



## **W14**

Test Techniques

Wednesday, May 2nd, 2018

3:00 PM

# **The Four Cs and One T of Requirements â€œTestingâ€**

Presented by:

**Evgeny Tkachenko**

EPAM Systems

Brought to you by:



350 Corporate Way, Suite 400, Orange Park, FL 32073  
888-268-8770 · 904-278-0524 - [info@techwell.com](mailto:info@techwell.com) - <http://www.stareast.techwell.com/>

# **Evgeny Tkachenko**

## **EPAM Systems**

Evgeny Tkachenko has more than eight years of experience in test management, automation testing, and release management, mainly on complex projects in the telecommunication, financial, and online entertainment industries. Previously his experience as a test manager in Russia's most important technology center gave Evgeny a unique perspective in test process development. Now at EPAM Systems in the United States he applies his experience and expertise to do whatever is required to deliver the highest quality product possible. Evgeny is a frequent speaker at QA/Dev/Agile conferences in the US and Europe, and an author for international test and QA publications.

# The Four Cs and One T of Requirements Testing



# The Four Cs and One T of Requirements "Testing"

EVGENY TKACHENKO

# Evgeny Tkachenko



## QA Consultant

- 10+ years of engineering experience in automation testing and test management
- experienced speaker at QA/Dev/Agile conferences
- author of articles for International Magazines

Video clip

## Agenda:

1. Requirements analysis
2. Practice
3. Software testing heuristics
4. Requirements Test Driven Development
5. How to reach success in Requirements "Testing"

Everybody says they test requirements



How exactly do you test your requirements?

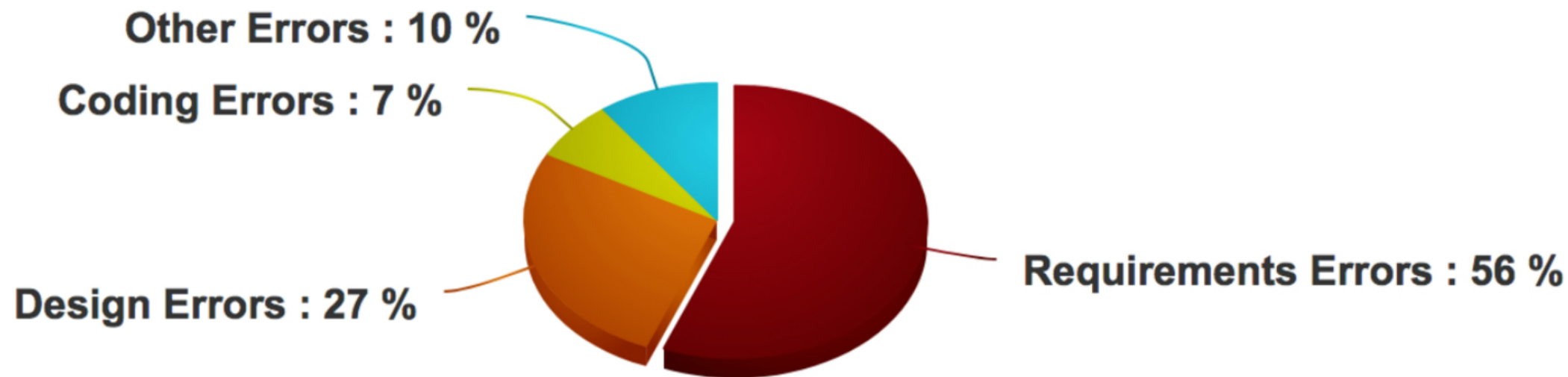






Everybody talks about it, but nobody does anything about it

# The Source of Software Defects

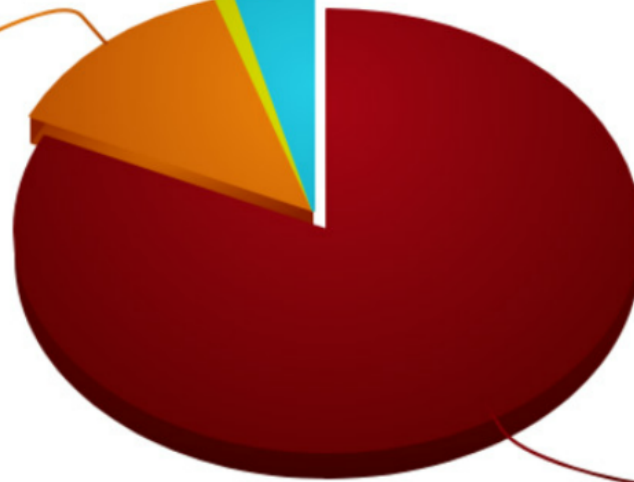


# The Cost to fix Software Defects

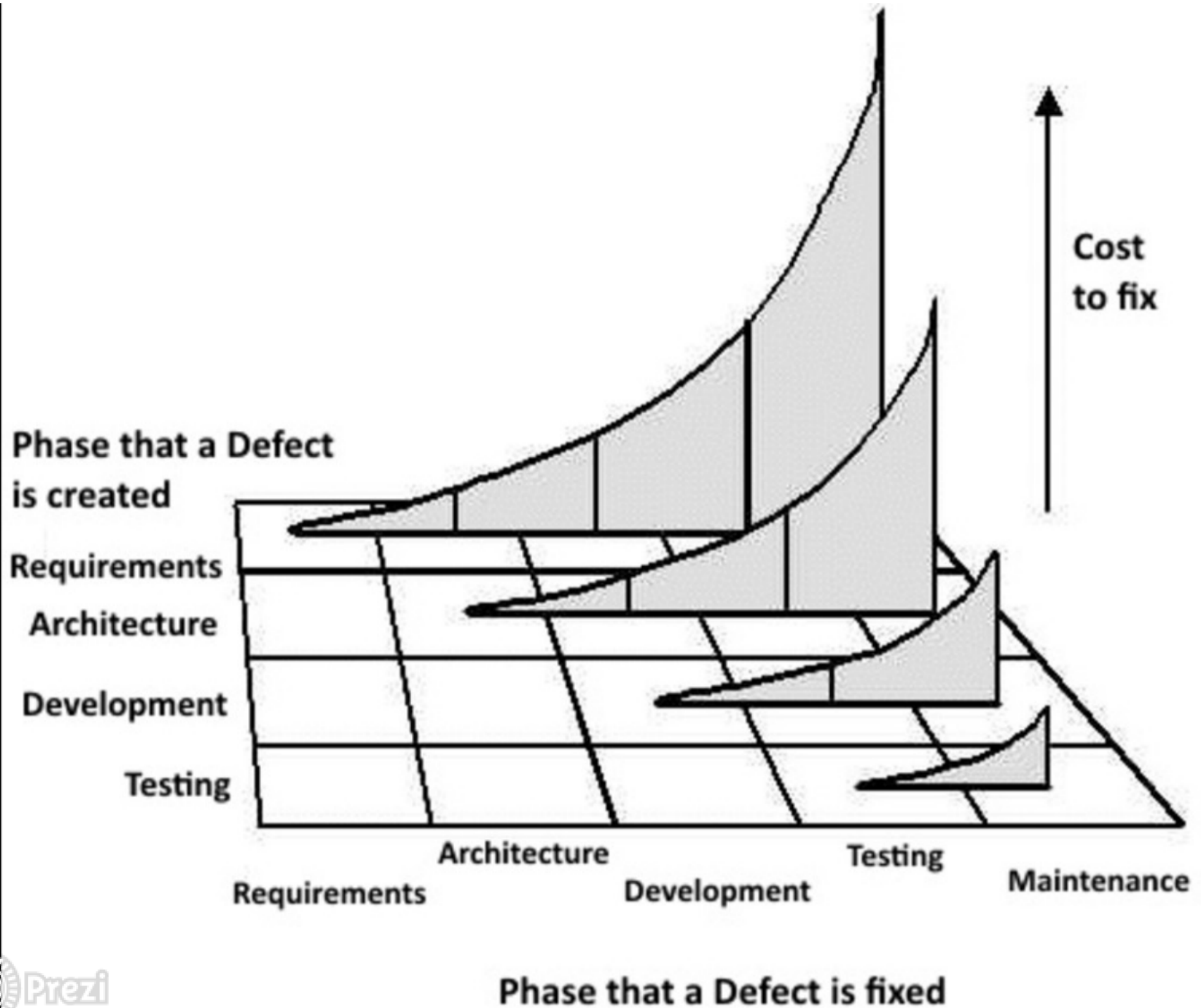
Other Errors : 4 %

Coding Errors : 1 %

Design Errors : 13 %



Requirements Errors : 82 %





# The Four Cs and One T of Requirements Testing

Completeness, correctness, clearness, consistency, testability

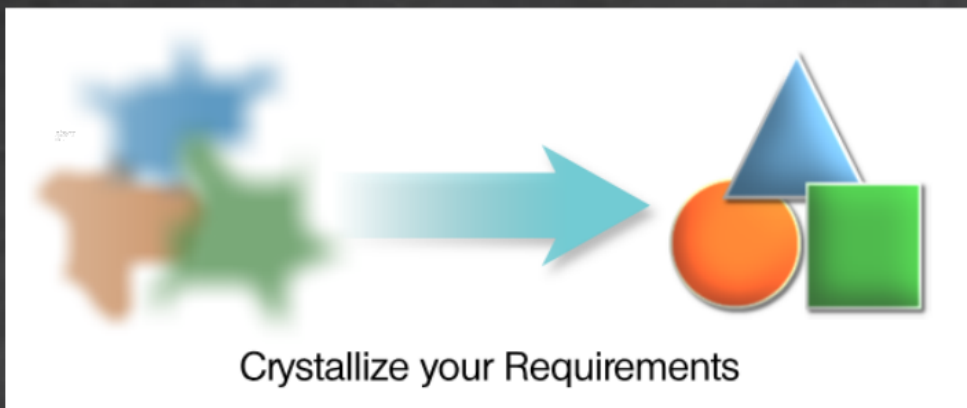
Crystallize your Requirements

LET'S PLAY A GAME

Heuristics

Who does heuristics

# Completeness, correctness, clearness, consistency, testability



- 1
- 2
- 3
- 4
- 5

## Completeness

Check that all possible scenarios have been considered in a requirement, try to find any gaps or uncovered cases

## Correctness

All statements should be correct, truthful and logical

## Clearness

Requirements should be transparent and clear to everyone. Try to find all ambiguities in requirements, only generally accepted terms should be used

## Consistency

Requirements should not contradict other requirements and fully comply with external documentation and with internal and/or external standards

## Testability

There should be a way to check if implementation meets a requirement or not

## Requirements example

When a user enters a valid email address in the field and clicks on a submit button he should see a message about successful registration. If a user enters an invalid email address then he has to see an error message in a user-friendly format. When a user puts a cursor in the text field a "submit" button (related to this field) should become disabled.

# Completeness

Check that all possible scenarios have been considered in a requirement, try to find any gaps or uncovered cases

What should we do if a user leaves this field empty and clicks the button?

E-mail

❗ Please enter email address

**Create account**

What should we do if a user with this email address already exists in the system?

evgenyamazon@gmail.com

❗ Email address already in use

**Create account**

"Great Sparrow Campaign". The biggest  
mistake due to incompleteness



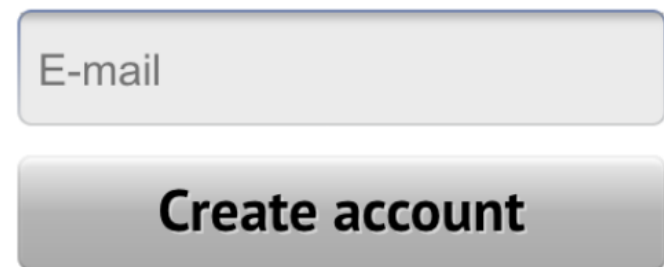
# Correctness



All statements should be correct,  
truthful and logical

# Requirements example

When a user enters a valid email address in the field and clicks on a submit button he should see a message about successful registration. If a user enters an invalid email address then he has to see an error message in a user-friendly format. When a user puts a cursor in the text field a "submit" button (related to this field) should become ~~disabled~~ **enabled**.



The image shows a user interface element consisting of two vertically stacked rounded rectangular buttons. The top button is light gray and contains the text "E-mail". The bottom button is a darker gray and contains the text "Create account".



Do you remember this guy?



The most famous typo.  
Spinach is one of the most  
iron-rich vegetables



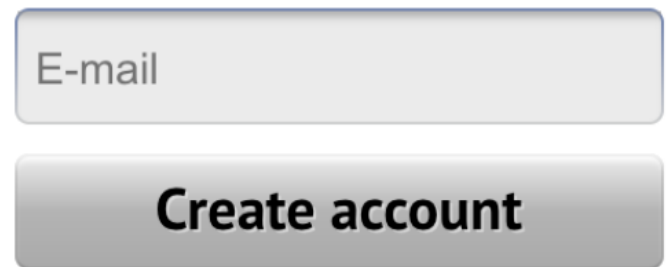
# Clarity



Requirements should be transparent and clear to everyone. Try to find all ambiguities in requirements, only generally accepted terms should be used

# Requirements example

When a user enters a valid email address in the field and clicks on a submit button he should see a message about successful registration. If a user enters an **invalid email address** then he has to see an error message in a user-friendly format. When a user puts a cursor in the text field a "submit" button (related to this field) should become disabled.



The image shows a user interface element consisting of two vertically stacked components. The top component is a light gray rounded rectangular input field with the placeholder text "E-mail". The bottom component is a darker gray rounded rectangular button with the text "Create account" in white, bold font.

# Mars Climate Orbiter.

One of the most expensive discrepancies



# Consistency



Requirements should not contradict other requirements and fully comply with external documentation and with internal and/or external standards.

All "Submit" buttons should be **blue** throughout the whole application



**Create account**

All "Submit" buttons should be blue throughout the whole application

**Create account**

evgenyamazon@gmail.com

❗ Email address already in use

**Create account**



**New French trains did not  
fit platforms  
One of the costliest  
inconsistencies**





There should be a way to check if implementation meets a requirement or not

# Requirements example

When a user enters a valid email address in the field and clicks on a submit button he should see a message about successful registration. If a user enters an invalid email address he has to see an error message in a **user-friendly format**. When a user puts a cursor in the text field a "submit" button (related to this field) should become disabled.

**One of the costliest bugs.  
No more long-distance calls**



Testable products are easy  
and less costly to maintain



How to automate testing of a registration functionality?

**By email**

evgenyamazon@gmail.com


128113  6 digits

 Please enter the security code


**Create account**

# The Four Cs and One T of Requirements Testing

Completeness, correctness, clearness, consistency, testability



Crystallize your Requirements



Heuristics

Who does heuristics



**LET'S PLAY A**



**GAME**



# 1st logic puzzle

# 2nd logic puzzle

# 3rd logic puzzle

# 4th logic puzzle

# 5th logic puzzle




1st logic puzzle

2nd logic puzzle

3rd logic puzzle

4th logic puzzle

5th logic puzzle



$RTDD = RT + ATDD$   
Begin with the end in mind

Tasks 7 Downstream Dependencies 1 Links 1 Attachments 0

PROJECT: RFPPro Release 12 (Feb2017)      BACKLOG GROUP:      STATUS: Accepted      PRIORITY: Must Have      TASK STATUS SUMMARY:    
SPRINT: **Sprint 53**      PORTFOLIO ITEM:      DATE COMPLETED: 19.09.2016 11:50      CLASS OF SERVICE: --      TEST STATUS SUMMARY:  
TEAM:      OWNERS: choward      ESTIMATE PTS.: 8,00  
RightFind Professional SUITE Team

DESIGN?: --      SPRINT READY?:   
PLATFORM VERSION: --      TESTABLE?:   
PRODUCT: Direct Path

DESCRIPTION:  
**Description**  
As CCC, I want to assign roles and/or permissions to a user belonging to an org and have the RFPPro application recognize all assigned permissions, so that users have access to what they were assigned.

**Acceptance Criteria**  
I will know this story is done when:  
1. I can assign a permission without a role in PRM and RFPPro behaves correctly for that permission.  
2. RFPPro makes a call to PRM to retrieve all permission and PRM returns all permissions - both assigned to a role and individual.  
3. RFPPro recognizes all roles assigned to a user, both at the org level and at the global level (ones without a PRM instance\_ID).

Test Data:  
org: CCC 2308baba- 6a0-ab00f3215c6

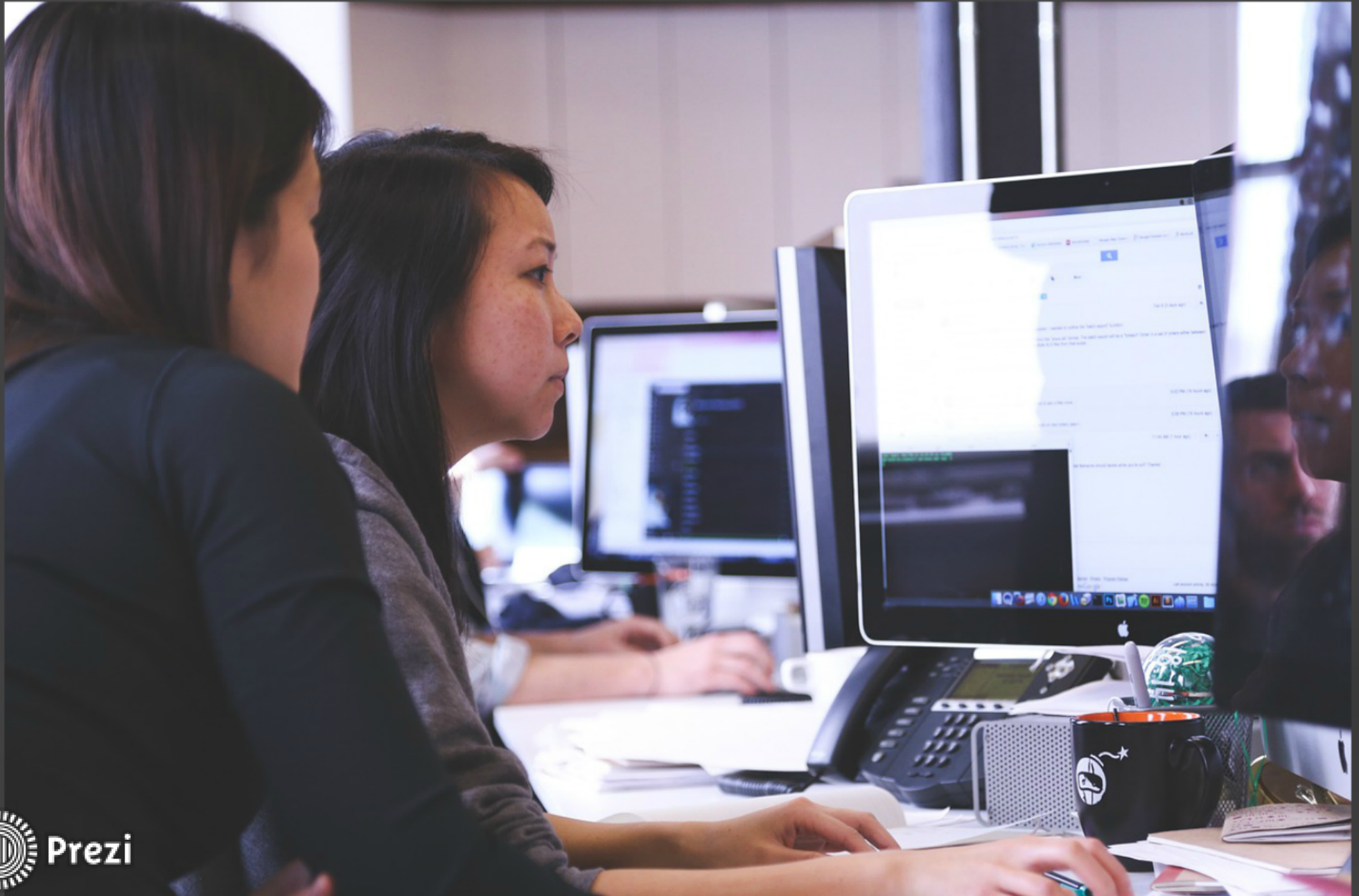
- st@copyright.com/  role RSP\_END\_USER\_WITH\_PURCHASE with an allowed permission RSP\_ACCESS\_ADMIN\_PORTAL
- ser1@copyright.com/  with no roles and with allowed permissions RSP\_ACCESS\_ADMIN\_PORTAL and RSP\_ACCESS\_AGREEMENT\_MANAGER
- we will not ignore roles and permissions which have been granted on User level. For rftesting.admin@copyright.com/  we have to have RSP\_END\_USER\_WITH\_PURCHASE role and an Admin role as well.

**Assumptions**

**Requirements**  
1. Use <method name> to retrieve all permissions assigned to a user within an org.  
2. Roll back the changes made as part of B-17903 so that the application recognizes all roles assigned to a user, and not just roles with PRM instance\_IDs.



Acceptance tests creation and test data preparation to help developers to check if their implementation fits acceptance criteria



For instance, we have an AC:

"A user with a positive balance on his account should have an ability to transfer the money to another account."

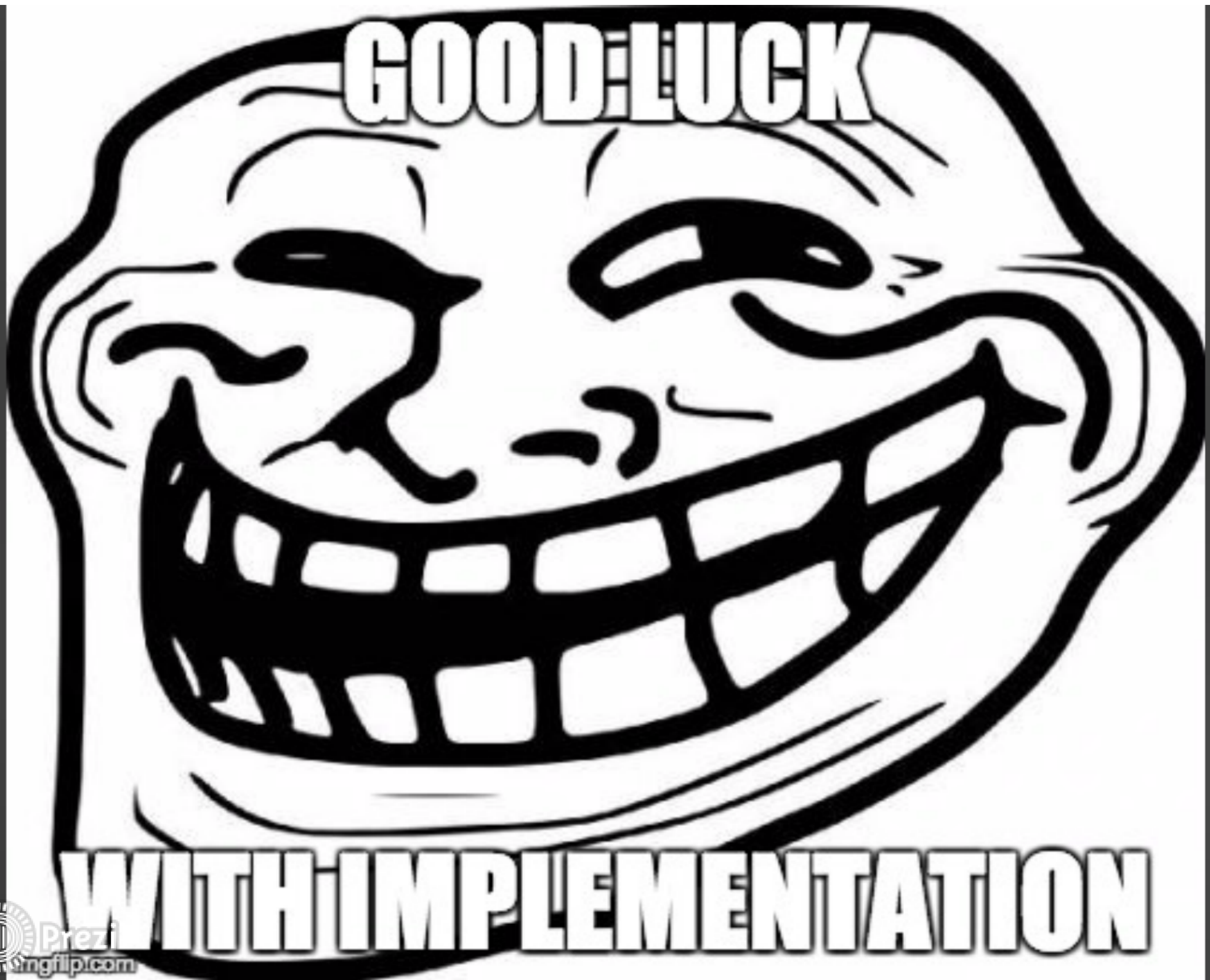
A QA engineer will have to prepare a user with a positive balance, one user with \$0 and one user with a negative balance and add this information to the story:

"A user1 (DEV) with a positive balance should have an ability to transfer his money."

"A user2 (DEV) with a \$0 balance must not have an ability to transfer his money."

"A user3 (DEV) with a -\$10 balance must not have an ability to transfer his money."

**GOOD LUCK**



**WITH IMPLEMENTATION**

Acceptance tests: essential and test case preparation to help developers to check if their implementation fits acceptance criteria



```
1 2014-03-18 10:00:00 [INFO] Starting application...
2 2014-03-18 10:00:01 [INFO] Database connection established.
3 2014-03-18 10:00:02 [INFO] Loading configuration from /etc/app/config.yaml
4 2014-03-18 10:00:03 [INFO] Application ready to serve requests.
5 2014-03-18 10:00:05 [INFO] Received request from 192.168.1.100
6 2014-03-18 10:00:06 [INFO] Processing request: GET /api/status
7 2014-03-18 10:00:07 [INFO] Response generated successfully.
8 2014-03-18 10:00:08 [INFO] Request completed.
9 2014-03-18 10:00:10 [INFO] Application running smoothly.
```



For instance, we have an AC:  
"A user with a positive balance on his account should have an ability to transfer the money to another account."  
A QA engineer will have to prepare a user with a positive balance, one user with \$0 and one user with a negative balance and add this information to the story.  
"A user1 (BEN) with a positive balance should have an ability to transfer his money."  
"A user2 (BEN) with a \$0 balance must not have an ability to transfer his money."  
"A user3 (BEN) with a -\$10 balance must not have an ability to transfer his money."



# Heuristics

Video about Heuristics

# Video about Heuristics

To make testing requirements more effective, you can use the approach called **heuristics testing**, or testing with a strategy that relies on past data about probabilities.

# SFDPO

Structure (what the product is)  
Function (what the product does)  
Data (what it processes)  
Platform (what it depends on)  
Operations (how it will be used)





# I SLICED UP FUN

I Inputs

S Store

L Location

I Interactions

C Communication

E Ergonomics

D Data

U Usability

P Platform

F Function

U User Scenarios

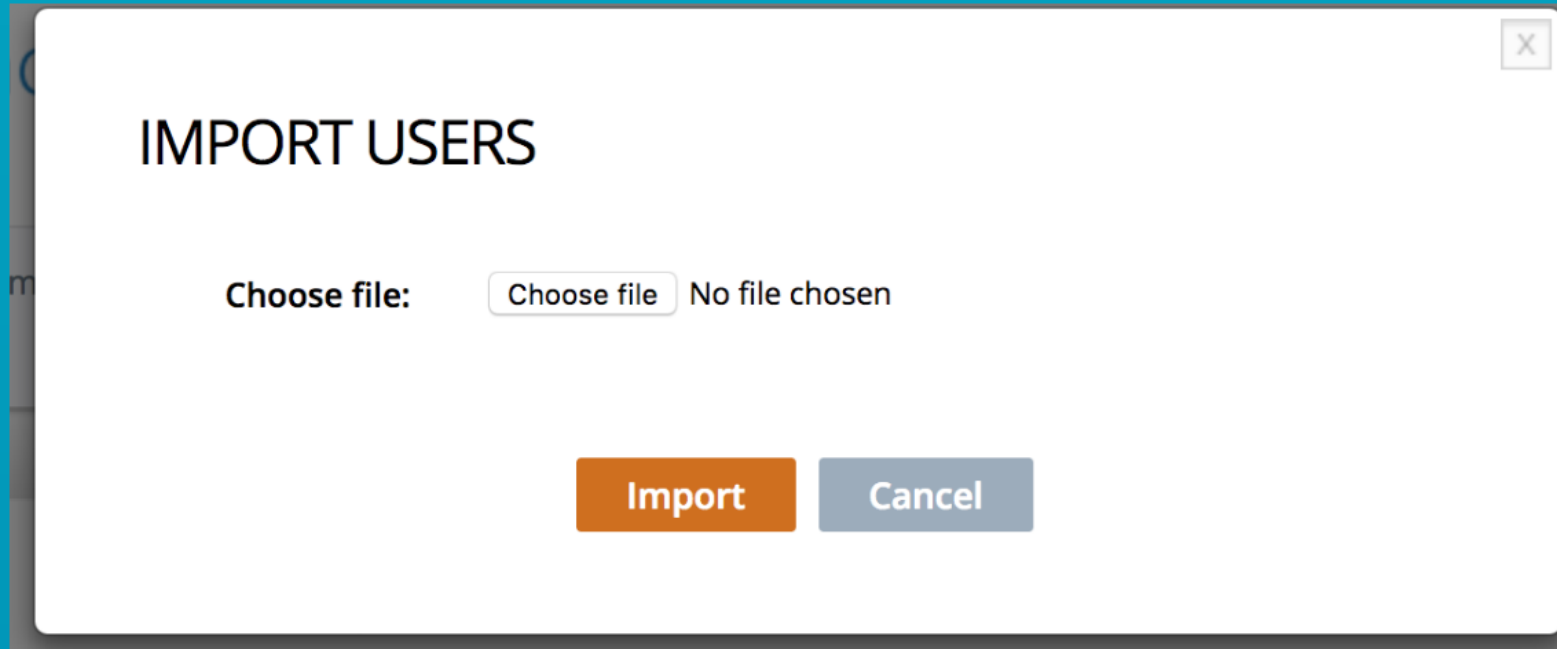
You can invent your own heuristics and apply them to the whole application as well as to requirements analysis.

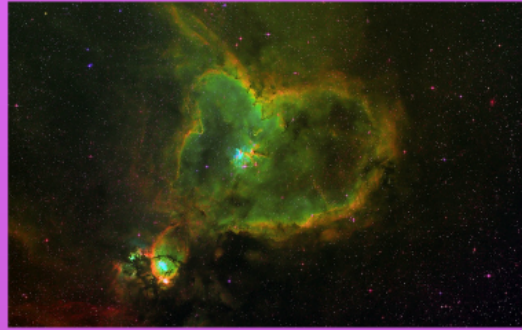




# STEELERS (file upload)

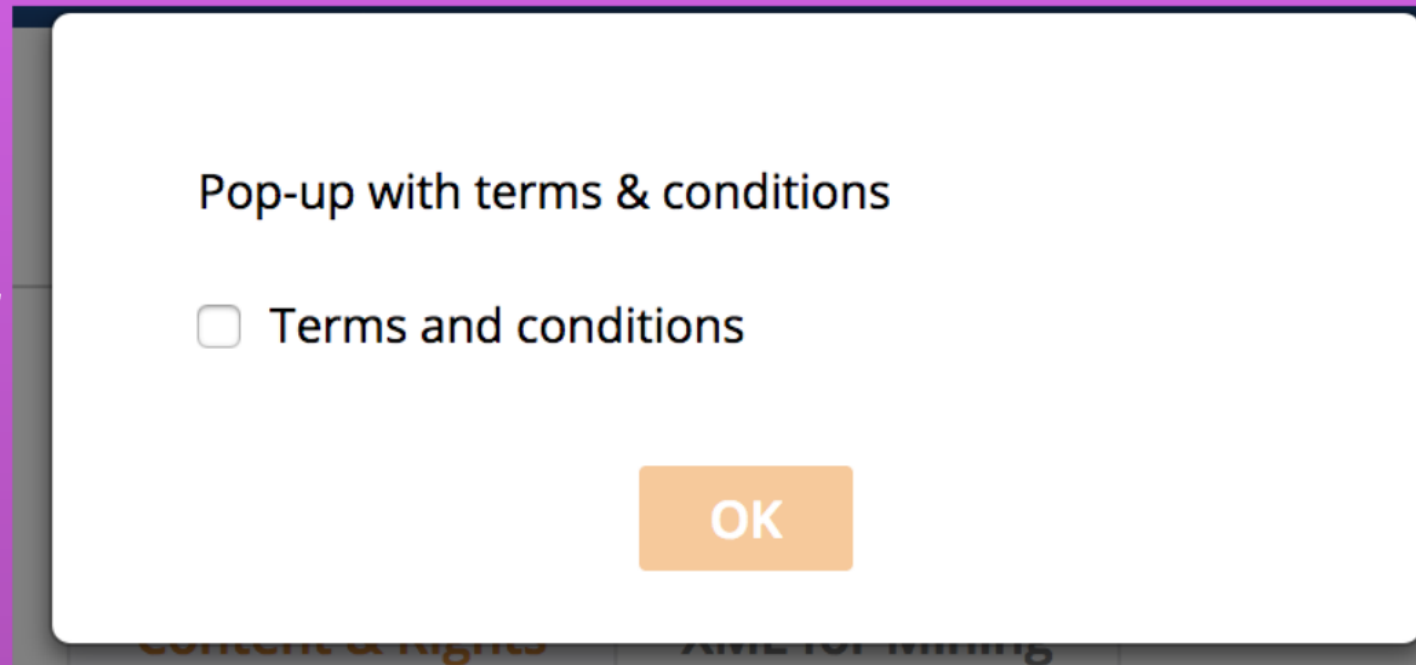
S Several files  
T Template  
E Error messages  
E Existing file  
L Large file  
E EnCoding  
R Rules  
S Success





# SPACES (pop-up)

S Several files  
P Places  
A Authentication  
C Close  
E Enter key  
S Schedule



# To have a success with requirements analysis you will need to:



- Have an analytical mindset

- Not hesitate to ask PO or BA any questions

- Have good domain knowledge in the field



- use heuristics

- Review the tracing up and down

- Pay attention to general wordings

- Be guided by common sense and experience

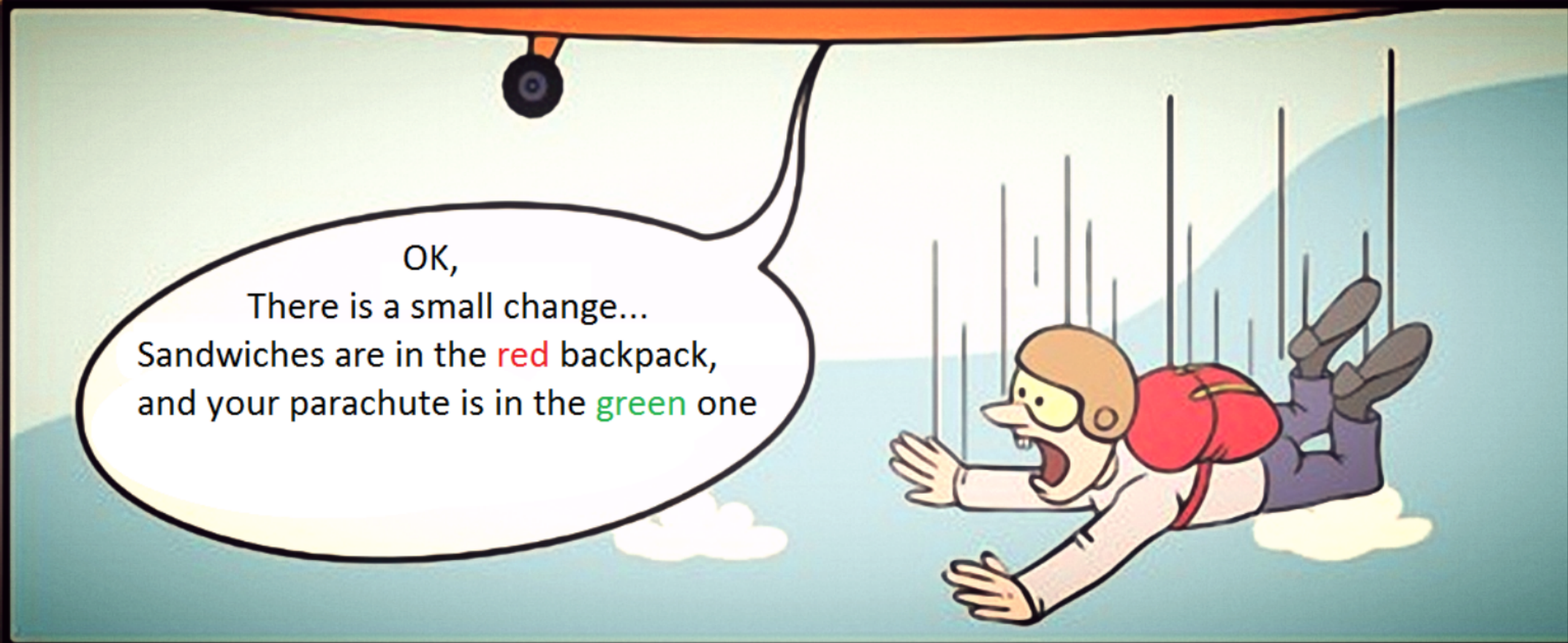
- Put yourself in the Product Owner's, Business Analyst's place and try to imagine whether this requirement is clear to you

To have a success with requirements analysis you will need to:

- Have an analytical mindset
- Not hesitate to ask PO or BA any questions
- Have good domain knowledge in the field
- Use heuristics
- Review the tracing up and down
- Pay attention to general wordings
- Be guided by common sense and experience
- Put yourself in the Product Owner's, Business Analyst's place and try to imagine whether this requirement is clear to you



# It helps to avoid this type of situations



OK,  
There is a small change...  
Sandwiches are in the red backpack,  
and your parachute is in the green one

# Begin with the end in mind

## Evgeny Tkachenko



QA Consultant



[evgeny\\_tkachenko@epam.com](mailto:evgeny_tkachenko@epam.com)



[innovatetesting.blogspot.com](http://innovatetesting.blogspot.com)



[evgenytkachenko](https://www.linkedin.com/in/evgenytkachenko)