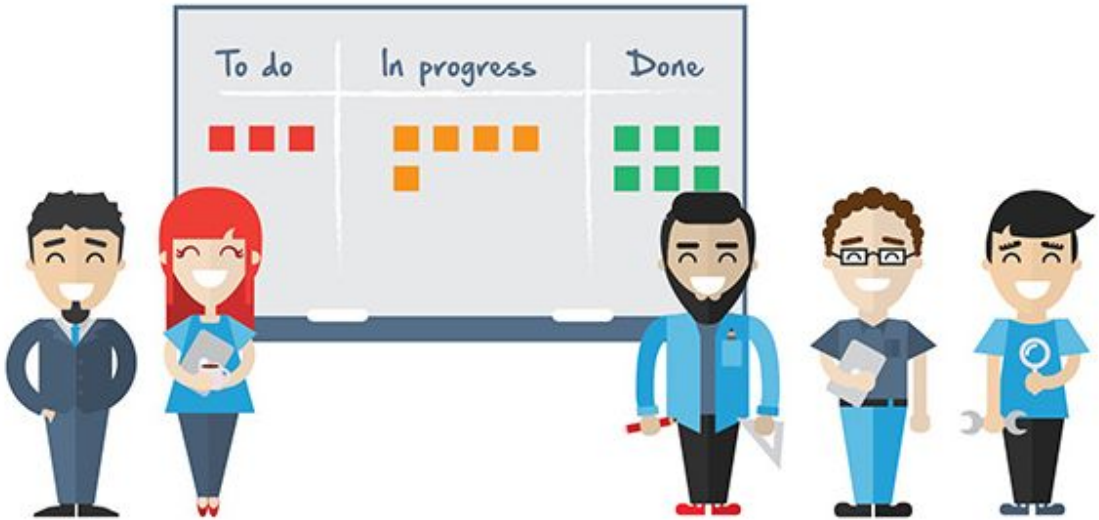
Lesson 5
Requirements and Estimations



Kit center

Requirement is

Software Requirements is a field within software engineering that deals with establishing the needs of stakeholders that are to be solved by software.

The IEEE Standard Glossary of Software Engineering Terminology defines a requirement as:[1]

1. A condition or capability needed by a user to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3. A documented representation of a condition or capability as in 1 or 2.

Types of software requirements (page 1)

1. **Business Requirements (BR)**
   - These are high-level business goals of the organization building the product, or the customer who commissioned the project
   - These are usually provided as a single page of high-level bullets

2. **UI Requirements (UIR)**
   - User interface specs are not considered “requirements” in *traditional* requirements management theory

Types of software requirements (page 2)

3. **Functional Requirements (FR) – Use Cases**
    - These cover the functionality of the product in detail.
    - Use cases are one of the best ways of documenting functional requirements
    - Depending on the product being built, FRs can run several hundred pages

4. **Non-Functional Requirements (NFR)**
    - These are not related to the “functionality” of the product – but cover goals such as Reliability, Scalability, Security, Integration, etc

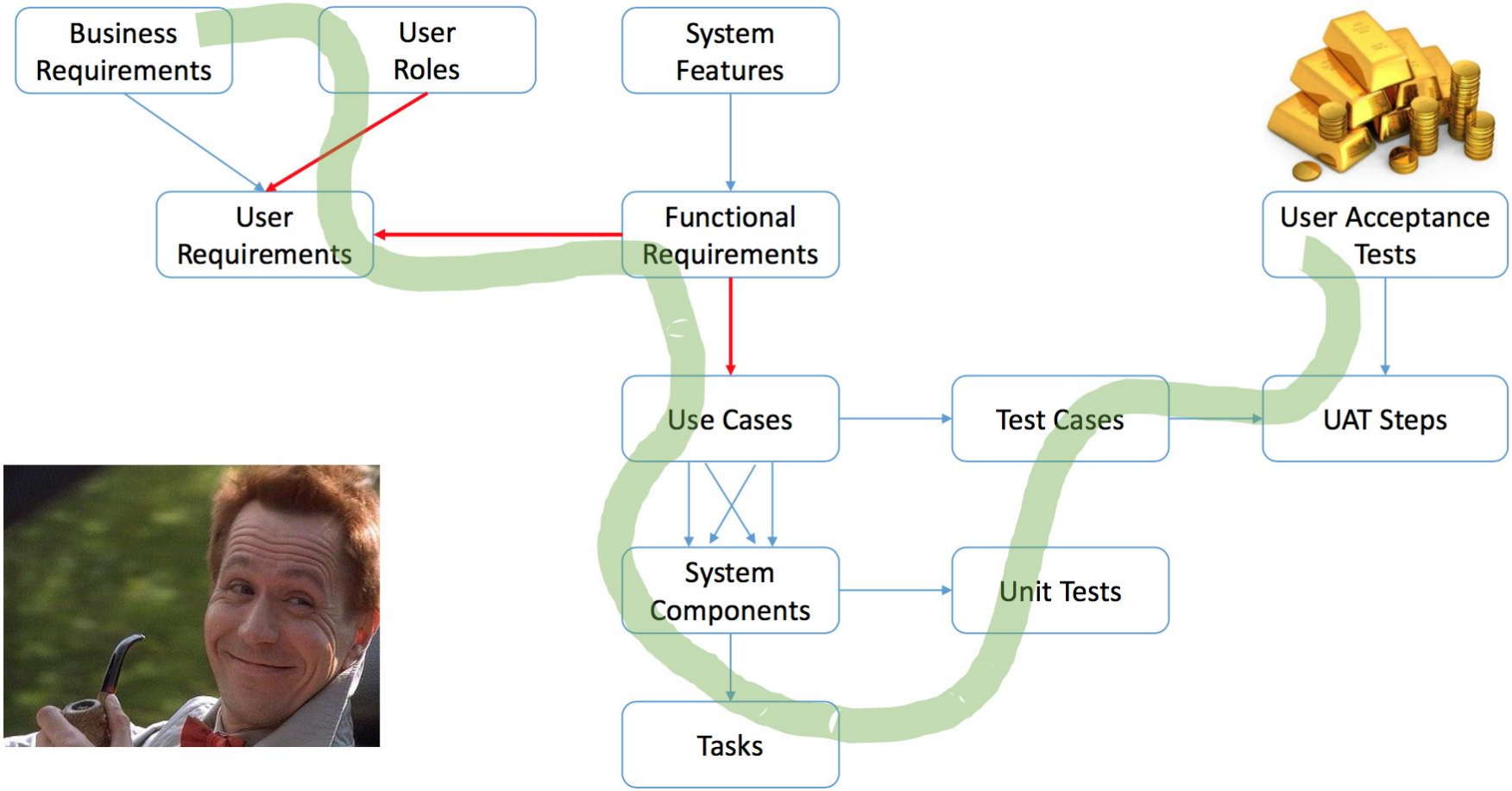
Трассировка

Business Requirements

Functional Requirements

System Components

User Acceptance Tests


Business Requirements
User Roles
System Features
User Requirements
Functional Requirements
User Acceptance Tests
Use Cases
Test Cases
UAT Steps
System Components
Unit Tests
Tasks

What Are “Features” then?

As you can see, the list above does not include a popular term that is frequently used in software projects: **Features**. Wondering why? The reason is simple…

A “Feature” is just a group of functional requirements (FRs) that together meet a specific customer need.

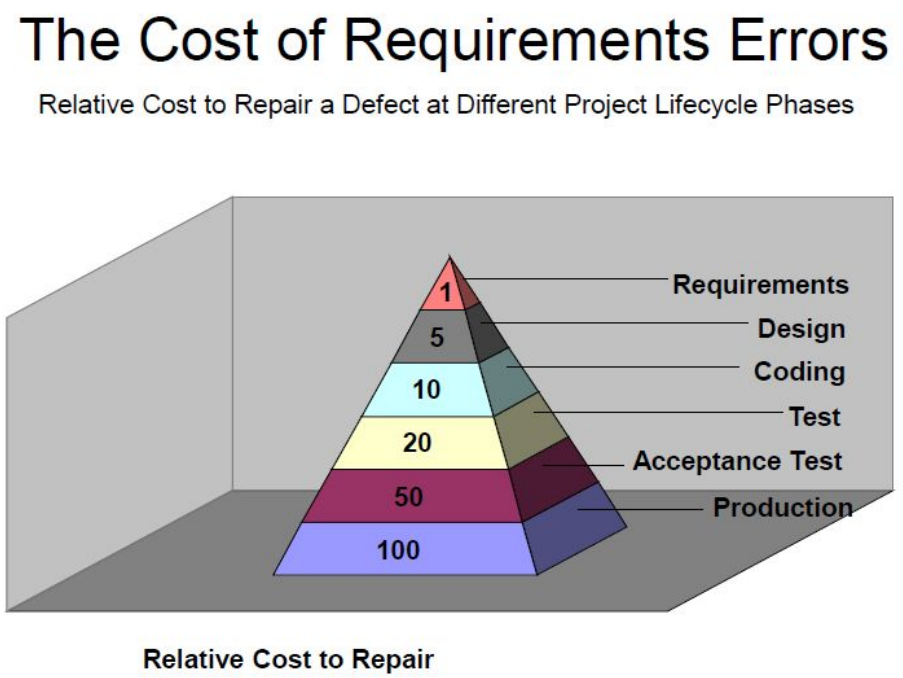
Why do we need to analyse inputs?

Business

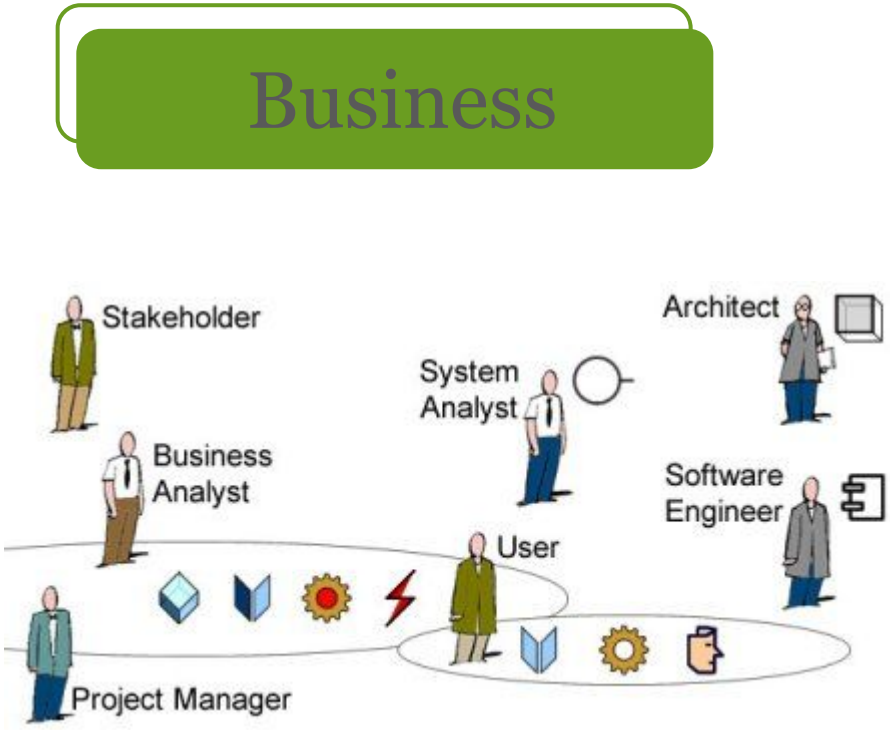
- Requirements
- User stories

Technical

- Technical specifications and diagrams



Business Requirements Document (BRD)



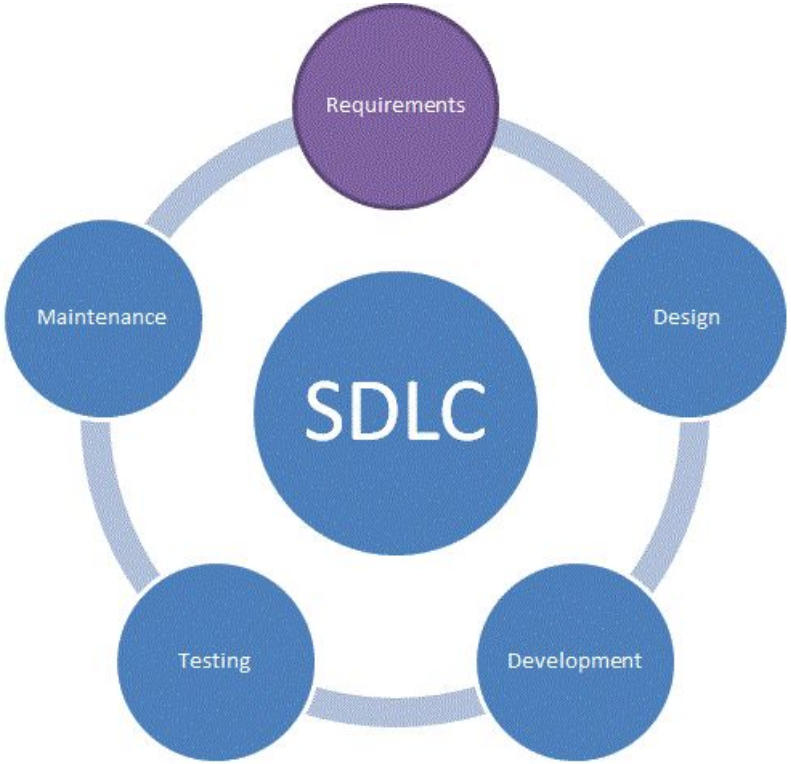
Tips for Business Requirements

- Keep business requirements simple, clear and concise. Simple sentences like ‘the application shall email the customer after they have placed an order’ is often enough. You don’t need to specific every last field on the email, or if you do it might be better to do it in a mockup (graphical form) rather than with the written word.
- Don’t go overboard on detail. Its tough for some people not to go into a lot of detail, but if you do you may lose people.
- Use lots of pictures, diagrams and (if you can) screen mockups. Even if these are just done in PowerPoint, it’s better than nothing.

Requirements in SDLC

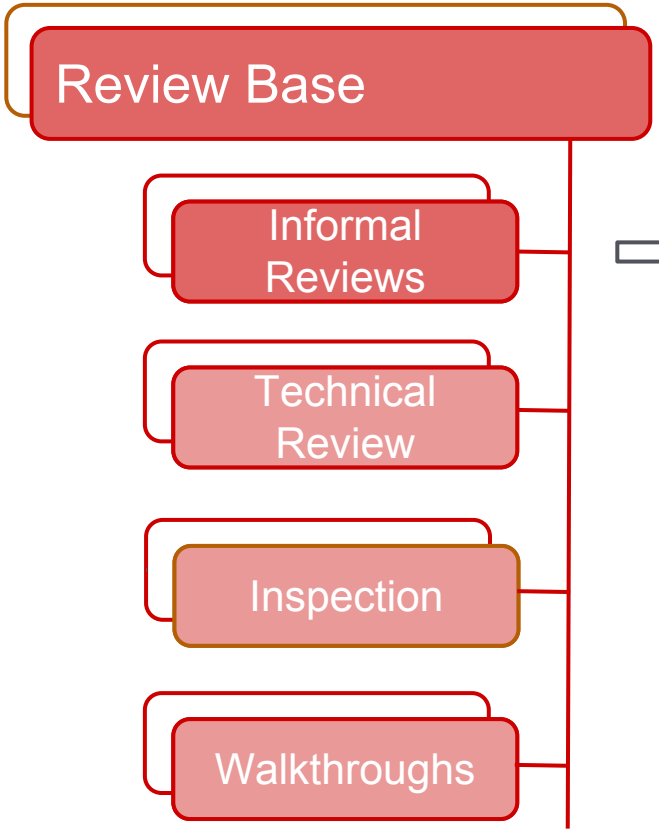
- Specify Priority Each requirement must specify a level of importance to stakeholders.

- Use traceability and version control The requirement meets all or part of a business need as stated by stakeholders and authoritatively documented. Has version and priority.



To analyse inputs we use

Static Testing techniques



- Author creates a requirement list
- QA team analyzes it
- Questions and issues are send to the author
- Autor makes updates in his document

Requirements analysis

Practical part

Requirements characteristics

Unitary Atomic Consistent (последовательный) - the requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation

Unambiguous (недвусмысленный) - It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation. Vague subjects, adjectives, prepositions, verbs and subjective phrases are avoided. Negative statements and compound statements are avoided.

Complete - The requirement is fully stated in one place with no missing information.

Testable (тестируемый)-The implementation of the requirement can be determined through basic possible methods: inspection, demonstration, test (instrumented) or analysis (to include validated modeling & simulation).

Transfer Money

Help with transfer Money

Step 1 of 3 - Input your transfer details

Please note - Transfers made after 23:45 will be processed the next day

Please complete the following details:

You want to transfer from: BANK A/C
KATE THOMAS 44-55-12-43

You want to transfer to: FLEX SAV PRE
KATE THOMAS 44-55-12-43

What's the amount? 100 00 Payment limits

When do you want to make the transfer? (up to 12 months in the future)

Immediately v
or
Future date DD MM YYYY

BACK CONTINUE

Requirement Description	Only 1 req-t	Only 1 meaning	Test able	**Question**
Ability to view “Transfer From” and “Transfer To” a/c details (A/C Type, A/C Name and A/C Number) as a read-only				
Ability to make transfer - Immediately				
Ability to make transfer - Future date				
Provision to navigate back to Account page and change transfer To a/c				
Provide a link to information for Help				
Confirmation page should respond quickly enough				

Requirement Description	Only 1 req-t	Only 1 meaning	Test able	Question
Ability to view “Transfer From” and “Transfer To” a/c details (A/C Type, A/C Name and A/C Number) as a read-only				
Ability to make transfer - Immediately				
Ability to make transfer - Future date				Only 1 meaning: No clarity / Not clear Testable:How far in the future should I test? How many months?
Provision to navigate back to Account page **and change transfer To a/c**				Only 1 req-t: Multiple requirements Only 1 meaning: Not clear transfer, transfer to a/c (which a/c)
Provide a link to information for Help				Not clear which Help
Confirmation page should respond quickly enough				How quick? How many users

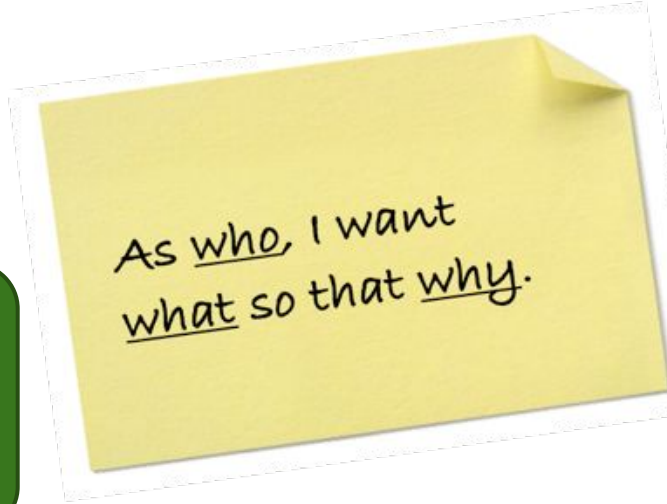
User story

User story is a description in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function.

Format of user story

"A [illegible] fit >"

As a *customer* I want *to create tax entries* so that tax collectors will confirm my payment.



When analyzing user's story: INVEST

Letter	Meaning	Description
I	Independent	The user story should be self-contained, in a way that there is no inherent dependency on another user story
N	Negotiable	User stories, up until they are part of an iteration, can always be changed and rewritten.
V	Valuable	A user story must deliver value to the end user.
E	Estimatable	You must always be able to estimate the size of a user story.
S	Small	User stories should not be so big as to become impossible to plan/task/prioritize with a certain level of certainty.
T	Testable	The user story or its related description must provide the necessary information to make test development possible.

User's story after analysis

As a *customer* I want *to create tax entries from my account* so that tax collectors will confirm my payment and my taxes are paid.

Acceptance criteria:

1. Tax payment can be created with needed description when we use check box
2. Messages are sent and approved by 3 selected collector systems.
3. Description field becomes disabled when we select to send a tax message.
4. Not-tax payments are still can be sent.

Home Work

http://transpologic.tk/en

1. Understand your feature
2. Prepare User Stories for your feature
3. http://scrumtrainingseries.com/ - watch videos, focus on Sprint Planning