Specification – the Everyday Madness

The sense and non-sense in the processes of specification

- What is more important than specifications?
- Who are the people who usually do systems engineering and how are they educated?
- Are there other ways to educate different people to be even more efficient in systems engineering task?

The Problem
Part 1: The customer

Has a problem

Doesn’t know exactly what he needs

Doesn’t have right expertise

Has unrealistic expectations on timeline

… but has money to spend.

The Problem
Part 2: The systems-engineer

Develop technical solutions

Technical expertise

Wants to bring good products on market

Wants to be ready on time

.. wants to earn money

How does this fit togehter?

Customer and systems-engineer seem to be a match:

1) Systems Engineer is able to deliver what the customer needs

2) Customer is able to pay money to the systems engineer.

.. problem:

time and budget don’t fit

Different ways to reduce these problems

- Models and methods to prevent engineers to just follow their assumptions and own ideas
- Contracts crafted to define and maintain the boundaries
- Milestones defined and coupled with payment installments

Do “models” help?

- **Waterfall**
  - validation of the developed systems against the customer expectations only in the end
    - Wasted time and effort – frustration to the customer
      - Project failure: more than 80 % out of time and/or budget
- **V-model**
  - Multiple intermediate verification and validation iterations during the development
    - Better predictability of overall project success probability
      - Project success ~7ß% to ~85%
- **Agile**
  - Very fast feedback from customer but often lacking performance, different developers do/redo in big loops as information is missing
    - Faster customer “satisfaction” - real results after iteration steps
      - Project success 65% to 75 %
- **MBSE**
  - Starts with architecture and functional modeling. Mostly without the performance context and aside of the customers scope
    - In theory iz should work …
      - Suceess reate should be highe according to the model calculations
- **……. <next Buzz-Word please>**

None of them really work! But why?

Developers view

Start of the project:

“Requirements? No thanks.”

“Just start developing – I know what to do.”

A bit later:

“If I only could start over with the gained knowledge...”

In the end:

“…... why the heck have I been to university when it doesn’t help getting this done?”

BURN-OUT

Results of the current practice

Tons of documents

Loads of redundant information

Leading to unknown amount of false information

+

more than 60% information garbage due to words not adding value

Remember?

Garbage in, garbage out!

If systems engineers would apply systems engineering:

The challenge:

Define what the system shall do under which conditions

Describe a possible solution

Decide what needs to be done and who is capable of doing this best.

Tasks and Benefits

Extract the requirements from customer's brain

Extract the know how from engineer's brain

Use "requirements" as blackbox for know-how

"know-how" is internal capital

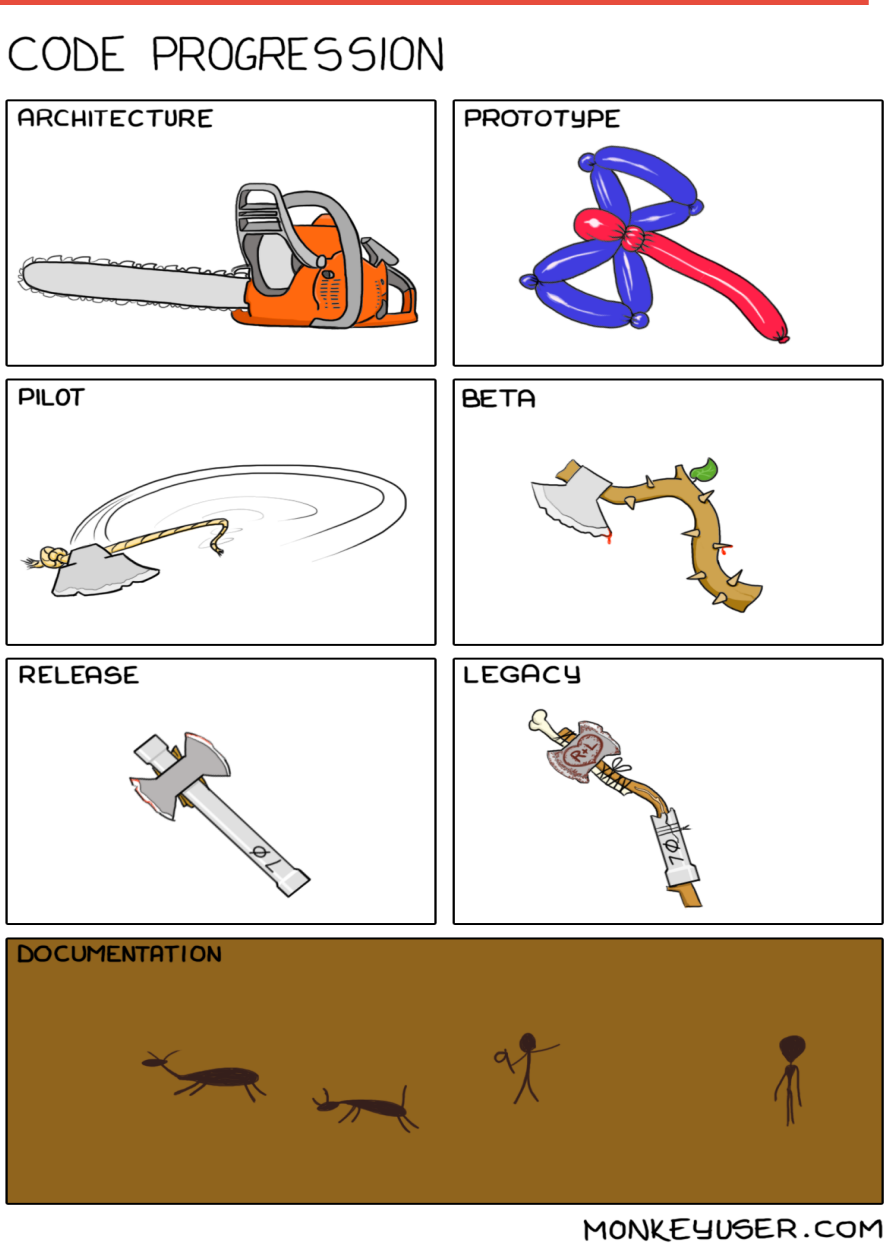
"Know-what" wins the project

The unconvenient truth

No organization needs

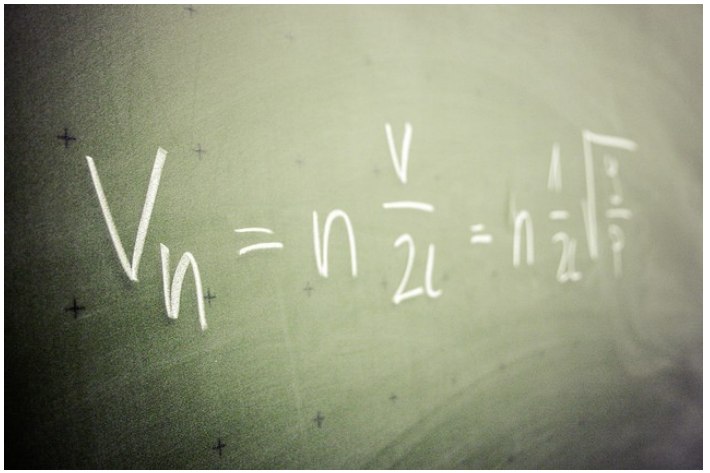
engineers

who write requirements.



Linguists and Formalists

Other people than engeineers can do a better job on requirements!



- **Linguists**
- **Legal Professionals**
- **Mathematicians**
- **People with special abilities**

Consequence for education and employment

Requirements = Crafting != Sience

Everyone can do it who is able to

- understand the language

- think in a logical and structured way

- imperturbably work a formalized way

- listen like a Sherlock Holmes

- stubborn enough to ask questions until they are answered

Your turn to ask now ....

Eckhard Jokisch

Passion on making things work by formalized simplification.

Contact:

eckhard.jokisch@orangemoon.ie

https://www.orangemoon.ie