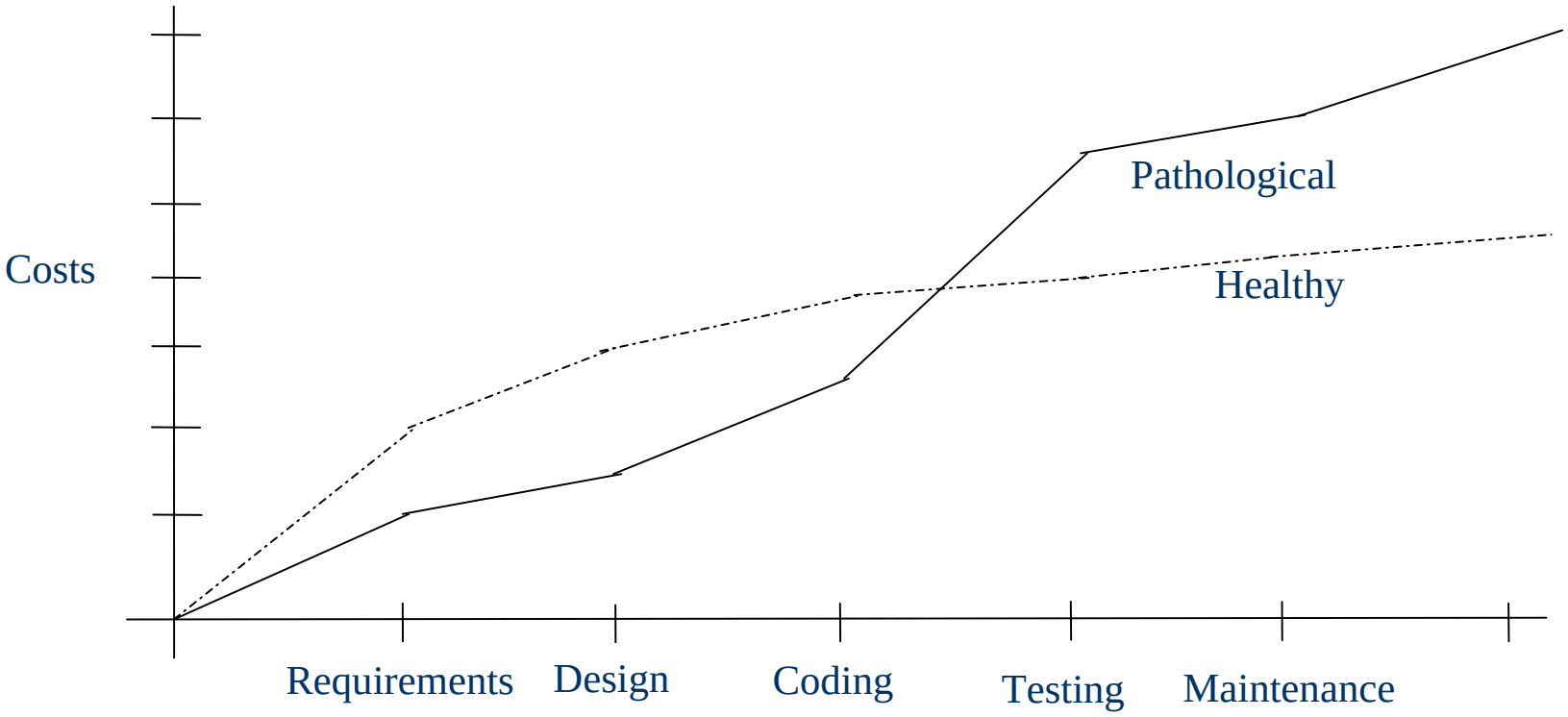
Creeping User Requirements Errors

- Creeping requirements
  - Appear or evolve during the software process
  - Thumb rule: 2% of initial requirements volume per month
- Tool support
  - Some of the tools try to predict the probable volume of creeping requirements
- Function points are more suitable
- Impact
  - Normally deviation of costs is linear to the difference between planned volume and unplanned functionality

Critical Path Errors (1)

- Critical path
  - Critical path through the net of activities
  - Late activities on the critical path makes the project late
- Most often the reason for this error: skimping in quality control measures:
  - Study mentioned in [Jones98]: 84 Projects of IBM and ITT
  - Reason for underestimated quality control measures: Executive and client errors
  - Pathological projects showed all the same behavior (see next slide)
- Impact: up to 25 % delay in project schedule and about +35% deviation in costs

Critical Path Errors (2)



Staffing Build-Up Errors

- Completion of Software development Contract
  - Often not all of the needed staff available or missing
  - Recruiting of the corresponding personnel from the market
- Error in staffing build-up occurs when recruiting takes longer than anticipated
- Problem amplification, if shortage of software personnel on the market. This problem is accompanied by rising salaries.
- Impact: Impact is not easy to predict

Technology Adjustment Errors

- Software technology evolves, new technology is developed
- Technology adjustment errors occur, if the influence of technology is not correctly anticipated
- Examples:
  - Advertising claims of cost estimation tool vendors
  - New methods / methodologies are not immediately supported by cost estimation tools
  - Steep learning curves for some technologies
- Impact: The range of uncertainty can be up to 150%

Special or Unique Situations

- Situations that have deep impact on the projects in the sense of schedule or costs
- Examples:
  - Fire, weather, other natural disasters
  - termination of more than 50 percent of the project team members
  - Major layoffs, downsizing
  - Travel costs for trips among geographically dispersed projects.
  - ....
- Often remaining risks, i.e. little probability of occurring
- Such situations are not easy to predict → therefore can not handled through tools (algorithmic estimations)
- Impact: Not easy to predict, wide range, depends on the situation

Conclusions

- Estimation process is very complex
  - Many variables to take into account
  - Problems of omissions (Example activity-selection error)
- Tool can support to get more accurate data
  - Algorithmic size estimation
  - Guidance through estimation process
  - Take into account of all variables which influence the software costs
  - ...
- But still needed a lot of experience

Finish

Questions?