

10 Deadly Sins of Software Estimation

© 2002 Construx Software Builders, Inc.
All Rights Reserved.

www.construx.com

Construx
Delivering Software Project Success



Background

- ❖ **Estimation Book**
- ❖ **Construx Estimate**
- ❖ **Construx's Training**
- ❖ **Construx's Consulting**



Art and Science of Software Estimation

- ❖ **Science is well-developed and well-supported by software tools**
- ❖ **The art of estimation relies more on rules of thumb and still needs some work**

Almost-Deadly Sins of
Software Estimation
Sins #20-#11



Sin #20

**Estimating how long “it” will take to build
before anyone knows what “it” is**



Sin #19

**Assuming that the most reliable estimates
come from the people with the most
powerful vocal chords**



Sin #18

Telling someone you're writing an estimation book, because they will say, "When do you *estimate* you'll be done, ha ha ha."



Sin #17

**Creating an estimate for a new project by comparing it to a past project ...
... which overran its estimates...
... and ultimately realizing that you based the new project's plans on the past project's *estimated results* instead of its *actual results***



Sin #17(a)

Creating an estimate for a new project by comparing it to a past project ...

... which worked massive overtime ...

... and ultimately realizing that, while you based the new project's estimates on the past project's *actual results* this time, by using the past project you have calibrated massive overtime into your estimates

9



Sin #16

Assuming that the sales department is better at estimating software projects than the programmers are

10



Sin #15

Creating estimates that assume that no one will go to training ...

or attend meetings ...

or be called to work on another project ...

or need to support a key customer ...

or take a vacation ...

or get sick ...

or ...

11



Sin #14

Presenting estimates with a high degree of precision (e.g., “67.4 days”) but that are supported by only a low degree of accuracy (“±2 months”)

12



Sin #13

Believing that results from commercial estimation software are no match for a pencil and a beer-stained napkin



Sin #12

Reasoning that, “The sooner we fall behind schedule, the more time we’ll have to catch up.”



Sin #11

Arguing that the software developers are padding their estimates just so they can look good ...

... when the last time the organization delivered a software project early was during the Nixon administration!

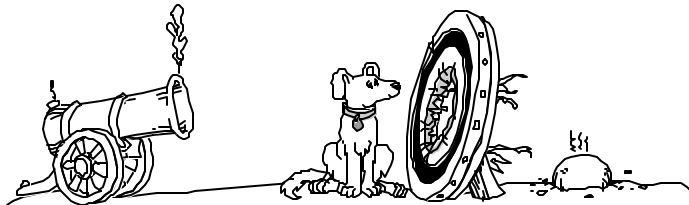


Sin #1 Confusing Targets with Estimates



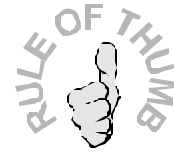
Confusing Estimates with Targets

- ❖ **The software industry does lots of target setting**
- ❖ **These targets are not created through any kind of analysis based on the work to be performed**
- ❖ **In practice, little real estimation is done**





Differentiate Between Targets and Estimates



- ❖ **Target setting is a key part of the art of estimation**
- ❖ **When you're asked to provide an estimate, determine whether you're really supposed to be *estimating* or *figuring out how to meet a target***
- ❖ **This is best treated as an iterative process that brings estimates and targets into alignment**



**Sin #2
Saying "Yes" When
You Really Mean "No"**



Why Developers Say “Yes”

It is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data, and certified chiefly by the hunches of the managers.

— Fred Brooks (1975)

21

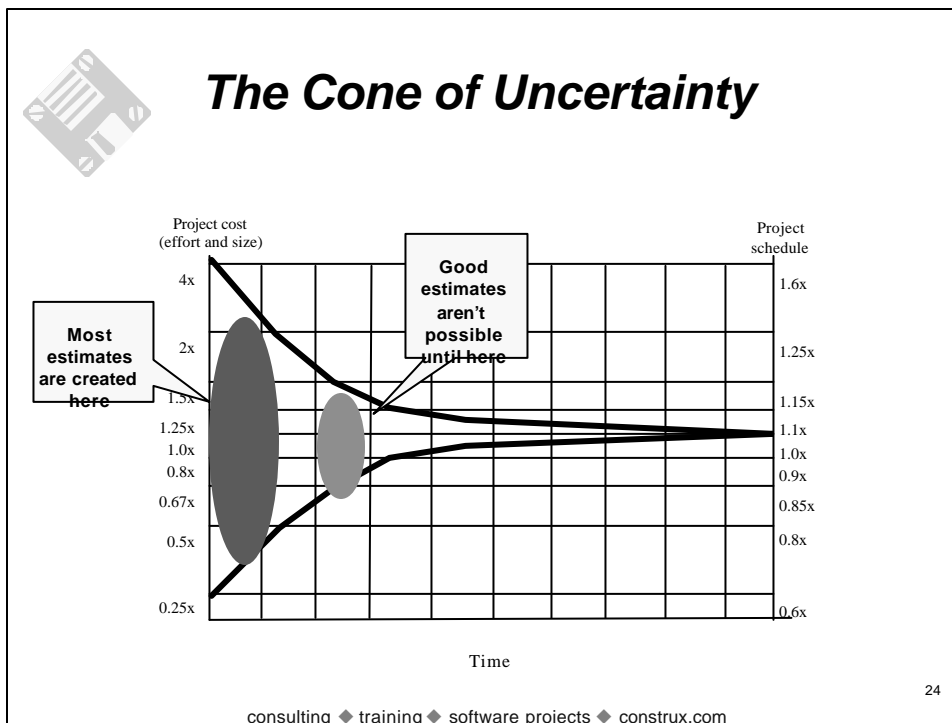


Schedule Negotiations

- ❖ **Software developers tend to be introverts and relatively young**
- ❖ **Marketing and sales personnel tend to be more extroverted and organizationally senior to the developers they negotiate with**

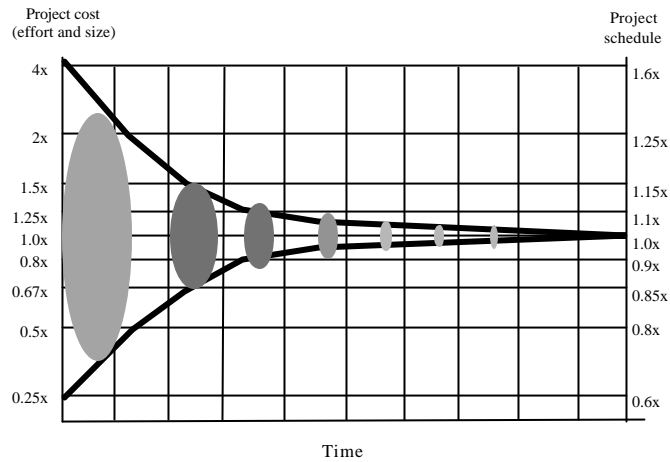
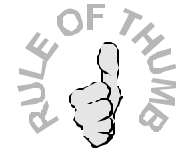
22

Sin #3 Committing to Estimates Too Early in the Cone of Uncertainty





Plan to Revise Estimates Throughout the Project



25

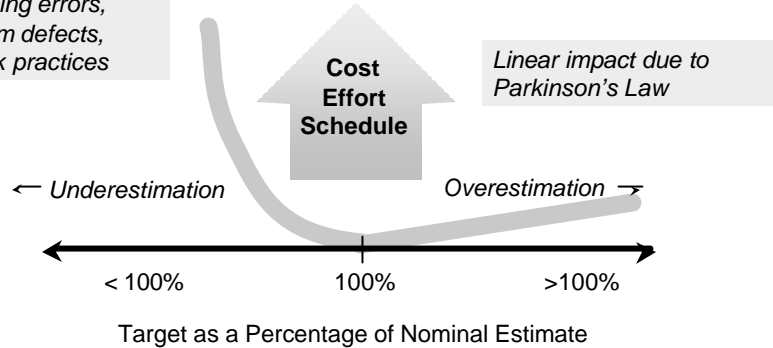
consulting ♦ training ♦ software projects ♦ construx.com

Sin #4
Assuming
Underestimation has a
Neutral Impact on
Project Results



Effect of Estimation Accuracy

Non-linear impact due to planning errors, upstream defects, high-risk practices



Linear impact due to Parkinson's Law

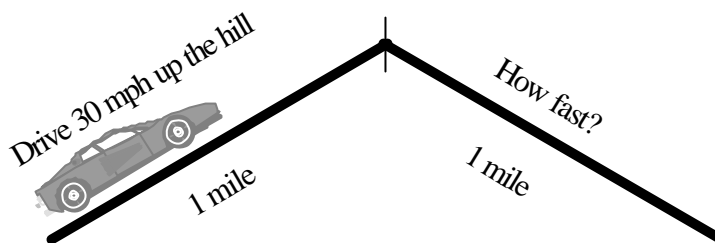
Cost Effort Schedule

Sin #5
Estimating in the
"Impossible Zone"



Puzzle

- ❖ Suppose you drive 30 mph up a hill 1 mile.
- ❖ How fast do you need to drive down the hill to average 60 mph for the entire trip?



29



Variation on Sin #5

[The common definition of estimate is] 'An estimate is the most optimistic prediction that has a non-zero probability of coming true' . . .

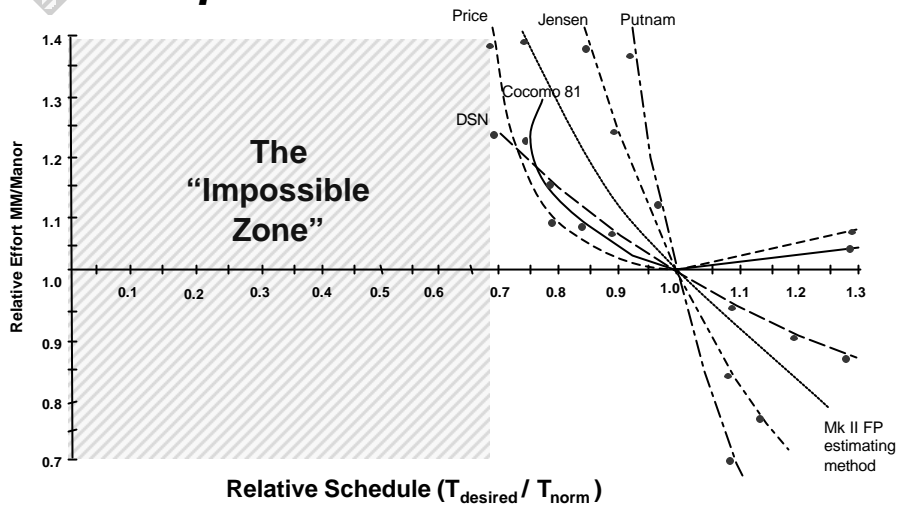
Accepting this definition leads irrevocably toward a method called what's-the-earliest-date-by-which-you-can't-prove-you-won't-be-finished estimating.

— Tom DeMarco (1982)

30



Schedule Compression and the Impossible Zone



Source: Adapted from Charles Simons, *Software Sizing and Estimating: Mk II*, John Wiley & Sons, 1991.

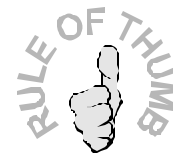
31

consulting ♦ training ♦ software projects ♦ construx.com



Effort/Schedule Tradeoff

- ❖ **All** researchers have found some tradeoff between schedule compression and effort
- ❖ **No one** thinks there's no tradeoff
- ❖ Assume a maximum possible schedule compression of about 25%

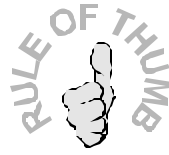


32

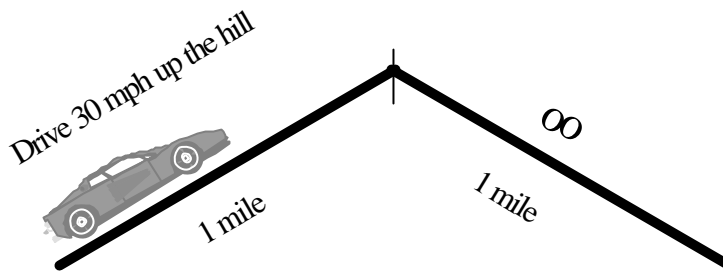
consulting ♦ training ♦ software projects ♦ construx.com



Don't Create Estimates in the "Impossible Zone"



❖ What's the solution to the puzzle?



Sin #6
Overestimating
Savings from New
Tools or Methods

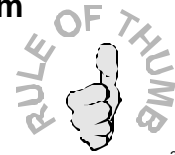


Savings from New Tools or Methods

Problems:

- ❖ **Must pay learning curve price during first use**
- ❖ **New tools and methods increase risk**
- ❖ **Maximum effectiveness doesn't appear during first use**
- ❖ **Payoff is less than expected when it does appear**

Best assumption is productivity loss from first use of new tool or method



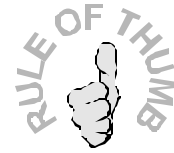
35



**Sin #7
Using Only One
Estimation Technique**



Use Multiple Techniques

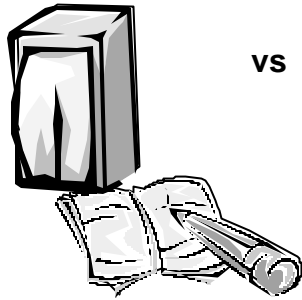


- ❖ **Lack of this contributes to Brooks' "vigorous defense" problem**
- ❖ **Most leading organizations use multiple techniques**
- ❖ **Create estimates different ways and look for convergence or spread among the estimates**
- ❖ **Personal experience**

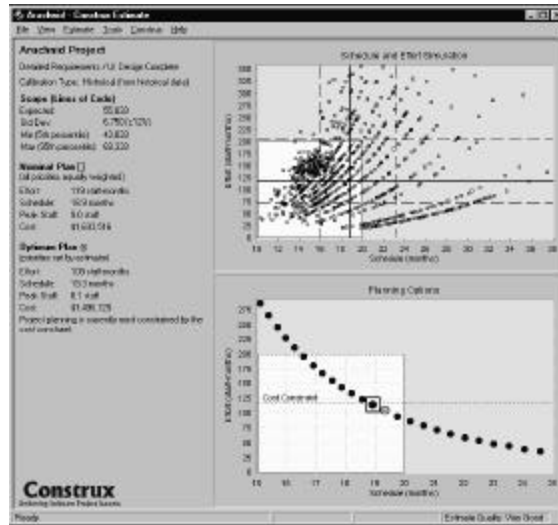
**Sin #8
Not Using Estimation
Software**



Estimation Software



VS



39



Use Estimation Software



- ❖ Best support for *science* of estimation is tools
- ❖ Estimates created with tools can have more credibility than estimates created by manual methods
- ❖ Construx Estimate--Free Download:
www.construx.com/estimate/

40



Sin #9 Not Including Risk Impacts in Estimates

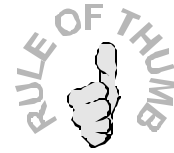


How Much Risk Gets Included in the Project Plan?

Risk	Probability	Impact	Exposure (RE)
New technology doesn't live up to expectations	25%	8 weeks	2.0 weeks
New technology requires staff training	50%	1 week	0.5 weeks
Demo version of software is required to support trade show	75%	2 weeks	1.5 weeks
Senior staff not available as planned	25%	10 weeks	2.5 weeks
Government regulations change before software ships	10%	2 weeks	0.2 weeks
Total	-	23 weeks	6.7 weeks



Addressing Risk in Estimates

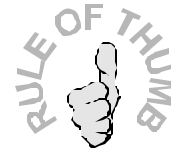


- ❖ **Software projects are inherently risky**
- ❖ **The total Risk Exposure (RE) is the expected value of the project overrun**
- ❖ **The RE is where “buffer planning” starts**

**Sin #10
Providing Off-The-Cuff
Estimates**



Treat Estimation as a Mini-Project

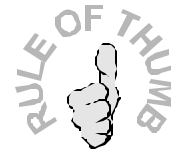


- ❖ Use of *guessing* and *intuition* to create estimates is correlated with cost and schedule overruns (at the 0.05 level of significance)
- ❖ Use of simple arithmetic formulas is negatively correlated with overruns (at the 0.01 level of significance)

45



Define a Standardized Estimation Procedure



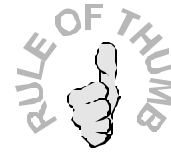
Elements of a standardized procedure:

- ❖ A clear description of an estimate's imprecision
- ❖ Use of multiple estimation approaches
- ❖ A plan to re-estimate at pre-defined points in the project

46



Use Historical Data

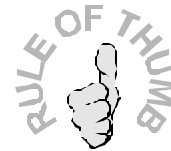


- ❖ **Most common estimation technique is comparing to past projects using personal memory**
- ❖ **Use of similar past projects based on documented facts is negatively correlated with overruns (at the 0.05 level of significance)**

47



Decompose Big Estimates Into Smaller Estimates

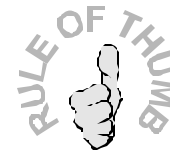


- ❖ **Decompose systems into modules**
- ❖ **Decompose big tasks into small tasks**
- ❖ **Makes use of a statistical property called “the law of large numbers”—highs and lows tend to cancel each other out**

48



Adjust Your Estimation Approach as the Project Progresses



- ❖ **Early in the project, algorithmic, “macro” approaches work best**
- ❖ **Late in the project, bottom-up, task-based estimates work best**
- ❖ **Different guidelines apply to different kinds of estimates**

49



Summary of 10 Deadly Sins

- ❖ **Confusing targets with estimates**
- ❖ **Saying “yes” when you really mean “no”**
- ❖ **Committing to estimates too early in the cone of uncertainty**
- ❖ **Assuming underestimation has a neutral impact on project results**
- ❖ **Estimating in the “impossible zone”**
- ❖ **Overestimating savings from new tools or methods**
- ❖ **Using only one estimation technique**
- ❖ **Not using estimation software**
- ❖ **Not including risk impacts in estimates**
- ❖ **Providing off-the-cuff estimates**

50



Construx

Delivering Software Project Success

Contact Information

Services

- ❖ Software Projects
- ❖ Consulting
- ❖ Seminars

Software Resources

- ❖ info@construx.com
- ❖ www.construx.com