SOFTWARE ESTIMATION TIPS

Wojciech Jaworski
jaworski.wojciech@gmail.com

“

Prediction is hard,
especially about the future

Niels Bohr, Danish physicist

A PROBLEM SCALE

adapted from Steve McConnell's "Software Estimation: Demystifying the black art"

WHAT IS THE ESTIMATE?

"A GUESS OF WHAT THE SIZE, VALUE, AMOUNT, COST, ETC. OF SOMETHING MIGHT BE"

http://dictionary.cambridge.org/dictionary/english/estimate

?

DOES BUSINESS HAVE THE SAME MEANING?


IT
BUSINESS
ESTIMATE
COMMITMENT
TARGET

GUESS $\not\Rightarrow$ GUARANTEE

GUESS $\longrightarrow$ PROBABILITY

LIMIT SIGNIFICANT DIGITS IN ESTIMATES

(THEY SIGNALISE A LACK OF UNCERTAINTY)

11 MONTHS 27,5 DAYS

VS

ABOUT 1 YEAR

“

It’s better to be roughly right than precisely wrong

John Maynard Keynes

DON'T GIVE A SINGLE NUMBER EFFORT ESTIMATE

YOU CAN EXPRESS UNCERTAINTY IN:

- A PERCENT-CONFIDENT NUMBER
- A RANGE

RANGES USE TO BE TOO NARROW:

- MANAGEMENT PRESSURE
- SELF-INDUCED PRESSURE (MORE OFTEN)

?

WHAT IS MORE UNPROFESSIONAL?

WIDER NARROW ESTIMATE

OR

NOT ACHIEVING A TARGET

SELF-CONFIDENCE

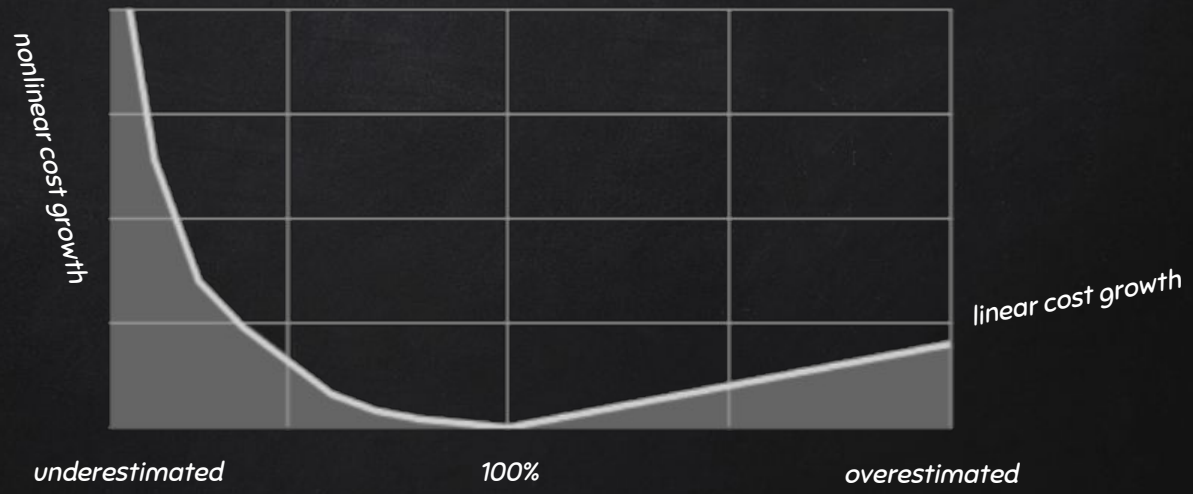
A lot of things went wrong in the last project...

This time we have no impediments.

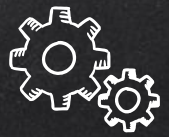
We learned a lot. We will be more productive.

DON'T LET ANYONE REDUCE DEVELOPERS' ESTIMATES – THEY ARE PROBABLY ALREADY TOO OPTIMISTIC

COSTS OF OVER- AND UNDER-ESTIMATION



adapted from Steve McConnell's "Software Estimation: Demystifying the black art"



PARKINSON'S LAW:

WORK DOES EXPAND TO FILL AVAILABLE TIME

PENALTIES FOR UNDERESTIMATION

MORE RESOURCES

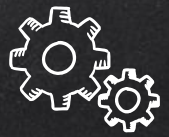
MORE BUGS

CUSTOMER DISSATIS-FACTION

MORE SYNCHRO-NISATION

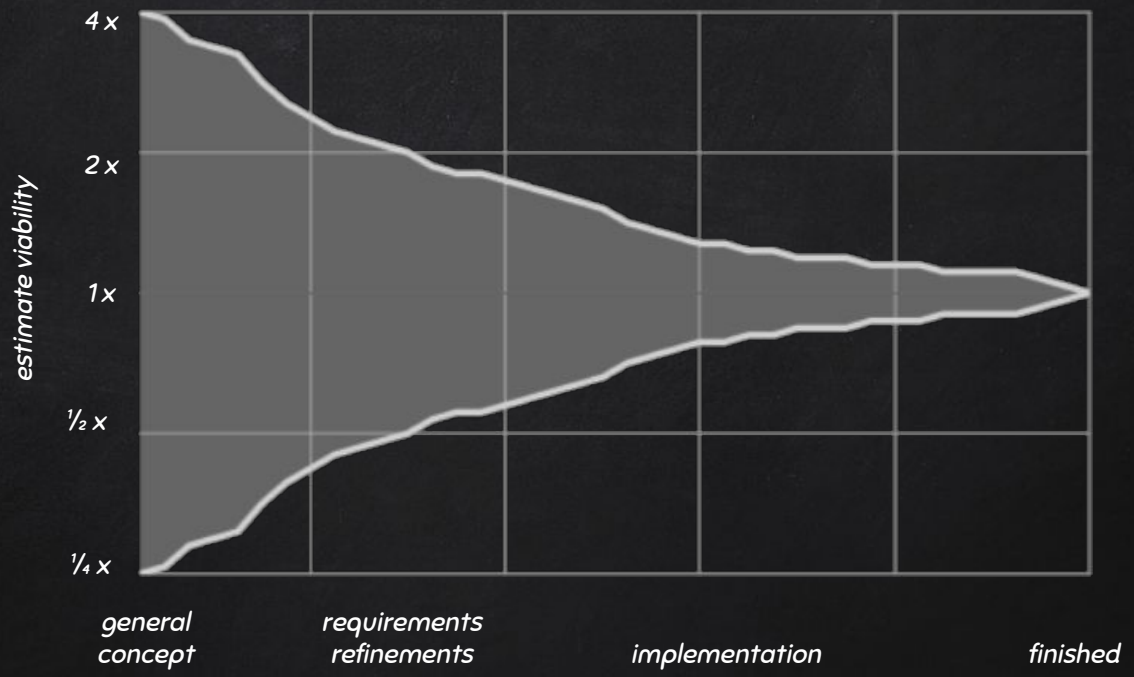
POOR QUALITY

DON'T INTENTIONALLY UNDERESTIMATE, THE PENALTY FOR UNDERESTIMATION IS MUCH MORE SEVERE THAN FOR OVERESTIMATION



THE ACCURACY OF THE SOFTWARE ESTIMATE DEPENDS ON THE LEVEL OF REFINEMENT OF THE SOFTWARE DEFINITION

CONE OF UNCERTAINTY



adapted from Steve McConnell's "Software Estimation: Demystifying the black art"

DON'T MAKE A COMMITMENT TOO EARLY

IF NEED OVERALL ESTIMATE BEFORE DEVELOPMENT STARTS,
USE DIFFERENT METHODS
AND COMPARE RESULTS

SAMPLE ESTIMATION METHODS

Estimate by analogy

Counting

Expert judgement

Decomposition

Group reviews

Estimation checklist

ESTIMATION BY ANALOGY IS USUALLY THE BEST METHOD, BUT REQUIRE CALIBRATION

TAKE INTO ACCOUNT ANY CHANGES:

- IS THE TECHNOLOGY THE SAME?
- IS THE DOMAIN THE SAME?
- IS THE TEAM THE SAME?
- ARE THE TOOLS THE SAME?
- IS THE WORKING ENVIRONMENT THE SAME?
- WERE THE ESTIMATES MADE BY THE SAME PEOPLE?

IF POSSIBLE BASE ESTIMATION

FIRST ON COUNTING,

NOT JUDGING

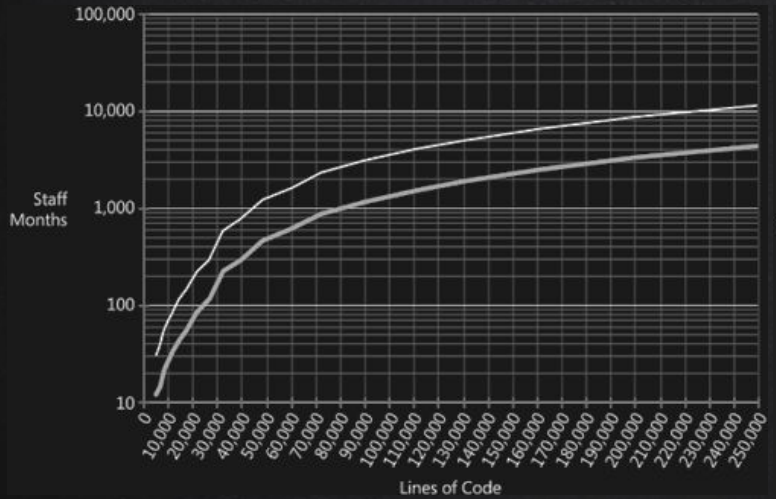
CONVERT COUNTS TO ESTIMATES USING:

- CURRENT PROJECT DATA
- COMPANY HISTORICAL DATA
- INDUSTRY DATA

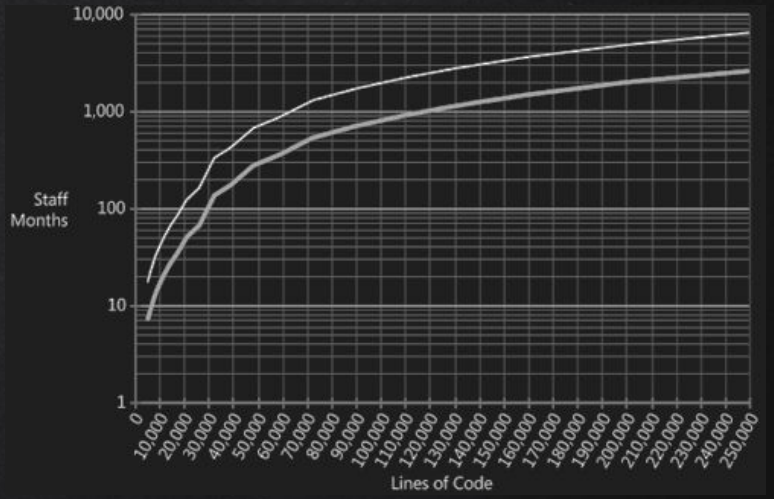
company data accounts for organizational influences – both recognized and unrecognized

current project data accounts additionally for team and software type influences

SAMPLE INDUSTRY DATA IS AVAILABLE



embedded systems



telecommunication systems

USE IT AS A HELP ONLY!

WHAT CAN BE COLLECTED FOR COUNTING

- LINES OF CODE
- DEFECTS
- API ENDPOINTS
- GUI ELEMENTS
- DOMAIN OBJECTS
- TEST CASES

“

The best time to start collecting historical data was years ago.
The second best time is now.

Chinese proverb paraphrase

DECOMPOSE
ESTIMATED FEATURES
INTO SMALLER

SMALLER FEATURES BENEFITS IN:

- BETTER FOCUS ON ESTIMATED FEATURE
- AVOIDING FORGOTTEN WORK
- BETTER SCOPE MANAGEMENT
- OPERATION OF THE LAW OF BIG NUMBERS

LAW OF BIG NUMBERS:

THE ERRORS ON THE HIGH AND LOW SIDE CANCEL EACH OTHER TO SOME DEGREE

"INVEST" FEATURES

- INDEPENDENT
- NEGOTIABLE
- VALUABLE
- ESTIMABLE
- SMALL
- TESTABLE

proposed by Will Wake

ESTIMATES SHOULD BE PREPARED BY PEOPLE WHO WILL ACTUALLY DO THE WORK

EXPERT JUDGEMENT IS SPECIALLY USEFUL WHEN:

- VERY EXPERIENCED EXPERT IS PRESENT
- TECHNICAL OR DOMAIN KNOWLEDGE IS VERY LIMITED

PLANNING POKER

- WHOLE TEAM INVOLVED
- ESTIMATE SIZE OF EACH FEATURE INDIVIDUALLY
- TALK DIFFERENCES
- REESTIMATE
- ARRIVE TO A CONSENSUS

IF REPEATING ISSUES
ARE OBSERVABLE
INTRODUCE ESTIMATION CHECKLIST

SAMPLE CHECKLIST QUESTIONS

Does the estimate include all the kinds of work?

Does the estimate contains best and worst case?

Were nonfunctional requirements defined and applied in estimate?

Was the estimated task small enough?

Are there some more risks?

ESTIMATE SIZE,
NOT THE EFFORT

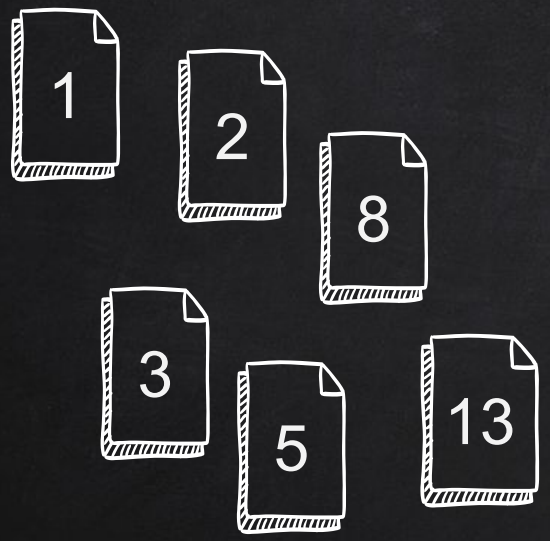
BENEFITS OF SIZE ESTIMATES

- EASIER
- HELPS TO DRIVE CROSS FUNCTIONAL BEHAVIOUR
- REDUCES PARKINSON'S LAW INFLUENCE

MEASURES OF SIZE

- "T-SHIRT SIZES"
- STORY POINTS
- IDEAL DAYS
- USER STORIES

STORY POINTS



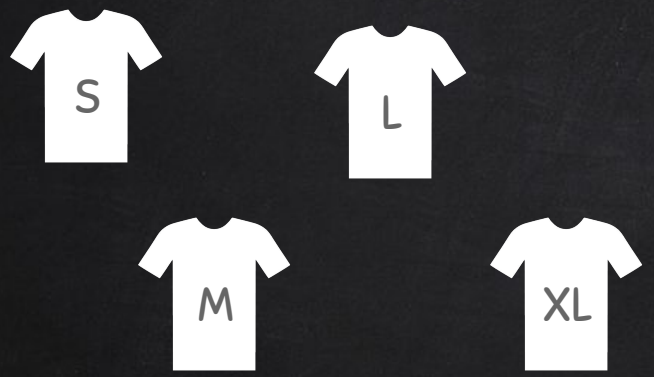
relative measure of size

relative estimation is easier than giving absolute number

values are Fibonacci numbers or powers of 2

non-linear sequences work well because they reflect the greater uncertainty associated with estimates for larger units of work

T-SHIRT SIZES



similar to story points but eliminates a desire to relate them to the time periods

easier to understand by stakeholders

a bit more sophisticated algorithm needed during iteration planning

IDEAL DAYS

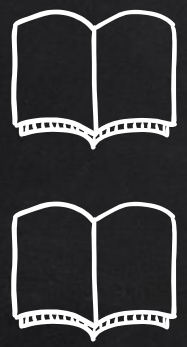
estimate of a time needed for a task completion with 100% focus on it

as in real world such focus is not possible it is rather a size than a time measure

not as much objective as story points (my ideal day is not your ideal day)

easier to understand outside of the team

USER STORIES



"#NoEstimates" approach proposed by Vasco Duarte – user stories count can be used as a measure of size

requires all requirements to be decomposed to the stories of similar size

enables the development team to focus rather on the user story quality than on its estimate

SWITCHING TO #NoEstimates

1. MOVE TO STORY POINTS FIRST
2. ONE-BY-ONE ELIMINATE SOME PLANNING POKER CARDS FORCING MORE FEATURES DECOMPOSITION
3. FINISH WITH ONE CARD ONLY

WHEN THE PROJECT IS RUNNING DO NOT ESTIMATE AN EFFORT, FORECAST INSTEAD

FEEL THE DIFFERENCE

"HOW LONG THE PROJECT WILL TAKE?"

"GIVEN THE RATE OF PROGRESS SO FAR, AND THE AMOUNT OF WORK STILL LEFT, WHEN WILL THE PROJECT END?"

VELOCITY

- NUMBER OF STORY POINTS AVERAGELY COMPLETED DURING ITERATION
- AFTER A FEW ITERATIONS ALLOWS TO CONVERT FEATURE SIZE ESTIMATES INTO TIME FORECAST
- IN #NoEstimates APPROACH IT IS NUMBER OF STORIES PER ITERATION

USE OF VELOCITY IN RELEASE PLANNING

- FOR TIME-BASED RELEASE

  *story_points_in_scope = number_of_iterations * velocity*

- FOR SCOPE-BASED RELEASE

  number_of_iterations = sum_of_story_points_in_scope / velocity

SCHEDULE BUFFER

- FOR EACH TASK PREPARE EXPECTED CASE (a_i) AND WORST CASE (w_i) ESTIMATE
- TAKE BUFFER EQUAL TO 2 STANDARD DEVIATIONS: $2\sigma = \sqrt{(w_1 - a_1)^2 + (w_2 - a_2)^2 + \ldots + (w_n - a_n)^2}$
- TAKE BUFFER 20% IF CALCULATED VALUE IS LOWER

BUFFER CALCULATION

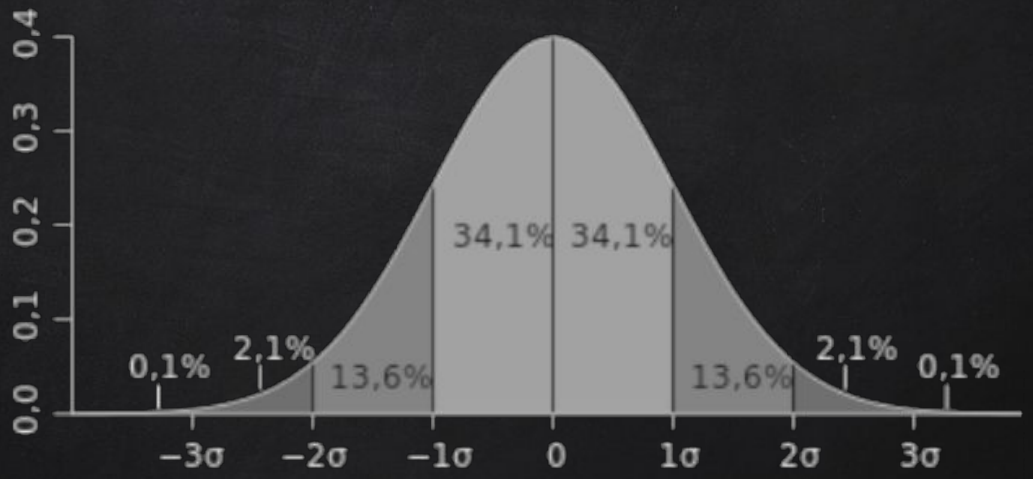
	expected case	worst case
Story 1	3	5
Story 2	5	8
Story 3	8	8
Story 4	5	13
Story 5	5	8
Story 6	2	8
Story 7	1	3
Story 8	8	8
Σ	37	61

$$2\sigma = \sqrt{((5^2-3^2)+(8^2-5^2)+(8^2-8^2)+(13^2-5^2)+(8^2-5^2)+(8^2-2^2)+(3^2-1^2)+(8^2-8^2))}$$

$$\approx 17$$

(46%)

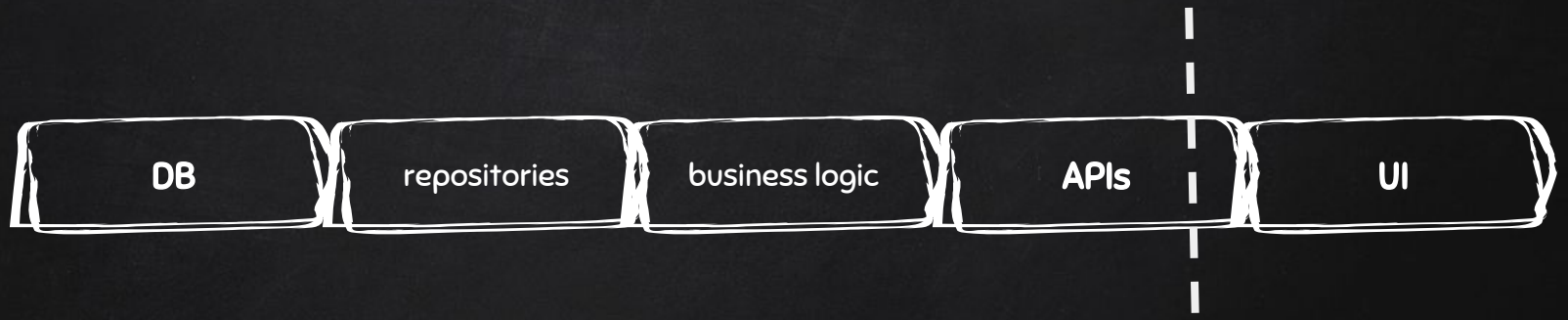
BUFFER CAPACITY VS STANDARD DEVIATION



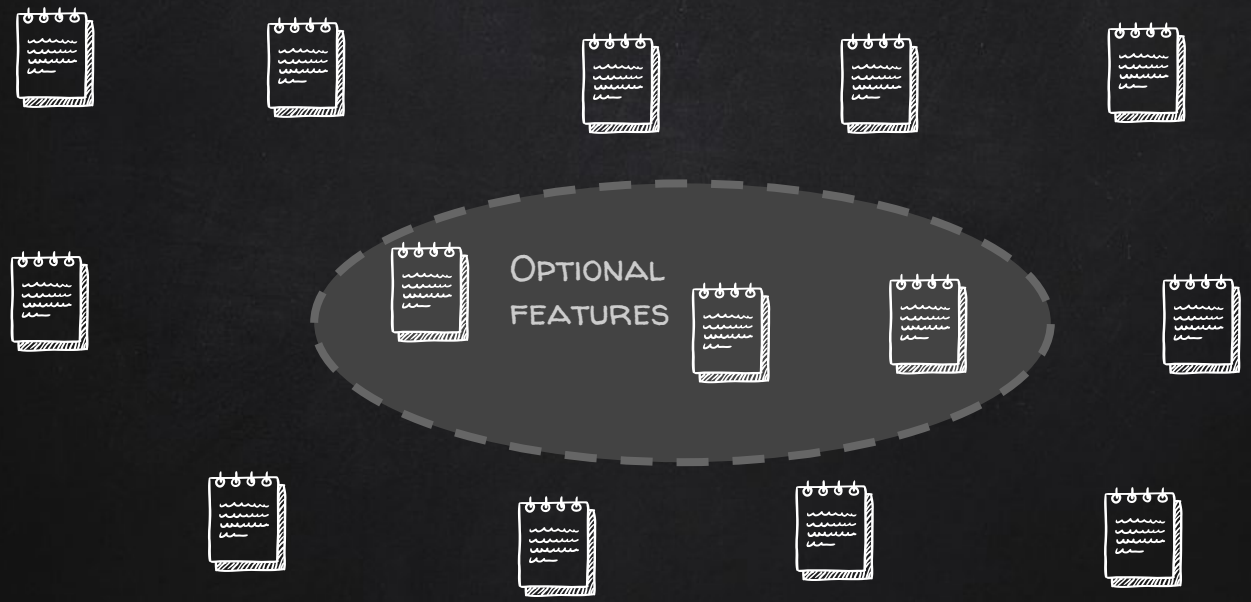
SCOPE BUFFER

- "MUST-HAVE" FEATURES < 70% OF ALL
- PRIORITIZE BEFORE ESTIMATING TO ENSURE DELIVERING MVP FIRST
- FOCUS ON PREPARING A BACKLOG OF WELL-PREPARED CROSS-FUNCTIONAL AND INDEPENDENT FEATURES

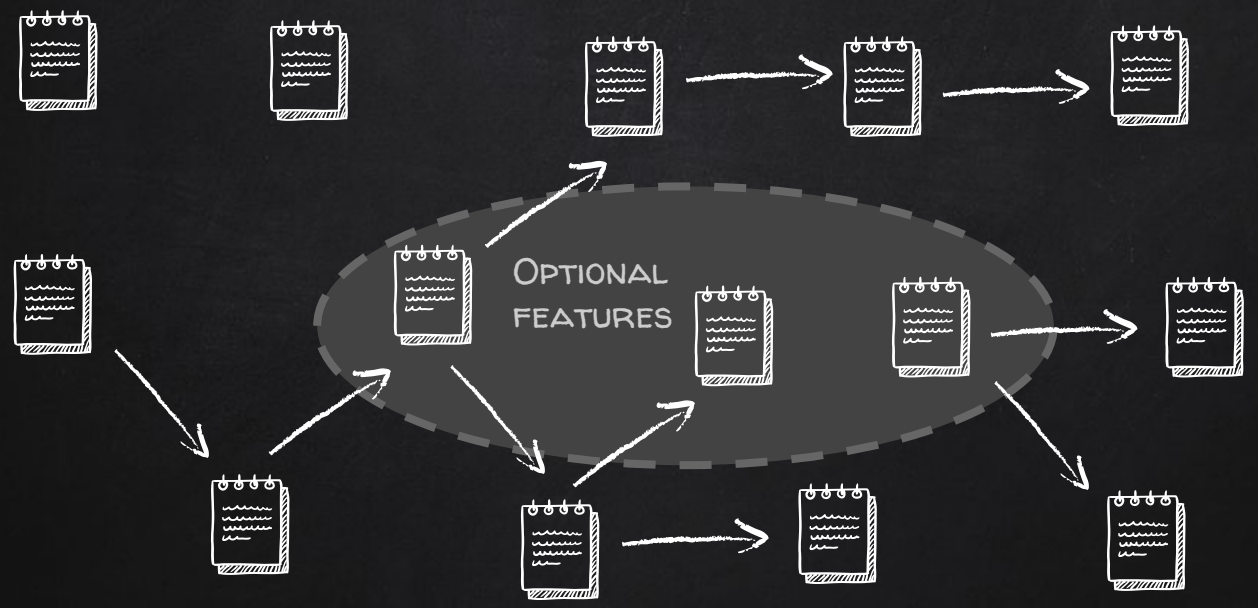
CUTTING SCOPE FAIL ON NON-CROSS-FUNCTIONAL FEATURES



CUTTING SCOPE FAIL ON DEPENDENCIES



CUTTING SCOPE FAIL ON DEPENDENCIES



RELEASE SCOPE MANAGEMENT IS MORE EFFICIENT THAN TIME MANAGEMENT

USUALLY 30-40% FEATURES DELIVERS 90% OF CUSTOMER VALUE

DEADLINE AND PREDEFINED SCOPE ARE NOT A VALUE, PRODUCT SUCCESS IS A VALUE

WHEN YOU ARE FINALLY HAPPY WITH YOUR ESTIMATES TAKE INTO ACCOUNT

HOFSTADTER'S LAW:

IT ALWAYS TAKES LONGER THAN YOU EXPECT, EVEN WHEN YOU TAKE INTO ACCOUNT HOFSTADTER'S LAW ☺

Thank you!

Any questions?

Wojciech Jaworski
jaworski.wojciech@gmail.com

NICE BOOKS ☺

- Steve McConnell "Software Estimation: Demystifying the black art"
- Mike Cohn "Agile Estimating and Planning"
- Vasco Duarte "No Estimates"