



Foundations of Software Engineering 605.601

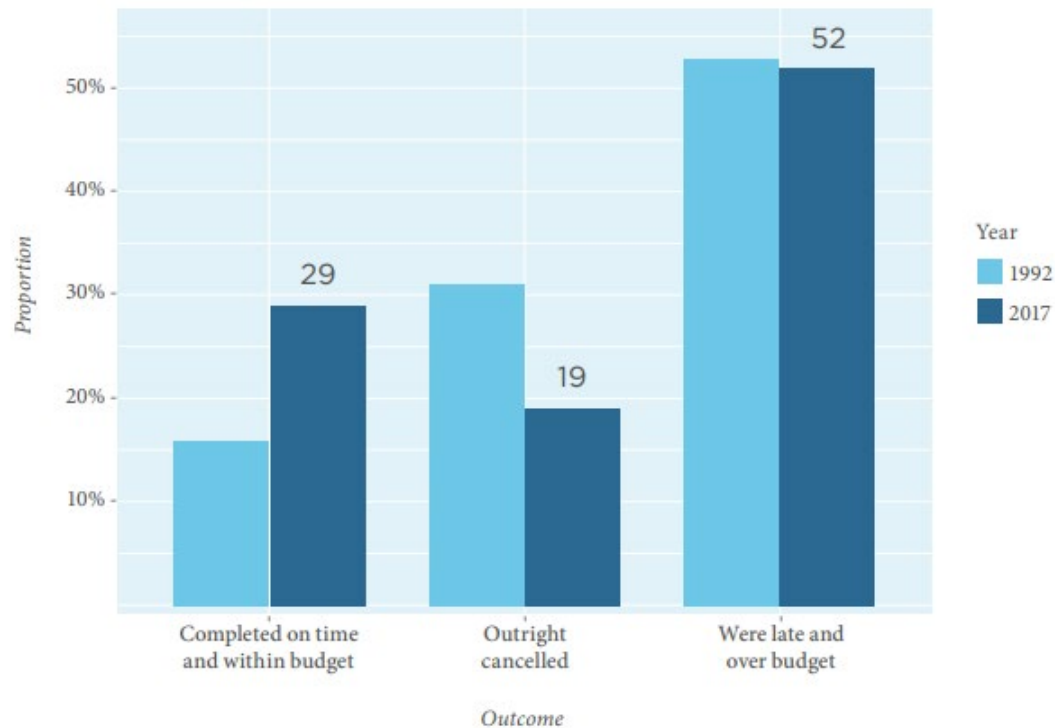
Some Recent Trends in Software Project Success Rates & Estimation Techniques

Johns Hopkins University
Whiting School of Engineering
Engineering for Professionals

Joseph M. Demasco
2022



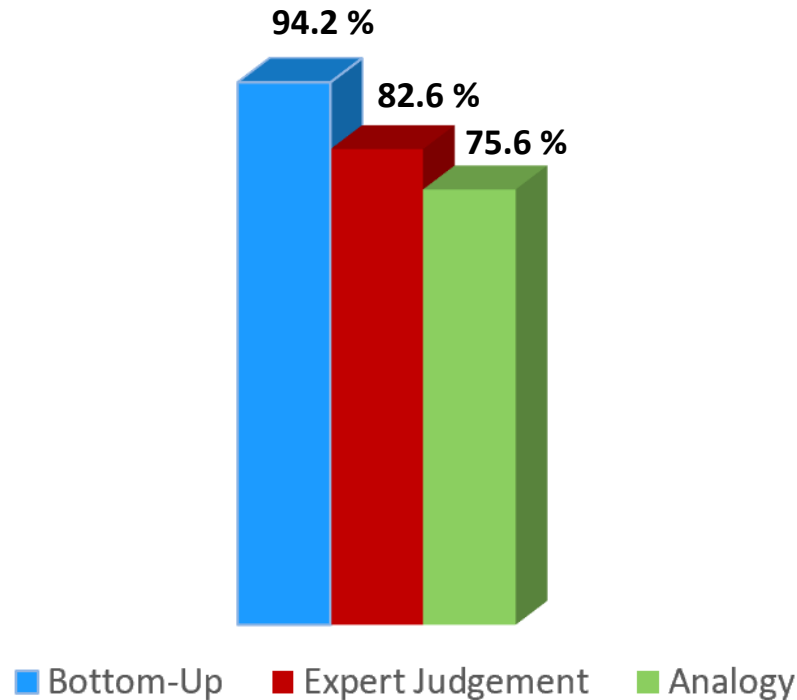
Success & Failure Rates Over 25 Years



Source: Reported in H. Krasner, "Quality Software in the U.S.: A 2018 Report," Consortium for IT Software Quality (CITSQ), September 26, 2018.



Most Commonly Used Estimating Techniques



J.B. Prater, K. Kirytopoulos, T. Ma, "An Investigation of Estimation Techniques for Information Technology Projects," [IEEE Explore](#), February 2020



What We Know & Don't Know About Estimation

- No significant improvement in estimating accuracy in several decades
- Cost and effort overruns are still very common in software projects
- Average overrun is about 30 percent
- Statistical models don't improve accuracy when compared to simpler models

Magne Jorgensen, "What We Do and Don't Know About Software Development Effort Estimation," IEEE Software, Volume 31, Issue 2, Mar-Apr 2014



What We Know About Estimation

- Sophisticated models don't improve accuracy
- Relevant historical data and checklists can improve estimation accuracy
- Combining independent estimates improves accuracy
- Minimum/maximum effort intervals are too narrow

Magne Jorgensen, "What We Do and Don't Know About Software Development Effort Estimation," IEEE Software, Volume 31, Issue 2, Mar-Apr 2014



What We Don't Know About Estimation

- How to accurately estimate the effort of mega-large complex projects
- How to measure software size and complexity for accurate estimation
- How to measure and predict productivity

Magne Jorgensen, "What We Do and Don't Know About Software Development Effort Estimation," IEEE Software, Volume 31, Issue 2, Mar-Apr 2014



Some Recommendations

- Use simple estimation models tailored to local contexts in combination with expert judgement
- Use historical estimation error
- Use expert judgement and independent estimates to set minimum/maximum effort intervals
- Use tailored checklists
- Use structured, group-based estimation processes where independence of estimates is assured
- Avoid early estimates based on highly incomplete information, if possible...and use probabilities, if necessary, to communicate uncertainties

Magne Jorgensen, "What We Do and Don't Know About Software Development Effort Estimation," IEEE Software, Volume 31, Issue 2, Mar-Apr 2014