



Anti-Patterns of Requirements Reverse Engineering

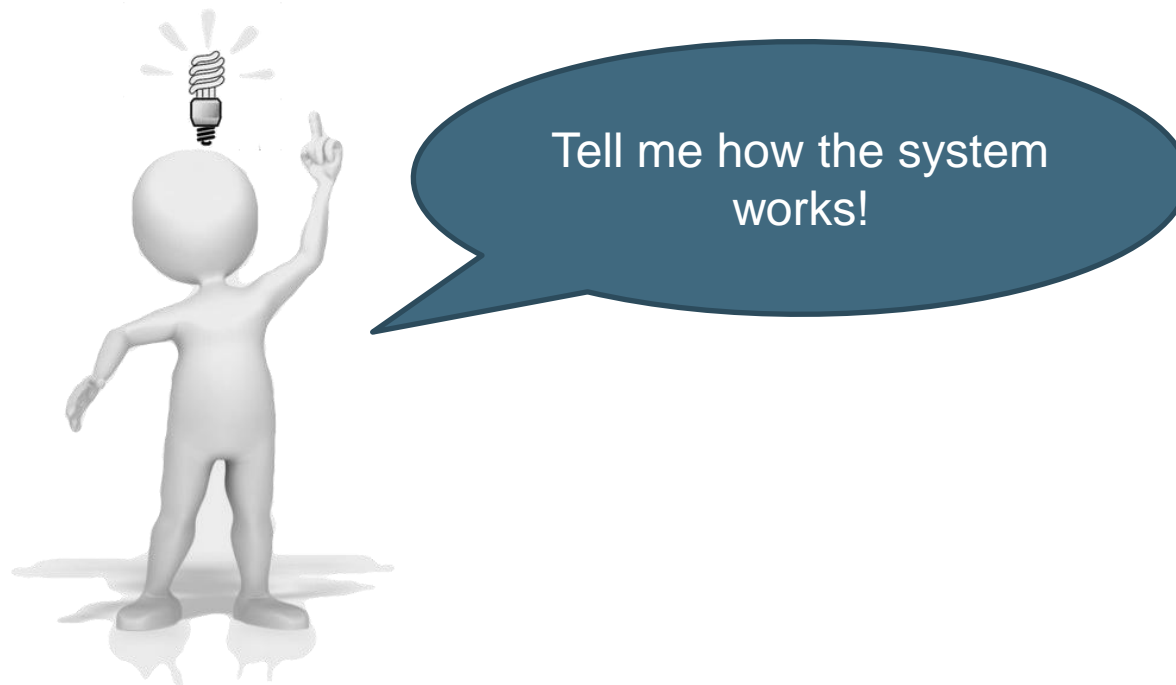
by Stanislav Rozhdestvenskiy, Nikolay Sokolovskiy



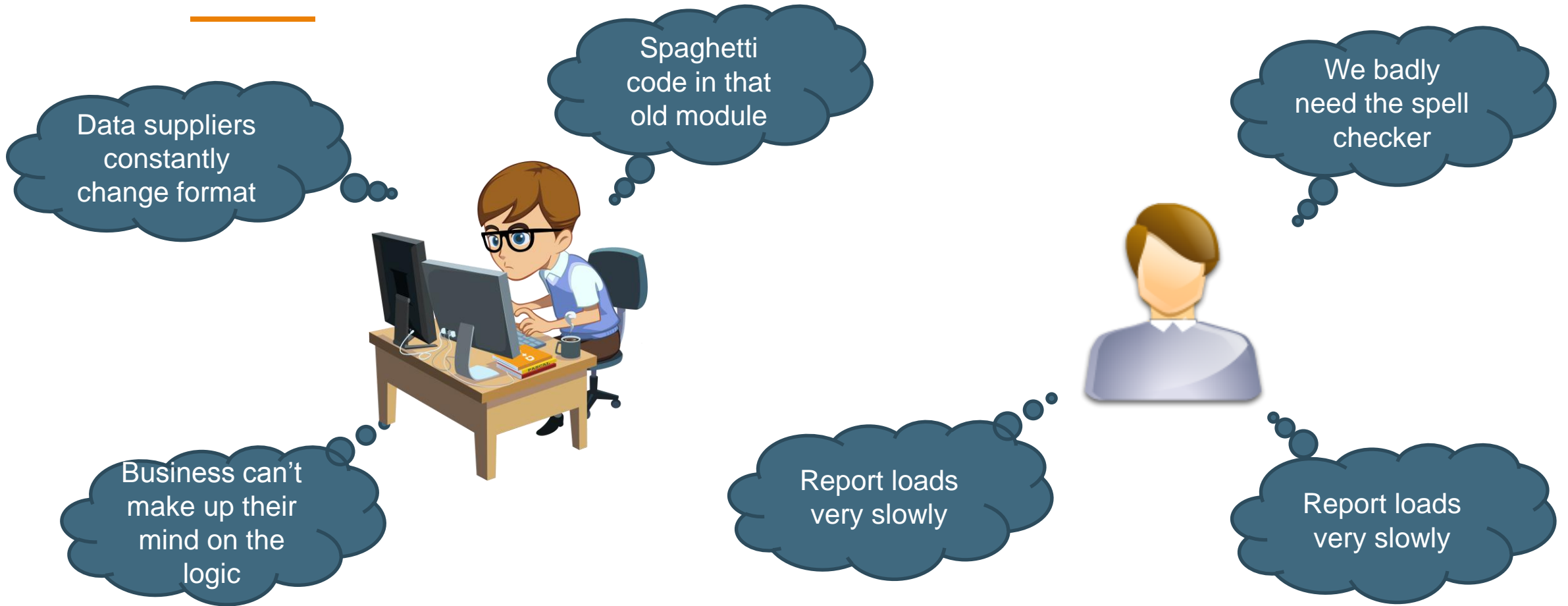
Anti-pattern 1: Ask SME how system works

Pre-conditions:

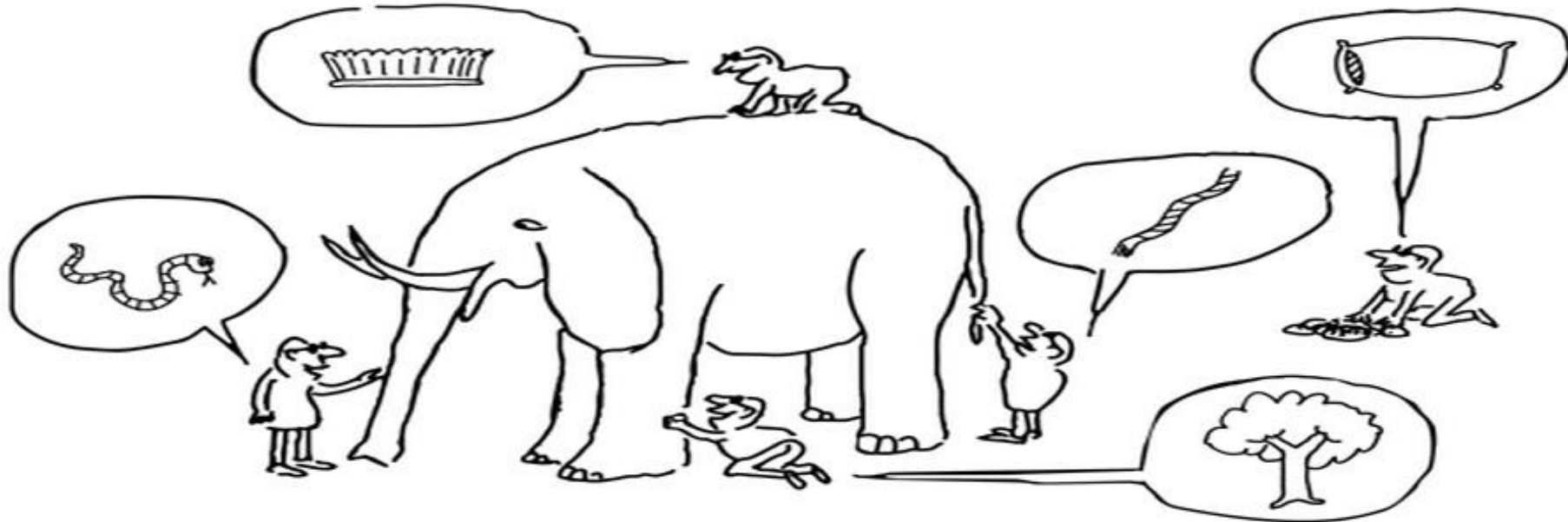
- You are new to the system
- The system is in use (test or production)
- There is a technical or business SME who's dealing closely with it



Pain points instead of requirements



Disconnected aspects instead of the full picture



Anti-pattern 2: Read the code

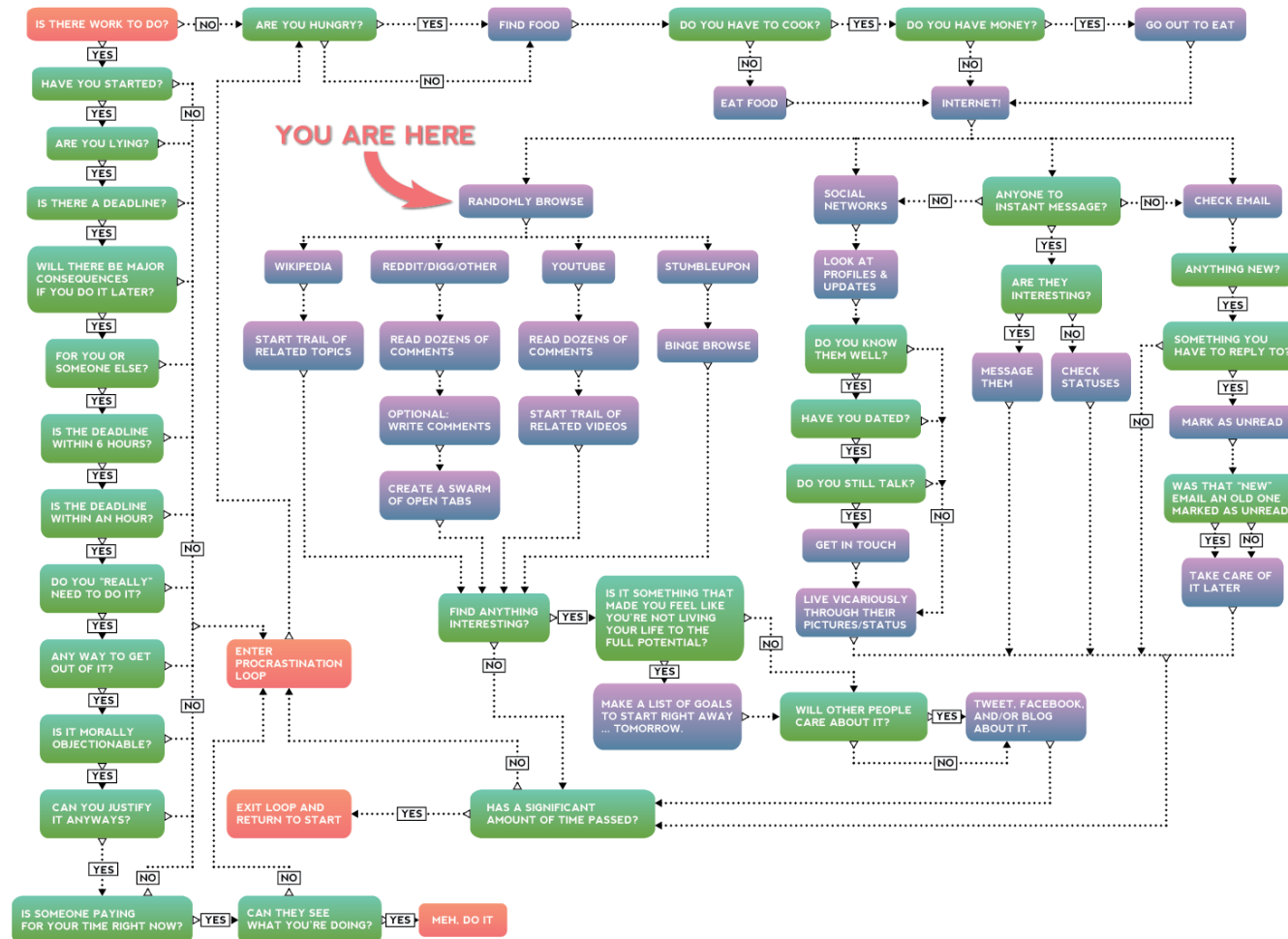
Pre-conditions:

- You have access to the source code
- You can read the code and there is a developer who can read it for me



Would you please read the code and lay down its logic on a nice diagram?

Unreadable diagram instead of clear business flows



How to deal with technical SMEs right: Structured interview approach

Q: Let's say you we've just installed the system on production. What should happen for it to start working?

A: Well, we need to set up some parameters

Q: What kind of parameters?

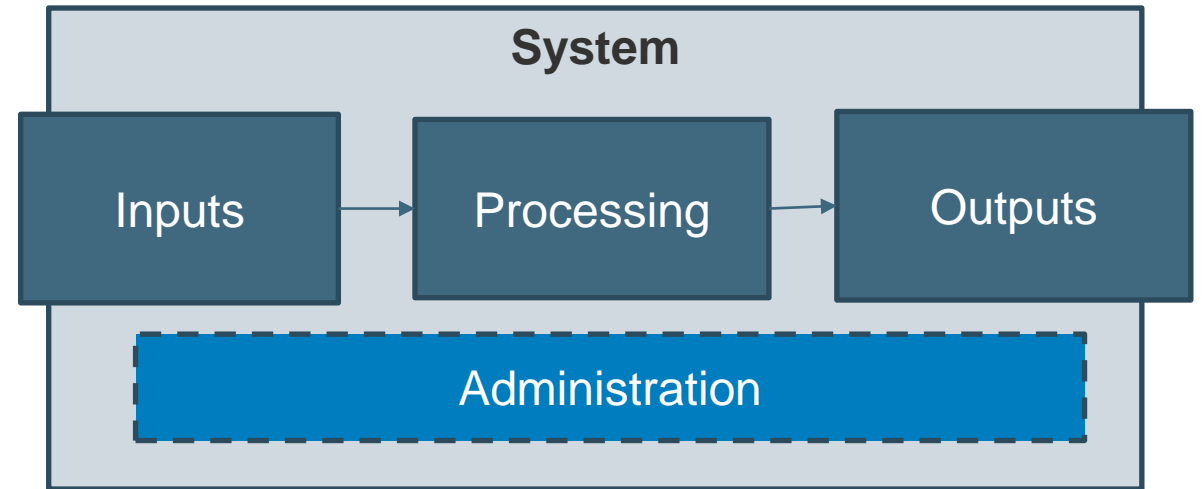
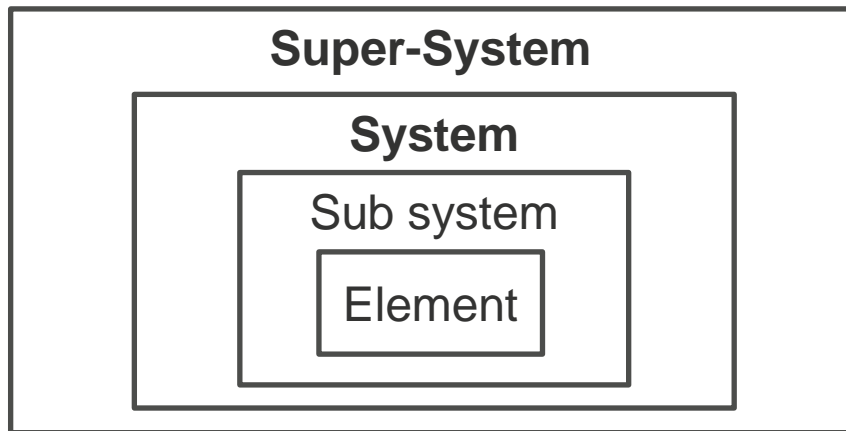
A: ...

(.....)

Q: So, after we set all the system parameters, how do we start working with it?

A: We need to receive data from... / The users should login and start entering...

All systems have something in common:
use system patterns to structure findings



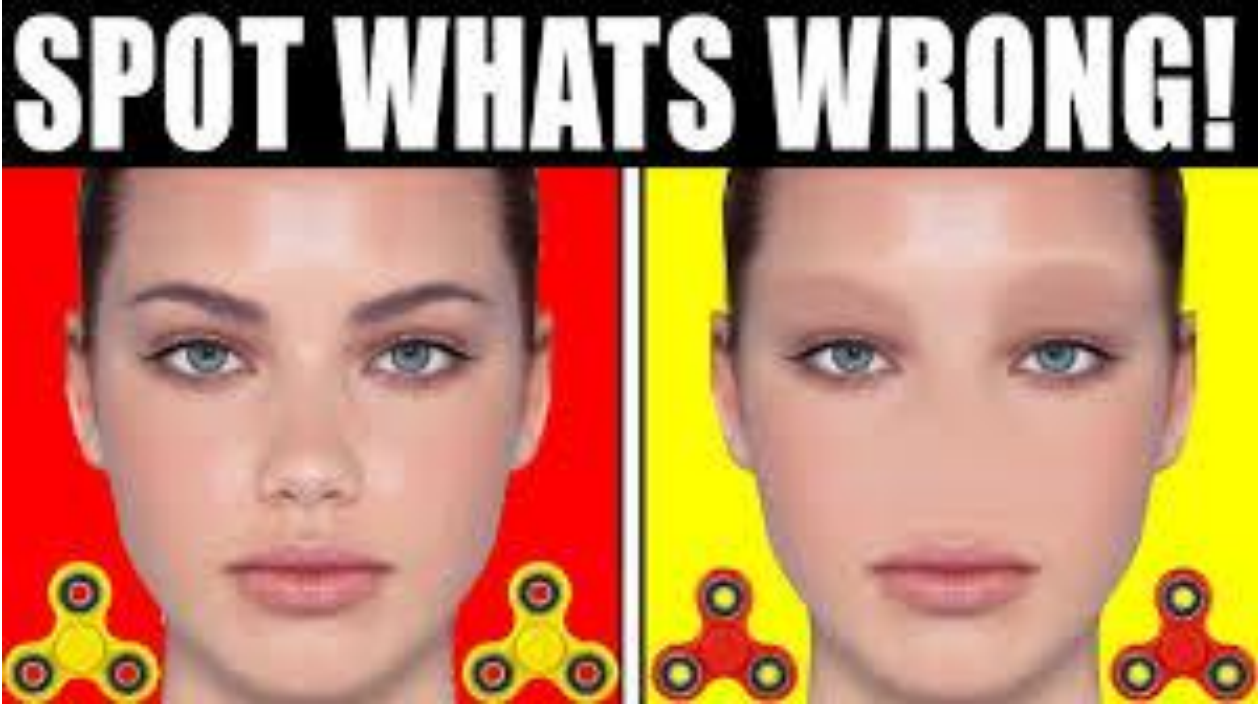
Anti-pattern 3: Sabotage

Pre-conditions:

- We need to replace the legacy system
- Knowledge of that system is monopolized by their support / developers



Anti-pattern 3: Sabotage Results



Anti-Pattern 4: All swans are white



Pre-condition:

- We met similar processes / systems before



Anti-Pattern 4: All swans are white

Results



Use other sources and double-check everything



Don't cut a time for investigations of:

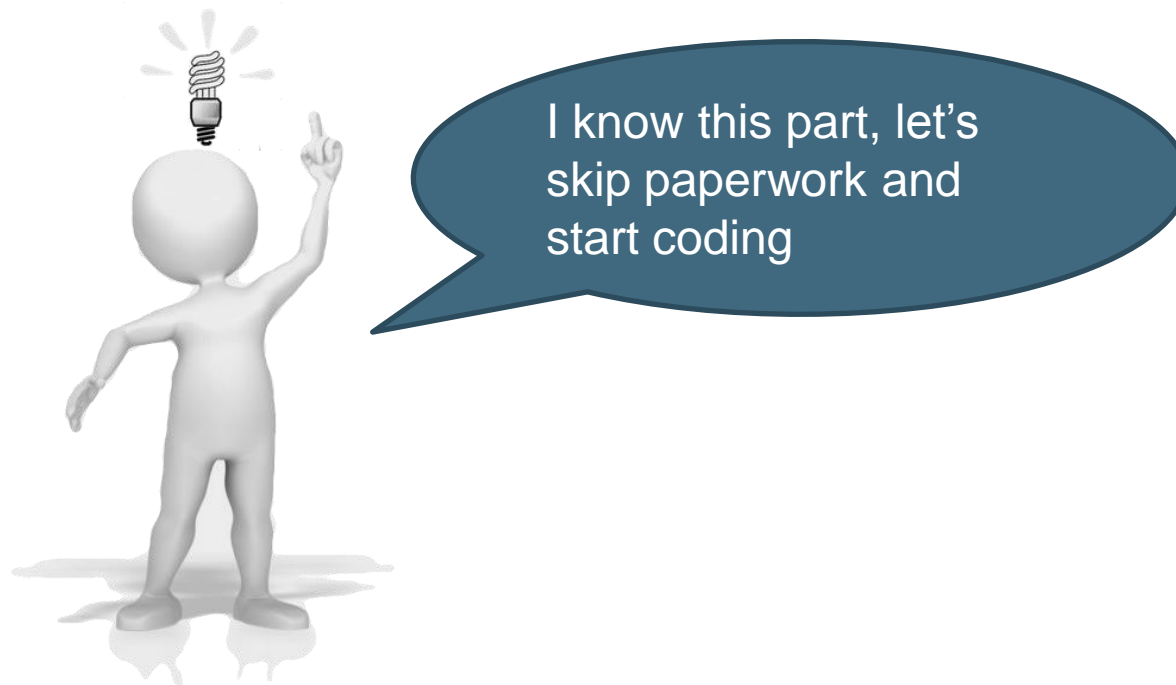
- ✓ Business Process
- ✓ Scenarios of legacy system (via testing)
- ✓ Ask about rare functions (“what will you do in the end of year?”)
- ✓ Ask about “impossible” situations (“What will you do, if a data in a report looks strange?”)

For more info on GAP Analysis please refer to “*Gap-analysis for the implementation of generic solutions*” (<http://analystdays.com/en/talk/44908>)

Anti-Pattern 5: Knowledge in the head

Pre-condition:

- You've got some knowledge of the legacy system
- You're in a rush to start building the new solution



Legacy solution has changed, while you were studying it



Document everything and keep it up-to-date



Don't just capture a snapshot, capture updates too:

1. Make sure you log all updates first
2. Then develop the complete specification
3. Merge the updates to the specification when appropriate

For more details – see Analyst Days 2016 «*Approaches to changes specification*» <http://analystdays.ru/ru/talk/40286>

Thank you!

Also on the topic:

- Recovering of data structure in details: “*Reverse engineering of data structure: cases and approaches*” (<http://analystdays.com/en/talk/52239>)
- More to come... keep in touch!



Соколовский Николай

Email: sokolovskynik@gmail.com

Facebook: <https://www.facebook.com/sokolovskynik>



Станислав Рождественский

Email: stigro@gmail.com

LinkedIn: <https://www.linkedin.com/in/sirojdestvensky>