Requirements analysis

Zotova Nina

31 April 2016

THE CUSTOMER EXPLAINED

The analyst designed

The programmer made

The customer wanted

Example

REQUIREMENTS” PROBLEMS

- Ошибки, допущенные на стадии сбора требований, составляют от 40 до 60% всех дефектов проекта

(Davis, 1993; Leffingwell, 1997)

- Две наиболее распространенные проблемы, о которых сообщается в Европейском обзоре индустрии ПО - определение требований и управление ими

(ESPITI, 1995)

- Оценка длительности\стоимости, базирующаяся на неполных требованиях, приводит к превышению бюджета на 200-300%

(наблюдения одного из слушателей)

THE SUCCESSFUL PROJECT …

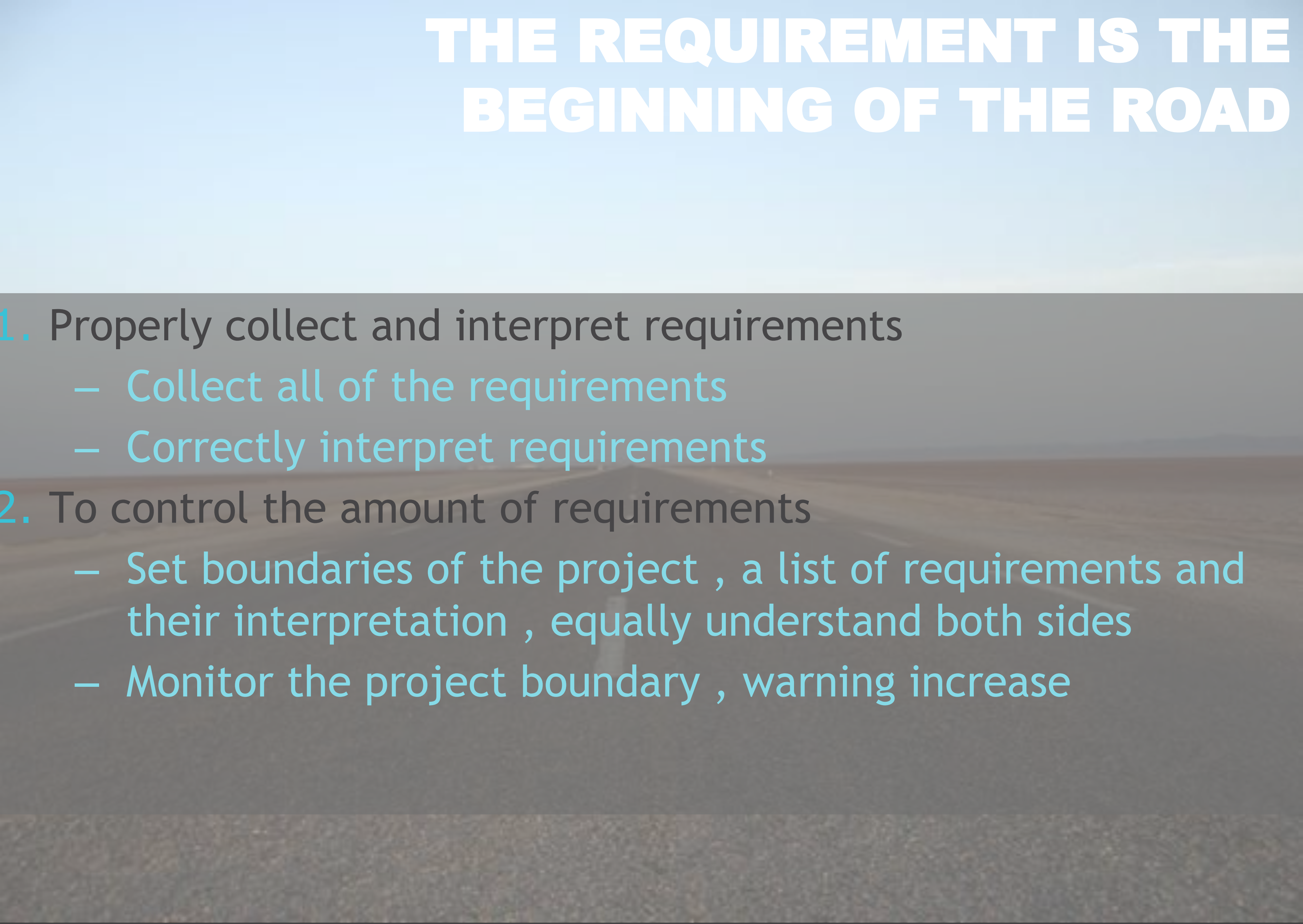
1. Make the final result , which the stakeholders are satisfied with the project
2. To finish it in time (at the time)
3. Remain at the required level of quality to meet the level of resources allocated
4. Were **profitable**

HOW TO ARCHIVE?

To be successful, the project must :

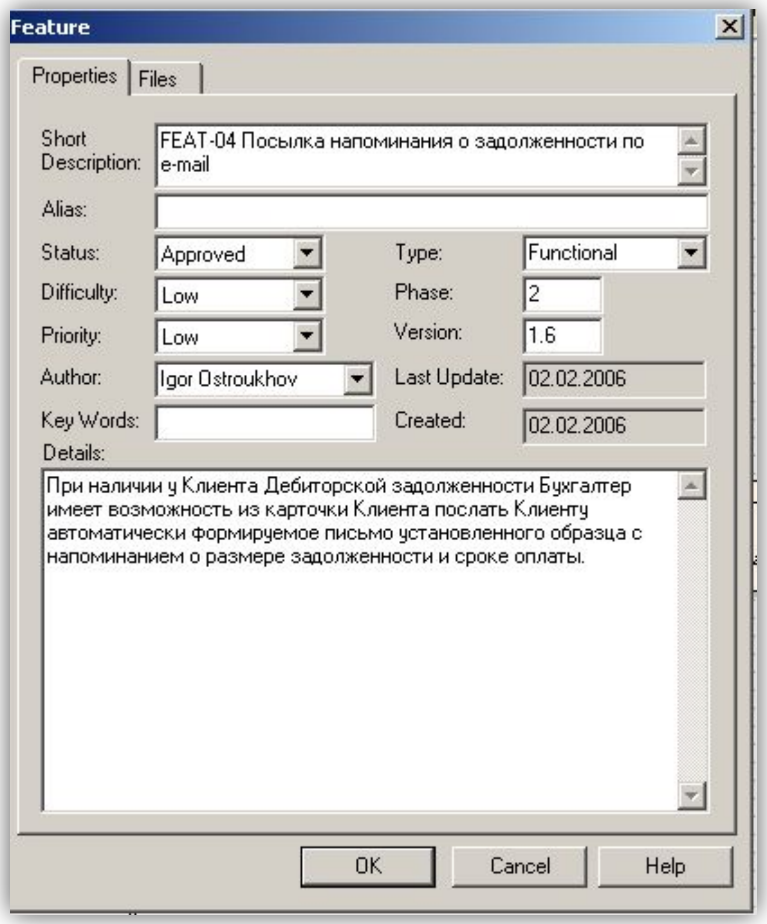
1. Make the final result (deliverable)
    - Properly collect and interpret requirements
    - Correctly implement the requirements
2. End no later than planned
    - Correctly determine the date based on the requirements
    - To organize and carry out work so that date has become a reality
3. Remains in the framework of quality requirements, budget constraints, and the volume of work
    - Consider non-functional requirements and quality attributes
    - Manage requirements

THE REQUIREMENT IS THE BEGINNING OF THE ROAD



1. Properly collect and interpret requirements
   - Collect all of the requirements
   - Correctly interpret requirements
2. To control the amount of requirements
   - Set boundaries of the project , a list of requirements and their interpretation , equally understand both sides
   - Monitor the project boundary , warning increase

THE REQUIREMENT

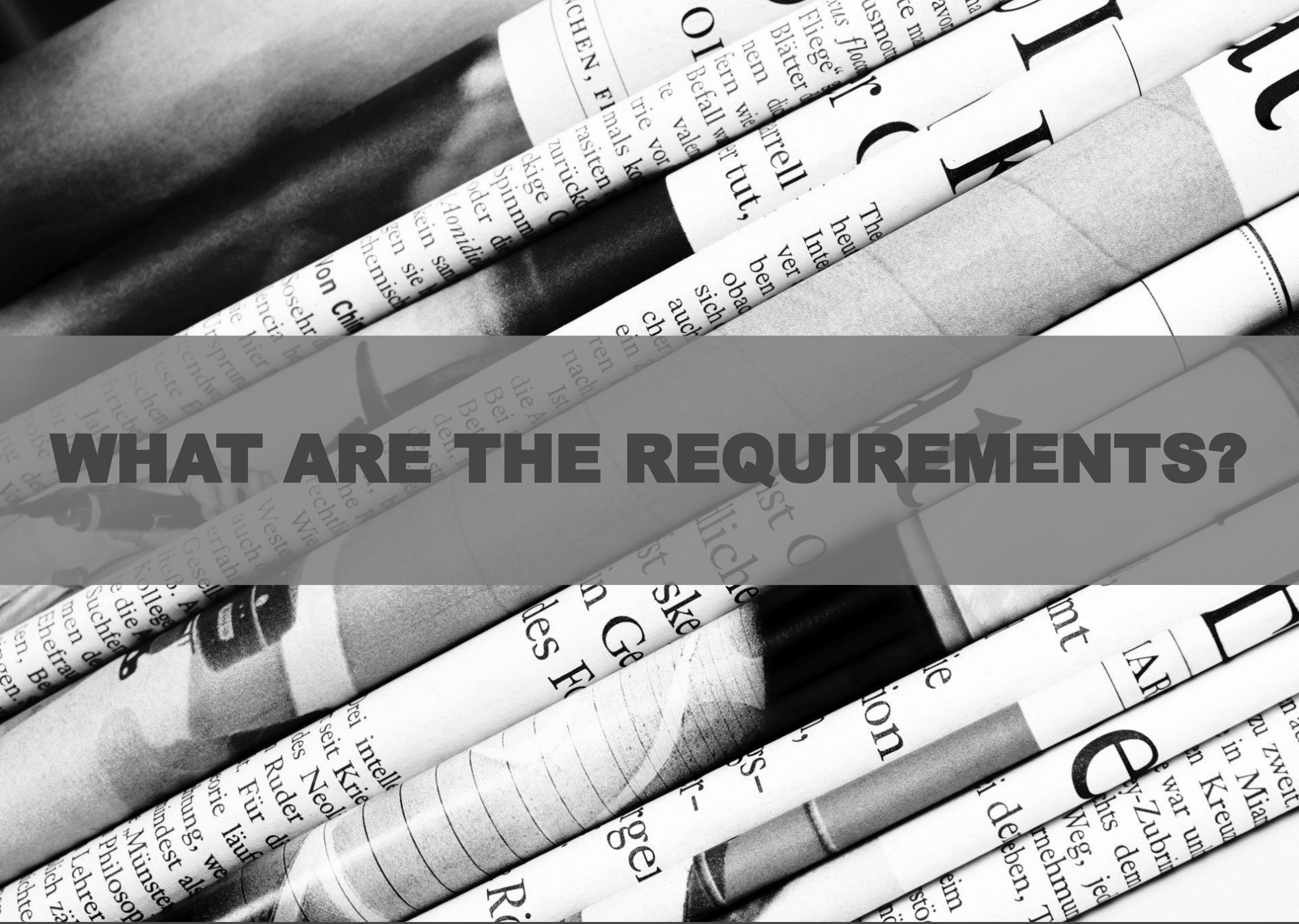


Short Description:	FEAT-04 Посылка напоминания о задолженности по e-mail		
Alias:			
Status:	Approved	Type:	Functional
Difficulty:	Low	Phase:	2
Priority:	Low	Version:	1.6
Author:	Igor Ostroukhov	Last Update:	02.02.2006
Key Words:		Created:	02.02.2006

Details:

При наличии у Клиента Дебиторской задолженности Бухгалтер имеет возможность из карточки Клиента послать Клиенту автоматически формируемое письмо установленного образца с напоминанием о размере задолженности и сроке оплаты.

Requirement - a specification of what is to be implemented. They described the behavior of the system, the system properties, or attributes.



WHAT ARE THE REQUIREMENTS?

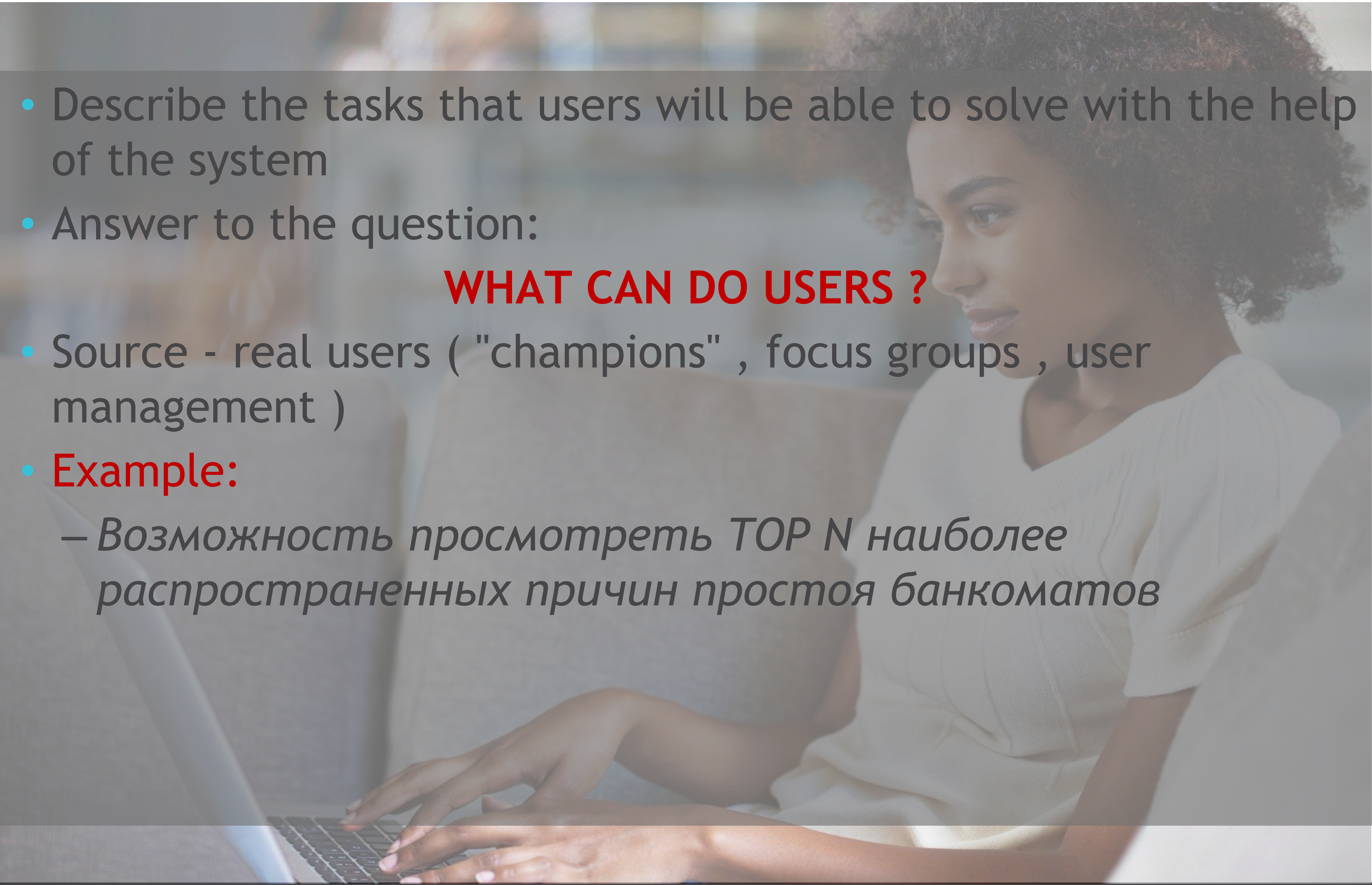
BUSINESS REQUIREMENTS

- They contain high-level objectives of the organization or system customers
- Answer to the question:

  WHY WE DO SYSTEM ?

- Source - the project sponsor (the customer system , marketing , " medium ")
- Example:
  - *Выпустить систему ATM QoS раньше, чем ближайшие конкуренты выпустят свой аналог*

USER REQUIREMENTS

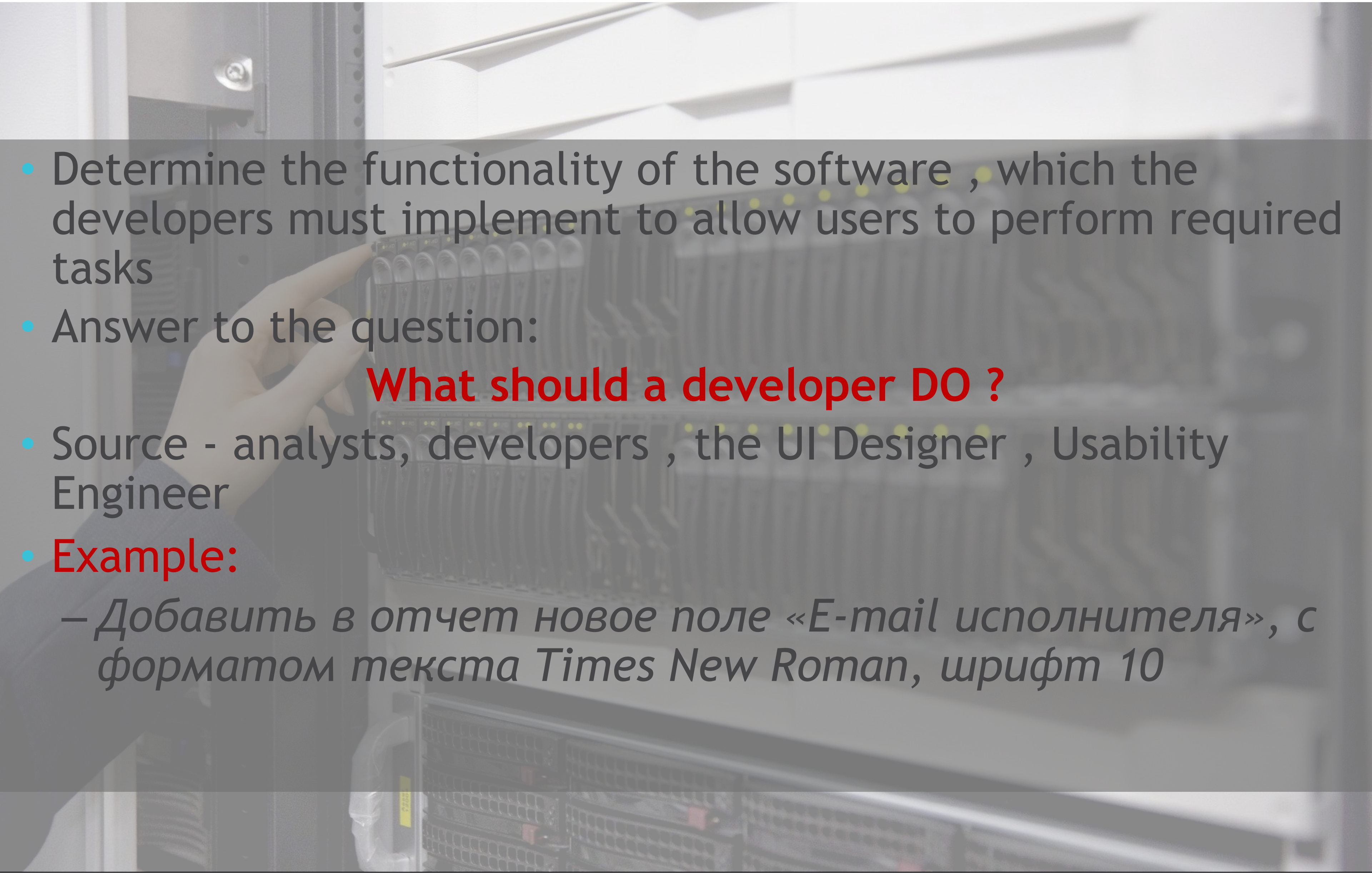


- Describe the tasks that users will be able to solve with the help of the system
- Answer to the question:

WHAT CAN DO USERS ?

- Source - real users ("champions" , focus groups , user management)
- Example:
  - *Возможность просмотреть TOP N наиболее распространенных причин простоя банкоматов*

FUNCTIONAL REQUIREMENTS

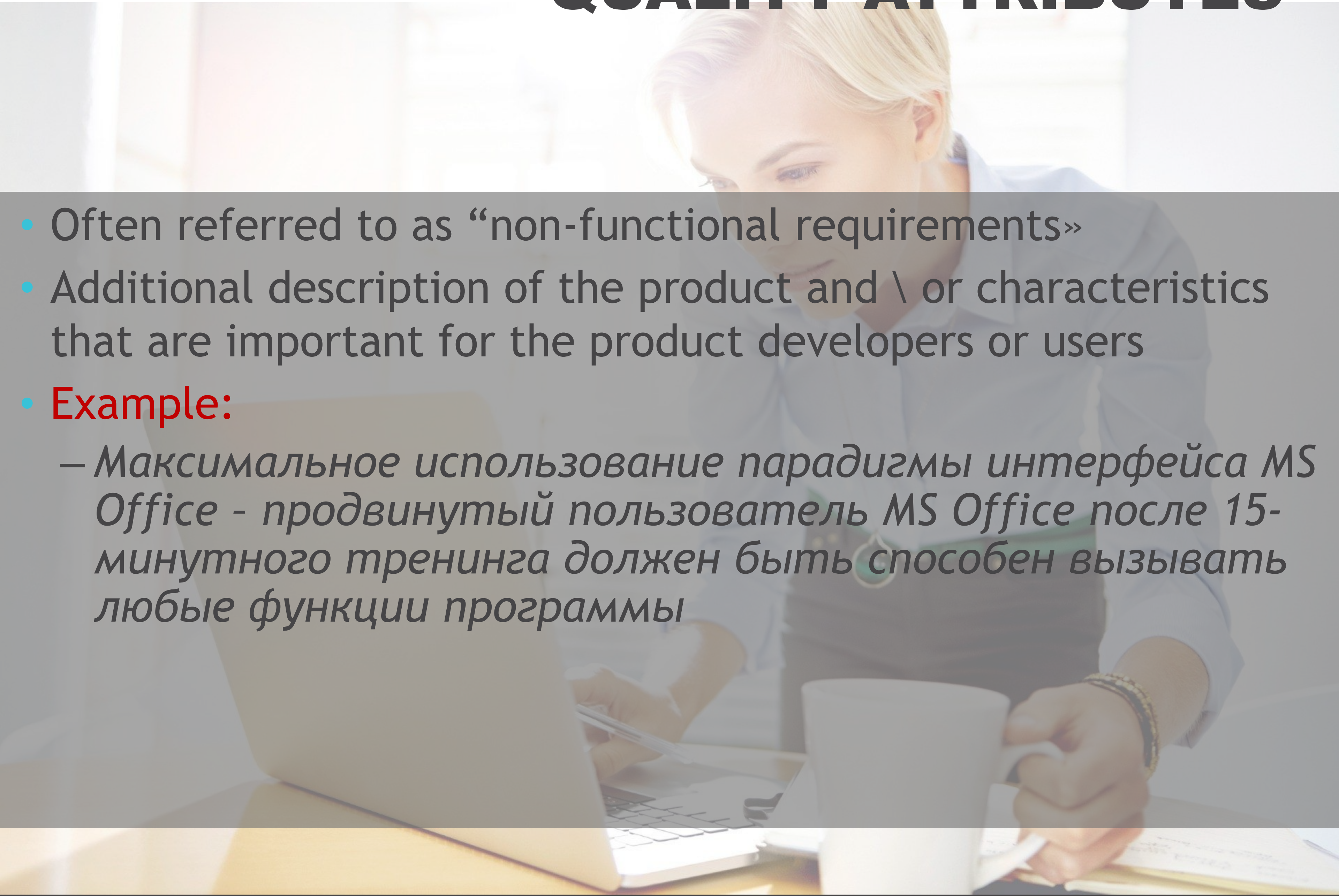


- Determine the functionality of the software , which the developers must implement to allow users to perform required tasks
- Answer to the question:

What should a developer DO ?

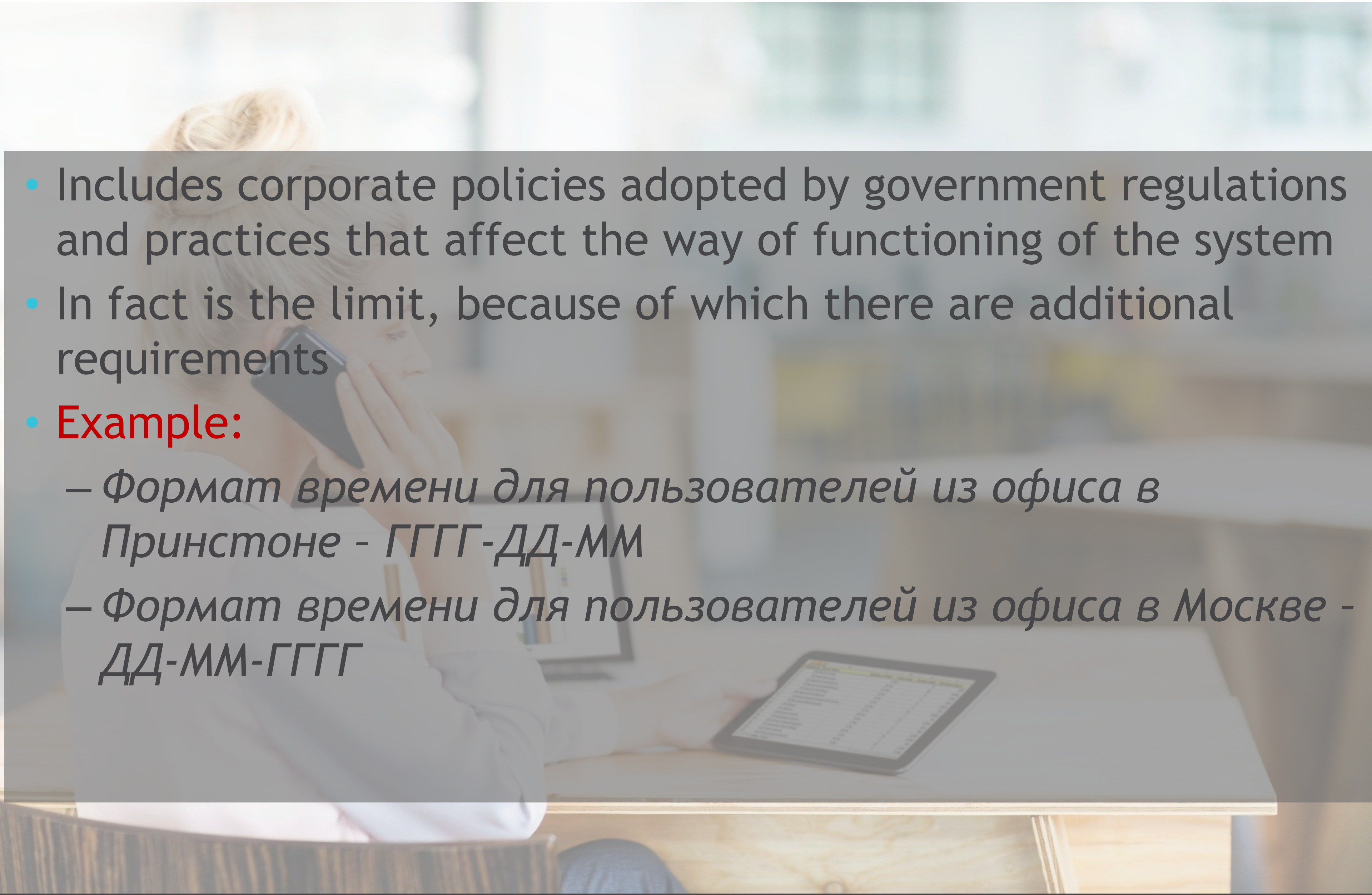
- Source - analysts, developers , the UI Designer , Usability Engineer
- Example:
  - *Добавить в отчет новое поле «E-mail исполнителя», с форматом текста Times New Roman, шрифт 10*

QUALITY ATTRIBUTES



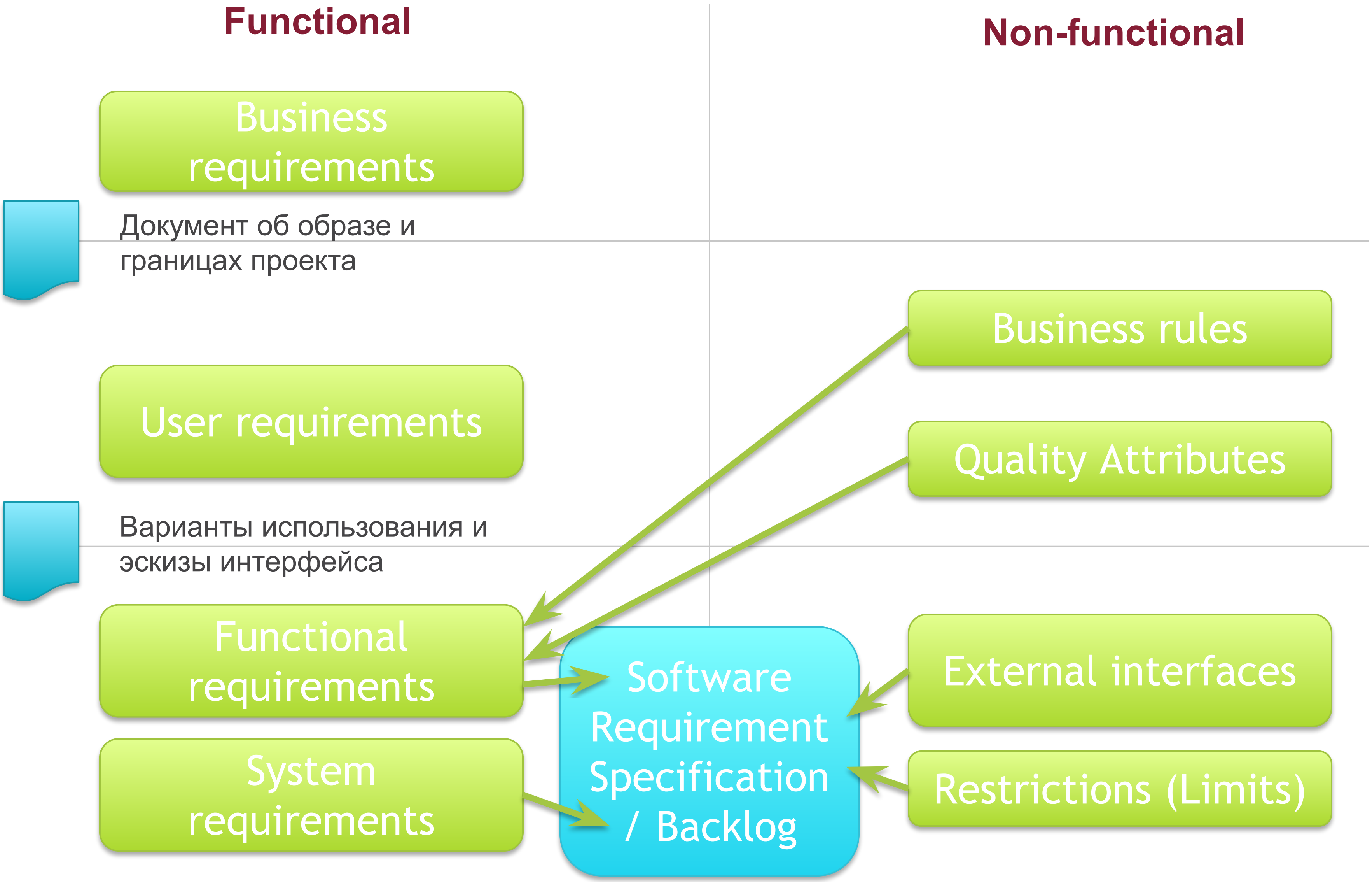
- Often referred to as “non-functional requirements»
- Additional description of the product and \ or characteristics that are important for the product developers or users
- Example:
  - *Максимальное использование парадигмы интерфейса MS Office - продвинутый пользователь MS Office после 15-минутного тренинга должен быть способен вызывать любые функции программы*

BUSINESS RULES



- Includes corporate policies adopted by government regulations and practices that affect the way of functioning of the system
- In fact is the limit, because of which there are additional requirements
- Example:
  - *Формат времени для пользователей из офиса в Принстоне - ГГГГ-ДД-ММ*
  - *Формат времени для пользователей из офиса в Москве - ДД-ММ-ГГГГ*

DRIVING SYSTEMS THINKING



Functional

Non-functional

Business requirements

Документ об образе и границах проекта

User requirements

Варианты использования и эскизы интерфейса

Functional requirements

System requirements

Software Requirement Specification / Backlog

Business rules

Quality Attributes

External interfaces

Restrictions (Limits)

NOT REQUIREMENTS

- Considerations for project management
- Considerations for testing
- Considerations on the application of the system
- Considerations regarding the implementation of the requirements in the code
  - In addition to restrictions

THE TYPES OF REQUIREMENTS AND THEIR RELATIONSHIP

Business requirements

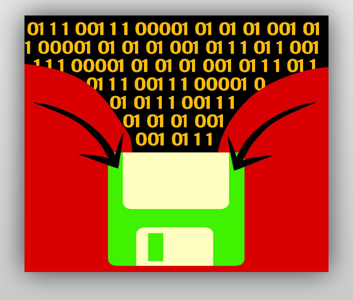
«Повысить прибыль путем предоставлени я новых каналов продаж»

User requirements

«Заказчик сам может сформировать заказ»

System requirements

«Веб-форма размещения заказа»



Restrictions (Limits)

«PlaceOrder.aspx»

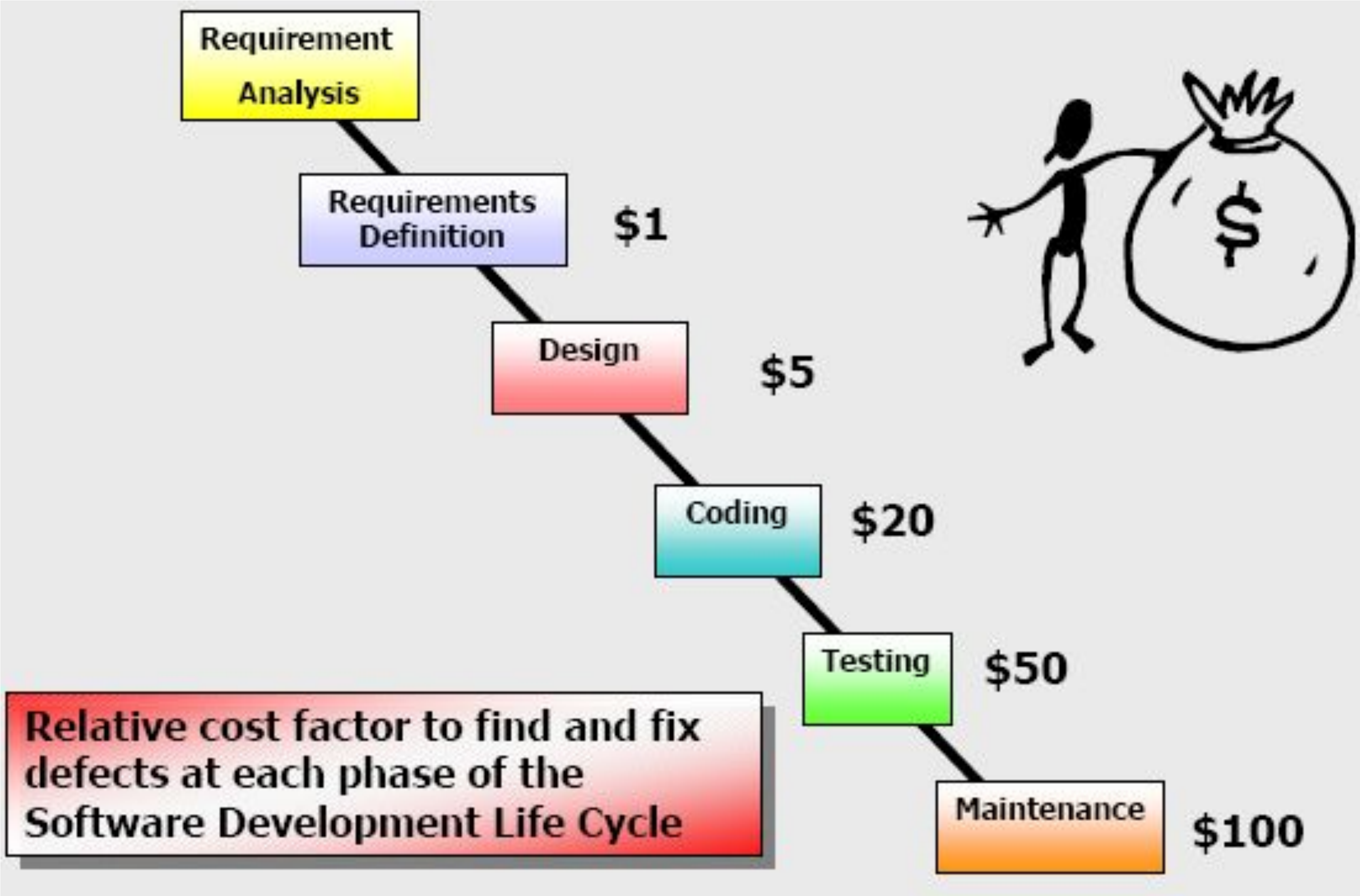
Customer::

cmdConfirm

tbl_Orders

sp_CreateOrder

INSTEAD OF SUMMARY



Requirement Analysis

Requirements Definition $1

Design $5

Coding $20

Testing $50

Maintenance $100

Relative cost factor to find and fix defects at each phase of the Software Development Life Cycle

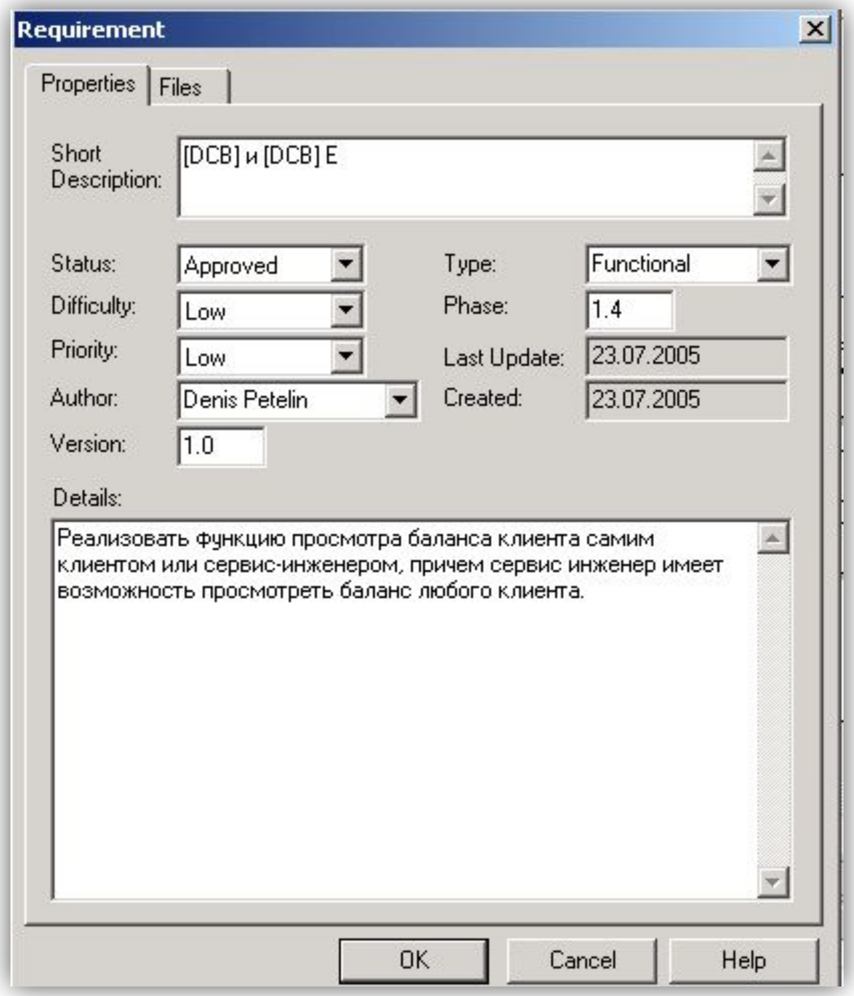
FEATURES SUPERIOR REQUIREMENTS

- Complete
- Correct
- Ambiguous
- Consistent
- Verifiable
- Prioritization

COMPLETE

- Each request must contain all the necessary information to the developer to ensure that design and implement the required functionality correctly
- Ideally, the requirement describes :
  - What need to do?
  - How does it look like?
  - How does it behave?

WHAT NEED TO DO



Short Description: [DCB] и [DCB] E

Status: Approved

Difficulty: Low

Priority: Low

Author: Denis Petelin

Version: 1.0

Type: Functional

Phase: 1.4

Last Update: 23.07.2005

Created: 23.07.2005

Details:

Реализовать функцию просмотра баланса клиента самим клиентом или сервис-инженером, причем сервис инженер имеет возможность просмотреть баланс любого клиента.

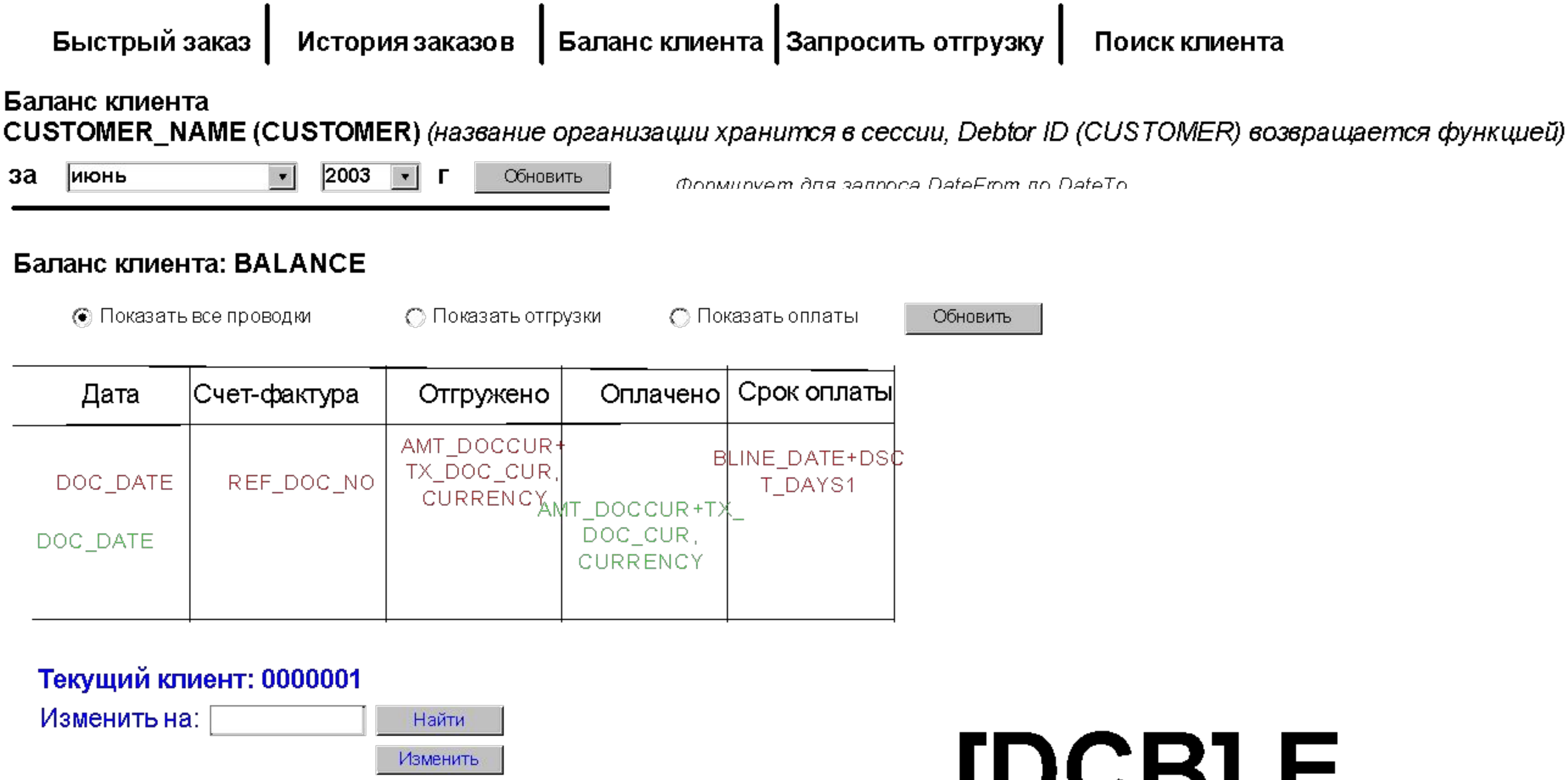
FR-WUI-005

Просмотр баланса

[DCB] и [DCB] E

«Реализовать функцию просмотра баланса Заказчика самим Заказчиком или сервис-инженером, причем сервис инженер имеет возможность просмотреть баланс любого Заказчика»

HOW DOES IT LOOK LIKE



Быстрый заказ | История заказов | Баланс клиента | Запросить отгрузку | Поиск клиента

Баланс клиента

CUSTOMER_NAME (CUSTOMER) *(название организации хранится в сессии, Debtor ID (CUSTOMER) возвращается функцией)*

за июнь ▾ 2003 ▾ г Обновить

Формирует для запроса DateFrom по DateTo

Баланс клиента: BALANCE

● Показать все проводки ○ Показать отгрузки ○ Показать оплаты Обновить

Дата	Счет-фактура	Отгружено	Оплачено	Срок оплаты
DOC_DATE	REF_DOC_NO	AMT_DOCCUR+TX_DOC_CUR, CURRENCY		BLINE_DATE+DSCT_DAYS1
DOC_DATE			AMT_DOCCUR+TX_DOC_CUR, CURRENCY	

Текущий клиент: 0000001

Изменить на: ______ Найти

Изменить

[DCB] E

Source: FR-WUI-005

HOW DOES IT BEHAVE

- Primary Actor: Инженер (пользователь)
- Precondition: Инженер (пользователь) прошел авторизацию в системе, получил доступ к базовой странице [NAV] и выбрал пункт.
- Success guarantee: На экране отображена страница с соответствующей информацией.
  - Main success scenario:

1. Система отображает страницу **Баланс Заказчика** [DCB]. По умолчанию в полях выбора даты указывается текущий месяц и год, а на самой странице отображается информация о *всех проводках* Заказчика за текущий месяц.
2. Инженер (пользователь) изменяет значения, заданные в полях выбора даты, и нажимает кнопку **Обновить**. Система выводит информацию о проводках Заказчика за указанный месяц.
   Или:
3. Инженер (пользователь) задает, какую именно информацию он хочет увидеть - установив переключатель Показать все проводки, Показать отгрузки, Показать оплаты - и нажимает кнопку **Обновить**. Система выводит информацию требуемого типа за указанный месяц.

- Extensions:
- Во всех случаях, когда система обращается к коннектору, а он оказывается недоступным, отображается страница Баланс Заказчика с информацией об ошибке **Ошибка доступа к системе**.
- Инженер может изменить текущего Заказчика, введя его номер в соответствующее поле и нажав кнопку **Изменить**. Для поиска номера Заказчика можно воспользоваться кнопкой **Найти** [DCS] Customer Search.

COMPLETE: TYPICAL PROBLEMS

Complet
e

All significant requirements are included.
No items have been left
for future definition.

Often:

- Non Functional requirements missing
- Hidden assumptions
- Too general statements

Examples:
"Application V.2 should be 50% faster"
Each operation? Maximum time? What is current performance?

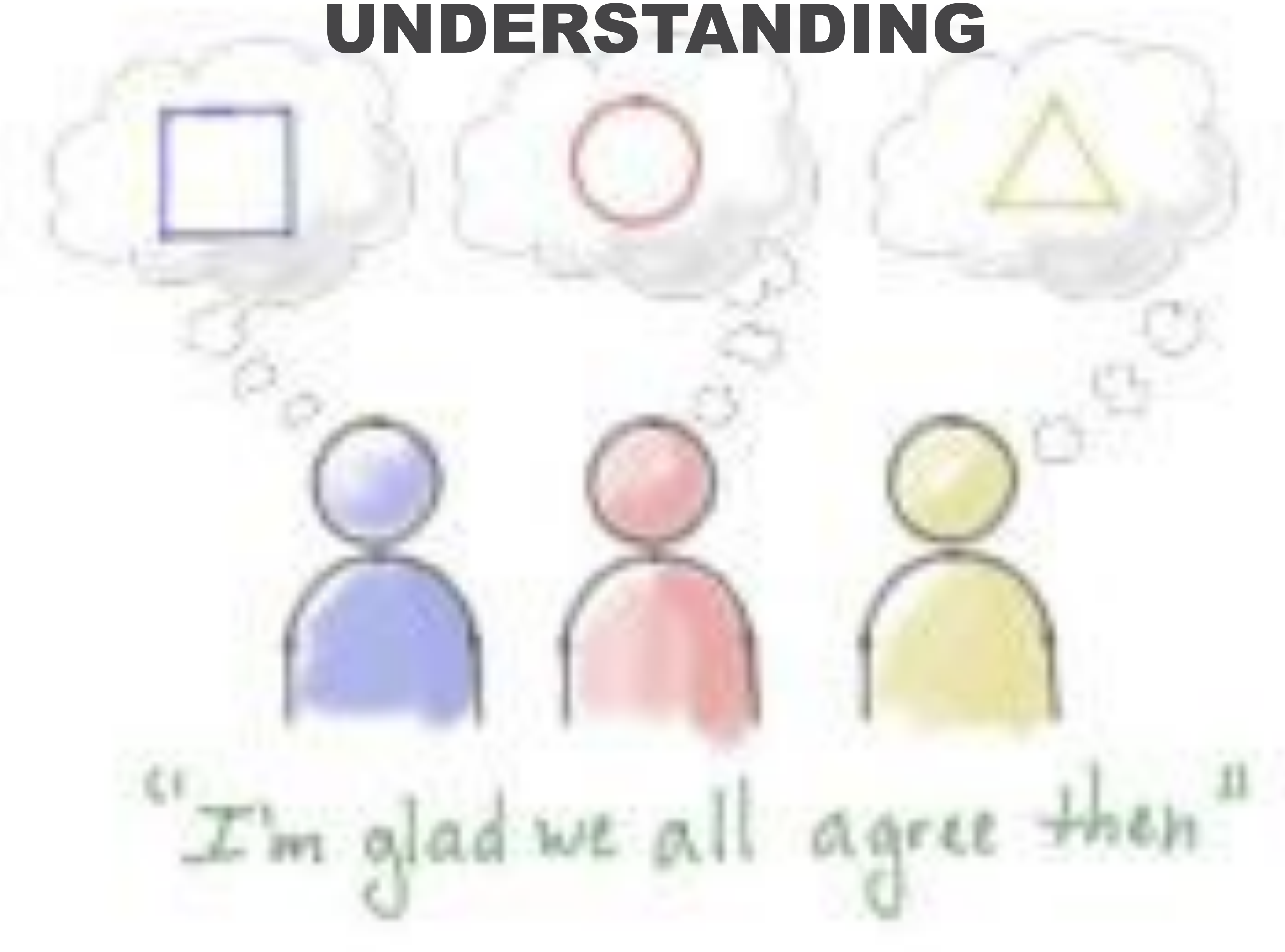
"Application will be localization ready'"
Customer assumes: we will just provide file with resources

CORRECT

Each requirement must accurately describe the desired functionality

- **Aspect One: Interpretation**
  - Make sure that the user understanding and the same developer
- **Aspect Two: Consistency**
  - There should be no conflict , duplicate or conflicting requirements

WHEN WE VERBALLY DISCUSS IDEAS, WE MAY INCORRECTLY BELIEVE WE HAVE THE SAME UNDERSTANDING



REPRESENTING OUR IDEAS ALLOWS US TO DETECT INCONSISTENCIES

CORRECT: TYPICAL PROBLEMS

Corre
ct

Every stated requirement represents
Something required of the system
to be built.

Usual problems:

- Incorrect statements (Copy-Paste, Mistypes)
- Obsolete requirements
- Gold plating – features that add cost, but not much value
- Technically impossible features
- Design mixed into requirements

AMBIGUOUS

- The ability to interpret the same requirement of different
- Several readers have several different understandings of what needs to be done to implement this requirement

AMBIGUOUS : TYPICAL PROBLEMS

Ambiguou
s

Every statement has one interpretation.
Terms are clear and well defined.

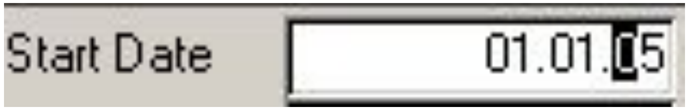
Example:

“The feature XYZ is optional”

Designer: “*The feature is optional, we do not have to implement this*”.
Customer: “*Wow, the product will have nice XYZ feature*”.
Marketing: “*The feature is optional, so we’ll provide it as additional package for extra money*”.

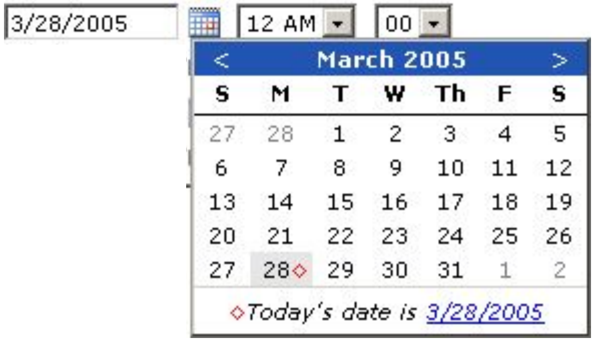
EXAMPLE

Data Input



Start Date 01.01.05

1-2h



3/28/2005 12 AM 00

< March 2005 >

S	M	T	W	Th	F	S
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28◇	29	30	31	1	2

◇*Today's date is 3/28/2005*

3-5h

WORDS THAT SHOULD RAISE SUSPICION:

- Приемлемый, адекватный
- Гибкий
- Улучшенный, более
- Необязательно
- Несколько
- Элегантный, прозрачный
- Удобный
- Быстрый, моментальный
- Эффективный
- Устойчивый к сбоям

WORDS THAT SHOULD RAISE SUSPICION:

easy
provide for
as a minimum
be capable of
effective
timely
as applicable
if possible
TBD
as appropriate
if practical
at a minimum
but not limited to
capability of
efficient

capability to
normal
minimize
maximize
optimize
rapid
user-friendly
simple
often
usual
large
flexible
robust
state-of-the-art
improved

CONSISTENT

Consiste nt

Conflicting terminology, contradictory required
Actions and impossible combinations
are notably absent

Example:
“For Customers, who are exempted from receiving billing reminder notices, ensure that two notices are generated”.

It is inconsistent: you don’t generate reminders for exempted customers.

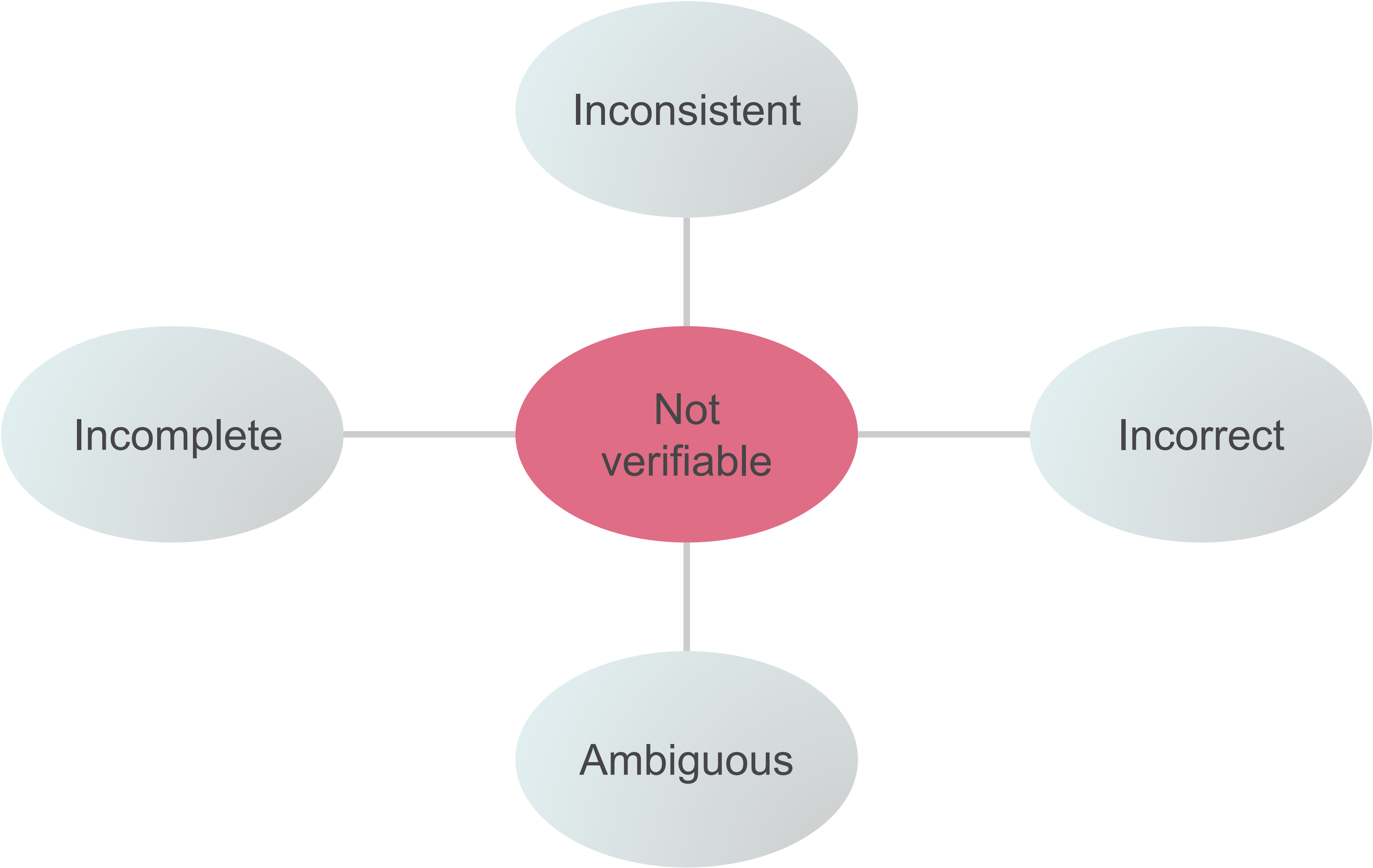
VERIFIABLE

Examples:

“*Application will have 0 bugs.*”

“*Application will be user friendly.*”

REQUIREMENTS: TYPICAL PROBLEMS

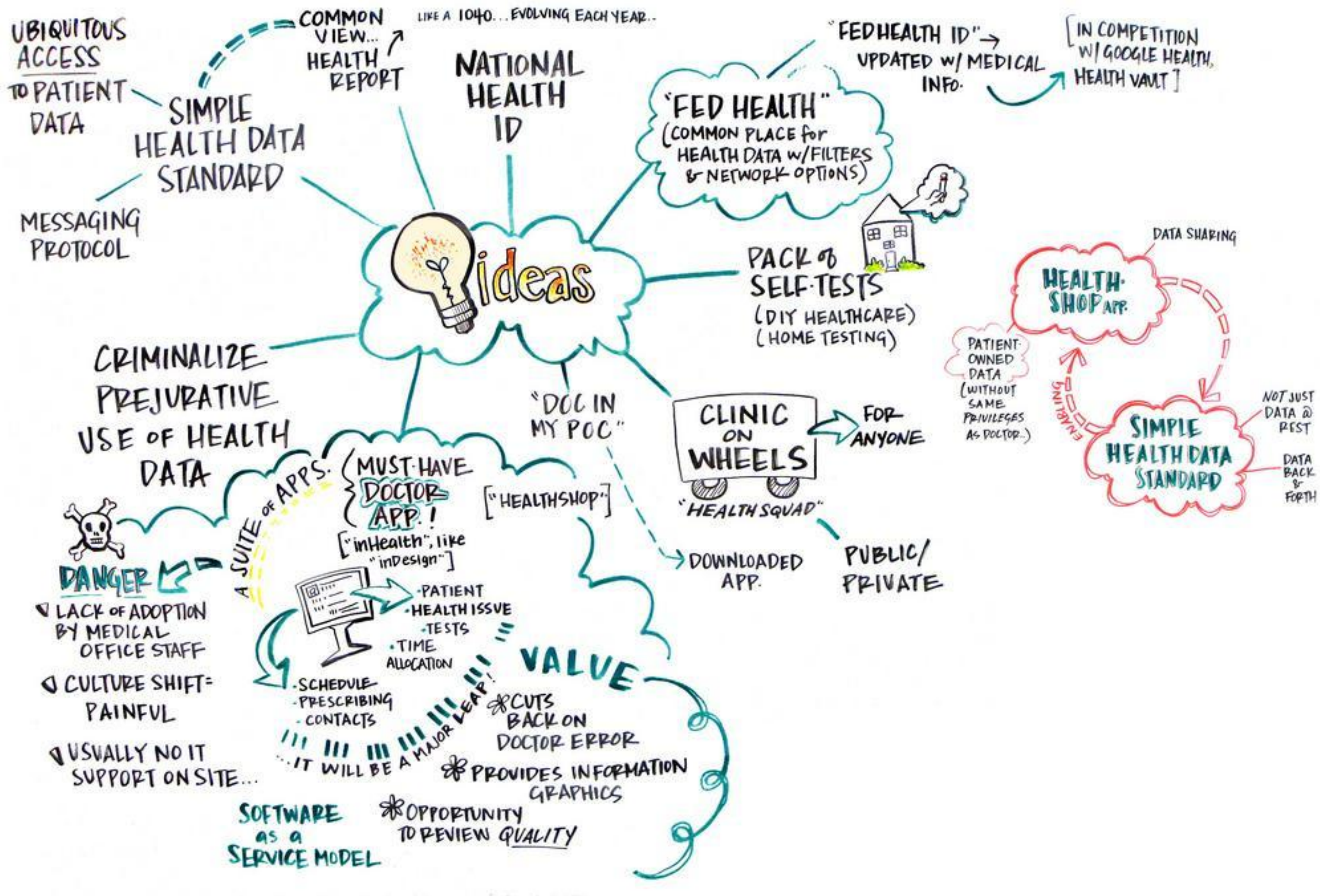


PRIORITIZATION

- **Not on the principle of separation " is important , it does not matter ," and ordering on the principle of "first - second-third ”**
  - Each request is mapped version of the application (iteration of development) , in which it should appear
  - The sponsor and the users should be aware that the priorities only describe what is being done first , and then what , not what will be done and what is not

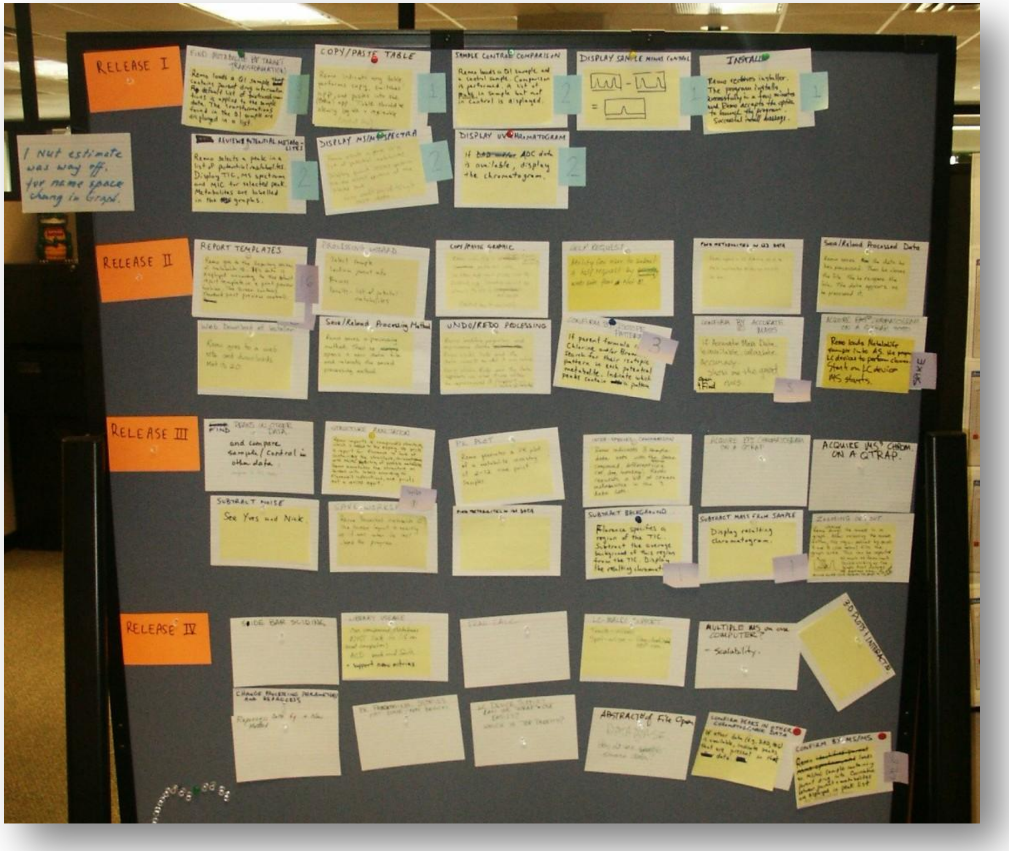
WHAT IS REQUIREMENTS IN AGILE?

PRODUCT BACKLOG

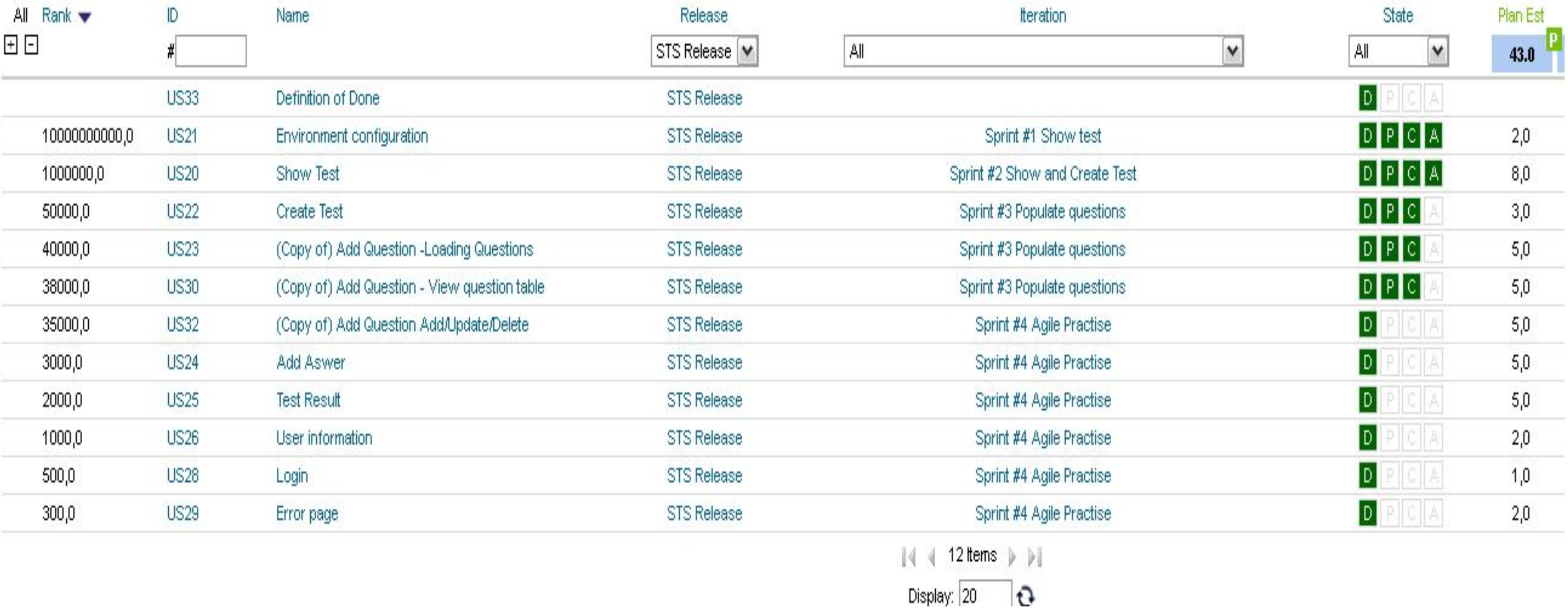

UBIQUITOUS ACCESS TO PATIENT DATA
SIMPLE HEALTH DATA STANDARD
MESSAGING PROTOCOL
COMMON VIEW... HEALTH REPORT
LIKE A 1040... EVOLVING EACH YEAR..
NATIONAL HEALTH ID
"FED HEALTH" (COMMON PLACE for HEALTH DATA w/FILTERS & NETWORK OPTIONS)
"FEDHEALTH ID" → UPDATED w/ MEDICAL INFO.
[IN COMPETITION W/ GOOGLE HEALTH, HEALTH VAULT]
ideas
PACK of SELF-TESTS (DIY HEALTHCARE) (HOME TESTING)
CRIMINALIZE PREJURATIVE USE OF HEALTH DATA
"DOC IN MY POC"
CLINIC ON WHEELS
FOR ANYONE
"HEALTH SQUAD"
DOWNLOADED APP.
PUBLIC/PRIVATE
A SUITE of APPS:
MUST-HAVE DOCTOR APP!
["inHealth", like "inDesign"]
["HEALTHSHOP"]
DANGER
LACK of ADOPTION BY MEDICAL OFFICE STAFF
CULTURE SHIFT= PAINFUL
USUALLY NO IT SUPPORT ON SITE...
-PATIENT
-HEALTH ISSUE
-TESTS
-TIME ALLOCATION
-SCHEDULE
-PRESCRIBING
-CONTACTS
...IT WILL BE A MAJOR LEAP!
VALUE
CUTS BACK ON DOCTOR ERROR
PROVIDES INFORMATION GRAPHICS
OPPORTUNITY TO REVIEW QUALITY
SOFTWARE as a SERVICE MODEL
DATA SHARING
HEALTH-SHOP APP.
PATIENT-OWNED DATA (WITHOUT SAME PRIVILEGES AS DOCTOR..)
ENABLING
SIMPLE HEALTH DATA STANDARD
NOT JUST DATA @ REST
DATA BACK & FORTH

PRODUCT BACKLOG

- Master list of all "features"
- High priority features are split into "stories" achievable within an iteration.
- Each "story" is prioritized and scoped.

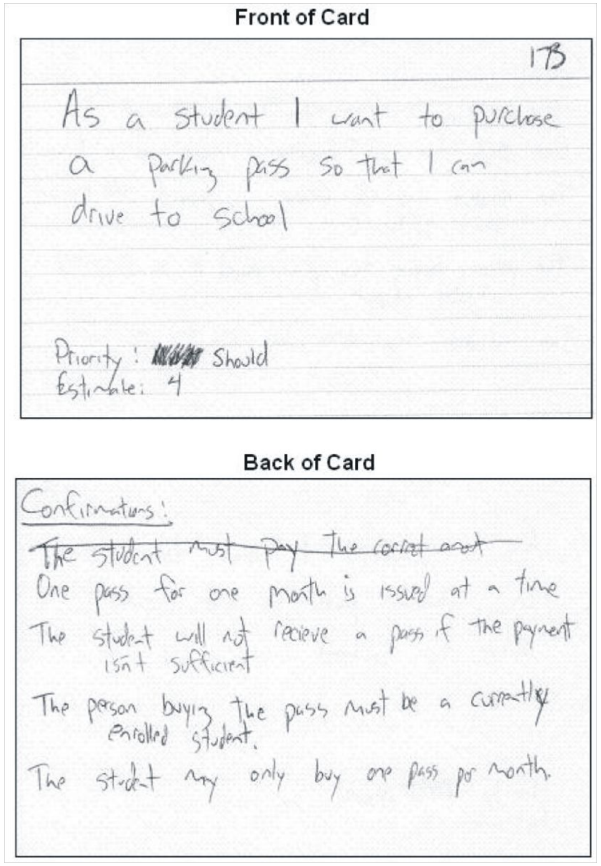


PRODUCT BACKLOG



All	Rank ▼	ID	Name	Release	Iteration	State	Plan Est
⊞ ⊟		#		STS Release ▾	All ▾	All ▾	43.0 P
		US33	Definition of Done	STS Release		**D** P C A	
	10000000000,0	US21	Environment configuration	STS Release	Sprint #1 Show test	**D** **P** **C** **A**	2,0
	1000000,0	US20	Show Test	STS Release	Sprint #2 Show and Create Test	**D** **P** **C** **A**	8,0
	50000,0	US22	Create Test	STS Release	Sprint #3 Populate questions	**D** **P** **C** A	3,0
	40000,0	US23	(Copy of) Add Question -Loading Questions	STS Release	Sprint #3 Populate questions	**D** **P** **C** A	5,0
	38000,0	US30	(Copy of) Add Question - View question table	STS Release	Sprint #3 Populate questions	**D** **P** **C** A	5,0
	35000,0	US32	(Copy of) Add Question Add/Update/Delete	STS Release	Sprint #4 Agile Practise	**D** P C A	5,0
	3000,0	US24	Add Aswer	STS Release	Sprint #4 Agile Practise	**D** P C A	5,0
	2000,0	US25	Test Result	STS Release	Sprint #4 Agile Practise	**D** P C A	5,0
	1000,0	US26	User information	STS Release	Sprint #4 Agile Practise	**D** P C A	2,0
	500,0	US28	Login	STS Release	Sprint #4 Agile Practise	**D** P C A	1,0
	300,0	US29	Error page	STS Release	Sprint #4 Agile Practise	**D** P C A	2,0

USER STORY



Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: Should
Estimate: 4

Back of Card

Confirmations:

~~The student must pay The correct amt~~

One pass for one month is issued at a time

The student will not recieve a pass if the payment isn't sufficient

The person buying the pass must be a currently enrolled student.

The student may only buy one pass per month.

Independent

Negotiable

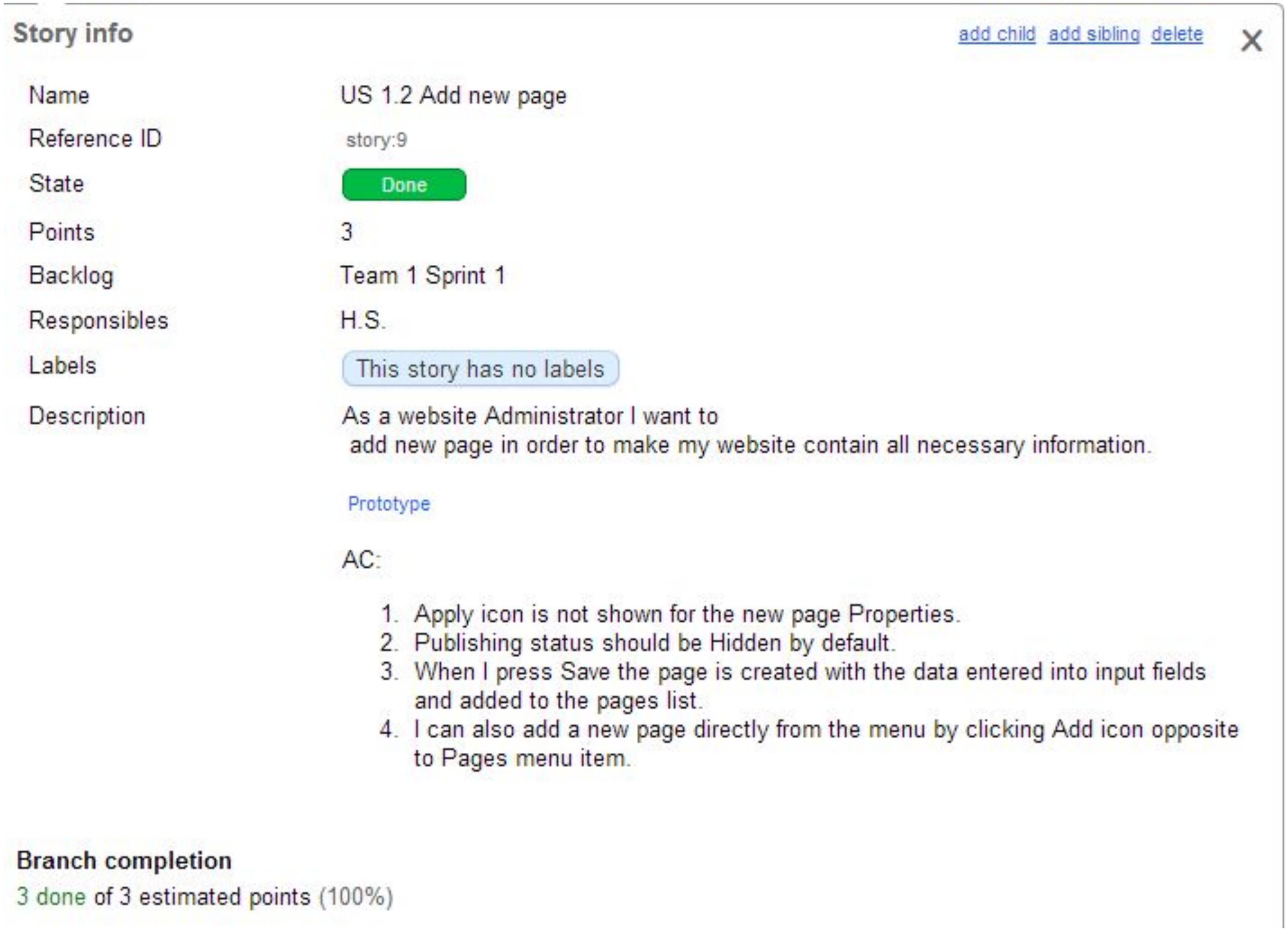
Valuable

Estimable

Small

Testable

USER STORY



Story info

Name	US 1.2 Add new page
Reference ID	story:9
State	Done
Points	3
Backlog	Team 1 Sprint 1
Responsibles	H.S.
Labels	This story has no labels
Description	As a website Administrator I want to add new page in order to make my website contain all necessary information.

Prototype

AC:

1. Apply icon is not shown for the new page Properties.
2. Publishing status should be Hidden by default.
3. When I press Save the page is created with the data entered into input fields and added to the pages list.
4. I can also add a new page directly from the menu by clicking Add icon opposite to Pages menu item.

Branch completion

3 done of 3 estimated points (100%)

CLARIFYING REQUIREMENTS – OPTION 1: GROOMING

When	Before a sprint, even 1 week before
Input	User Stories created and described by PO
Who participates	All team, dev and test, PO
What testers do before grooming	Analyze user stories from the point of: • What information is missing and will prevent us from testing? • What is strange /not logical/contradicts other reqs? • Do I know how to test this? • What is missing to test this? (e.g. data)
On grooming	Ask questions by user stories Clarify PO answers Plan if needed additional discussions (e.g. with architecture)

CLARIFYING REQUIREMENTS OPTION 2 - 3 AMIGOS SESSIONS

3 amigos are

- Business Analysis or Product Owner -What problem are we trying to solve?
- Developer(s) -How might we build a solution to solve that problem?
- Tester(s) -What about this, what could possibly happen?

3 amigo sessions are conducted to discuss user stories, questions by user stories

WHAT CAN GO WRONG?

What can happen?

- Not enough stories exist/have details in the backlog before grooming
- PO does not know what the purpose of the story or details
- PO is not ready to answer on questions

- PO does not come on grooming

- Team works with several POs and they contradict each other
- PO changes opinion a bit later
- Requirements are changed on the fly

What can we do?

- Plan grooming in advance
- Prepare and share questions with PO before session
- Push to “move out of sprint” not clear stories on plann

- Plan grooming in advance
- Prepare and share questions with PO before session

- Store PO’s answers in common source
- Have answers recorded; measure and communicated impact of changes

ACCEPTANCE CRITERIA

WHAT ARE ACCEPTANCE CRITERIA?

- **Acceptance Criteria** are the conditions that a software product must satisfy to be accepted by a user, customer, or in the case of system level functionality, the consuming system

- **Acceptance Criteria** are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic, Feature, and Story Level. Acceptance criteria constitute our “Definition of Done”, and by done I mean well done.

ACCEPTANCE CRITERIA FORMAT

The Given/When/Then format is helpful way to specify criteria:

Given *some precondition*
When *I do some action*
Then *I expect some result*

HOW CAN ACCEPTANCE CRITERIA BE USED FOR TESTING

All acceptance criteria must be checked during testing

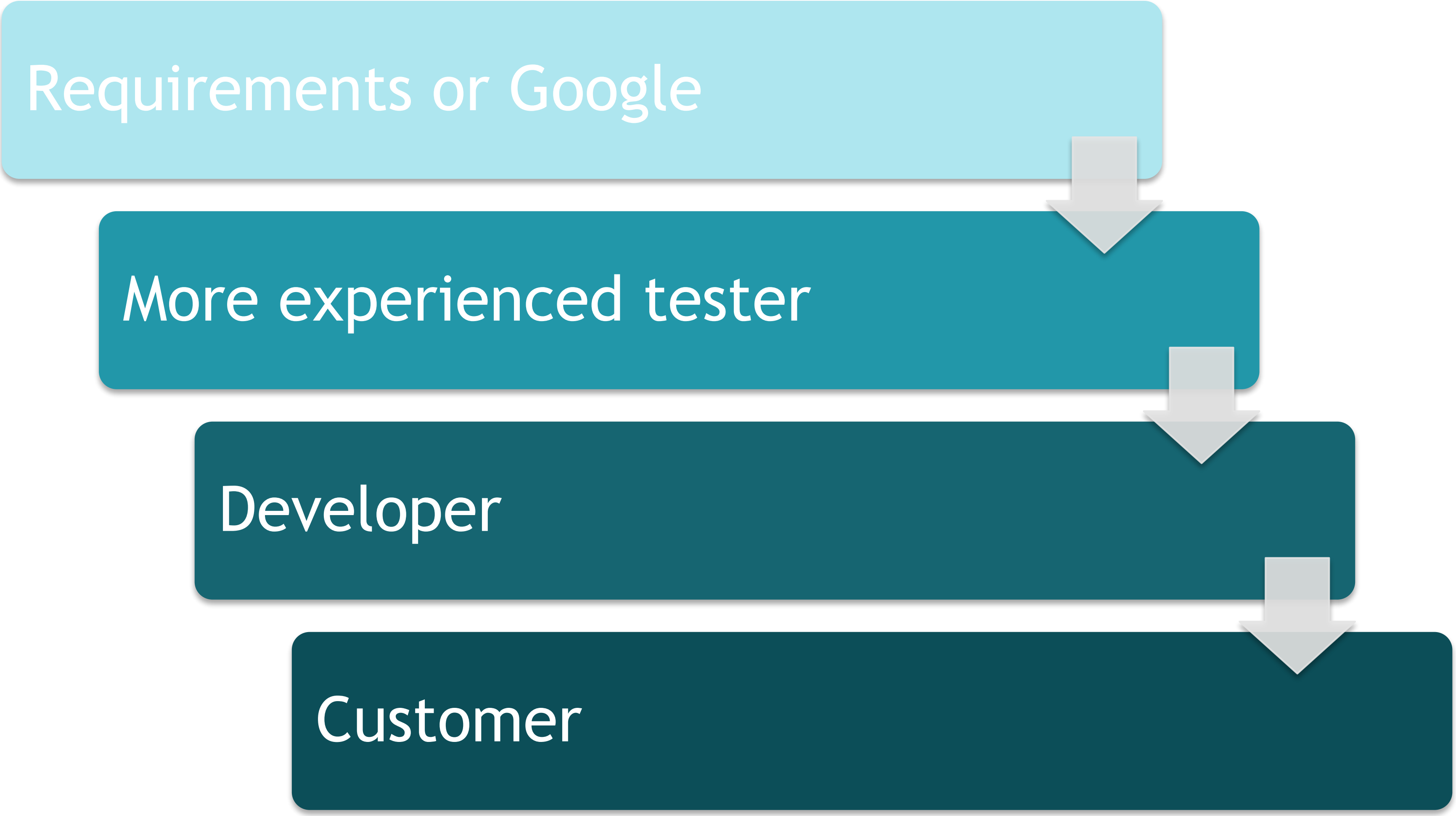
Besides them tester also should think about

- All possible negative verifications
- Functional verifications for controls
- UI check
- Integration this functionality with the system
- End-to-end scenario
- …

Acceptance criteria is a start for testing, but not all what can we should verify!

ART ASK QUESTIONS

QUESTION: WAY TO ANSWER



Requirements or Google

More experienced tester

Developer

Customer

WHEN THE RESPONSE IS SUFFICIENT ?

- Specific answer
- No “may be”, “I think”
- No contradictions
- You are 100% sure
- You understand “why”

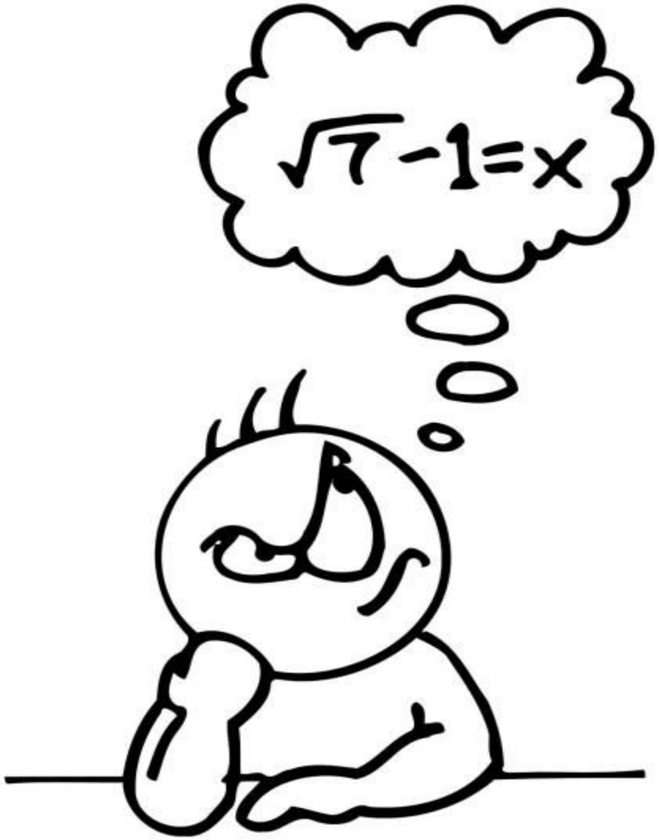
NO ANSWER?

- Remind by email
- No answer in a week- escalate his manager

ASKING QUESTIONS REQUIRES COURAGE

Many people do not ask questions because they are afraid

ASKING RIGHT QUESTIONS REQUIRES THINKING AND SKILL



There is a lot of mental work between “I do not understand” and the **right question**

TO ASK A QUESTION, YOU NEED TO NOTICE THE PROBLEM FIRST

SHOULD WE CLARIFY FIELDS LENGTH, VALIDATION?

Absolutely yes for client facing cites

Things to clarify

- Unique(+case sensitive)
- Required
- Min, Max
- Allowed symbols
- Languages
- Default values
- Messages texts

First you work for your reputation, then your reputation starts working for you

SEEING GENERAL PICTURE

THE CLASSIC CONTEXT-FREE QUESTIONS

The traditional newspaper reporters' questions are:

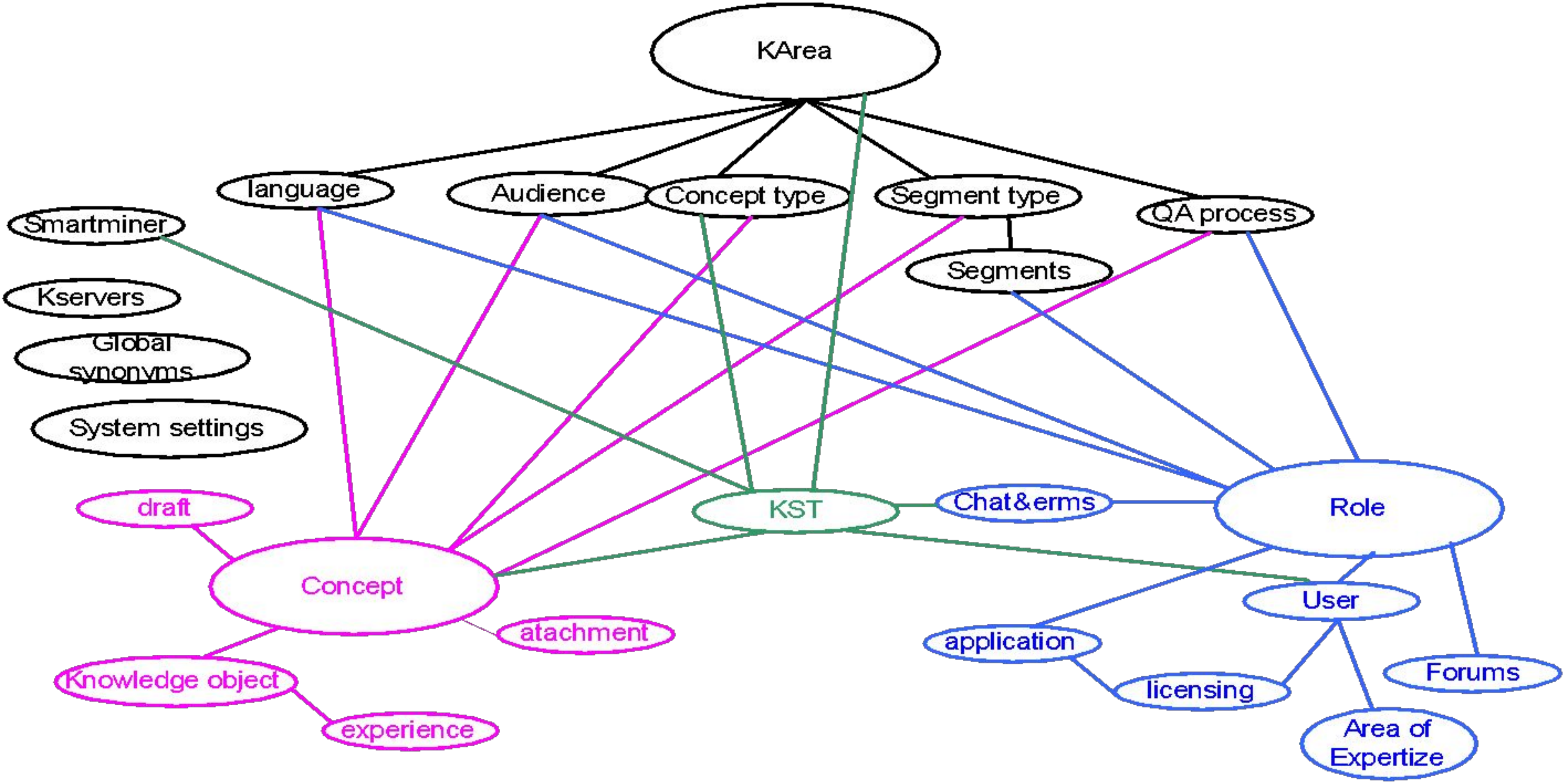
- – Who
- – What
- – When
- – Where
- – How
- – Why

For example, *Who will use this feature? What does this user want to do with it? Who else will use it? Why? Who will choose not to use it? What do they lose? What else does this user want to do in conjunction with this feature? Who is not allowed to use this product or feature, why, and what security is in place to prevent them?*

SEEING GENERAL PICTURE

- To ask a question of standing , it is necessary to go beyond the requirements of
- To see the image as a whole (a piece of the puzzle does not fit)

MODEL OF APPLICATION - EXAMPLE



1 page model of a system described in 100+ pages functional spec and tested by ~10 people. Shows relations and system as a whole.

MODEL

- Draw items, not screens
- Draw process
- If something is getting too complex, you are not seeing a model yet. Models can be of different scale.
- Main thing happens in your head: you digest requirements, and begin to understand your system
- Draw what is not clear

HOW TO REVIEW REQUIREMENTS

HOW TO READ THE REQUIREMENTS

1. First reading- with pencil
2. Resolve some questions
3. General picture-draw it
4. Second reading
5. Final list of questions - edit it

WORKING WITH ANSWERS

- Work with answer as a new requirement
- Are there any contradictions with what is already know?
- Modify test cases to incorporate answers
- Ask more questions (sometimes you get thread of 5+ questions/answers to resolve problem) Never leave a question unrecorded, at least at your personal sheet.
- Where will you put all your questions so they do not get lost – me –notebook?
- With all the answers combined, does the picture stay logical?
- What else should we check?

Всем спасибо за внимание!