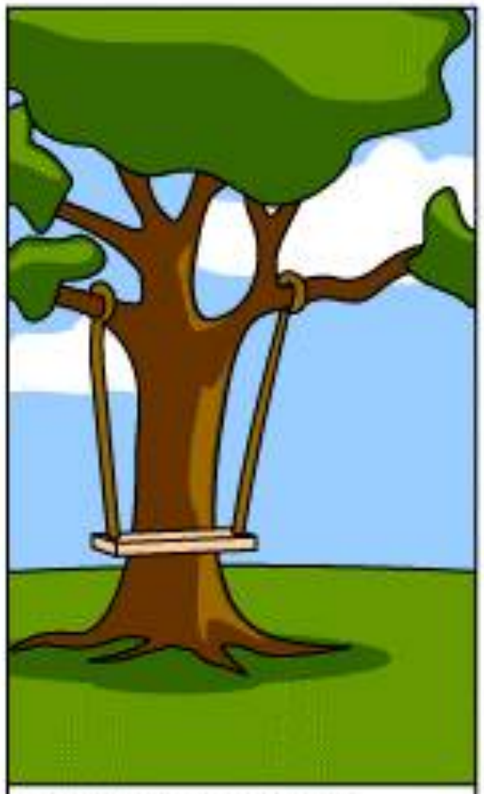
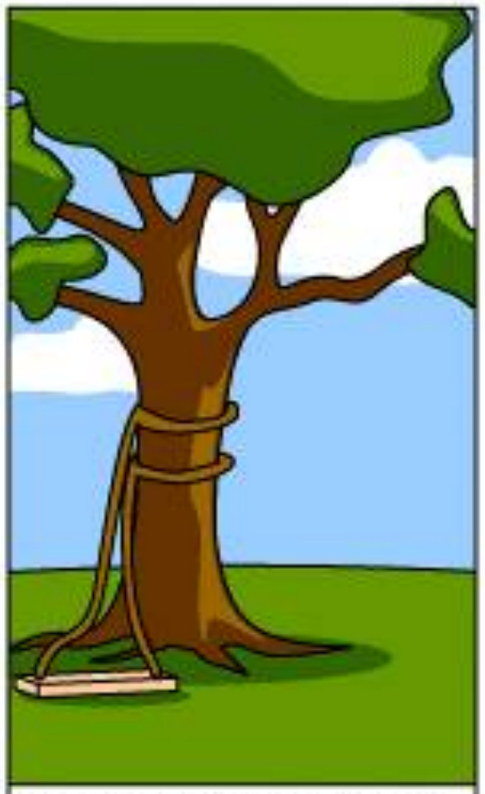
How the customer explained it



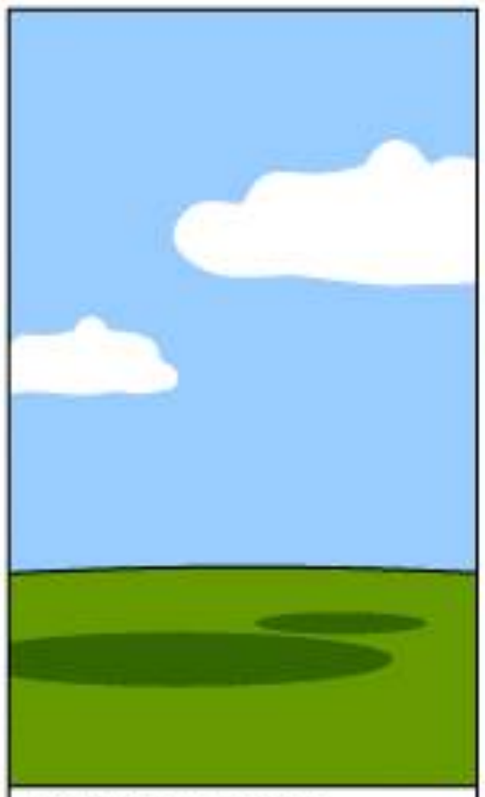
How the Project Leader understood it

How the Analyst designed it

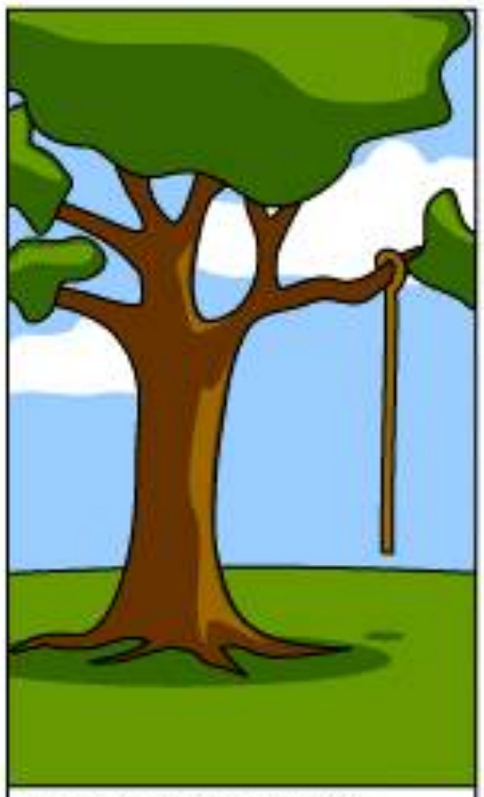


How the Programmer wrote it

How the Business Consultant described it

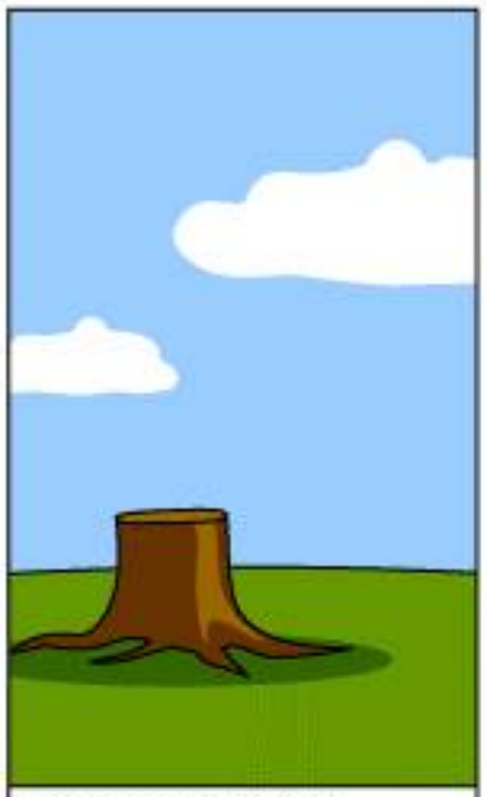


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed

Requirements

Thierry Sans

Software Requirement in Agile Development

- **Personas**
  knowing who the users are in terms of behaviours and goals towards the software

- **User Stories**
  identifying the software features that will match the users' behaviours and goals

Personas

What are personas?

- archetypical users of a system
- representation of users goals, behaviors and abilities

➡ Personas are used for product design and marketing

Why defining personas?

- effective software is designed for specific people
- help understand requirements
- separate market segments
- facilitate communication within team members

How to write personas?

1. collect data on real (potential) users
2. identify the different roles
3. for each role, create a persona for the most typical user
4. for each role, identify whether there should be an additional type of user and eventually create its persona
5. In the end, review your personas and decide whether you really need them or not

What information do you find in a persona

- fictional name
- demographics : gender, age, education, ethnicity and family status
- job title and responsibilities
- physical, social and technology environment
- goals when using the system
- attitude towards technology, domain, etc.
- technical skills

Exercises

- Analyzing personas
  `personas-exercise.pdf`

- Writing personas
  `grademywork.pdf`

User stories

Principles of Agile Development

- Agile sets a **fast and incremental development**

Step 1: build the first feature, validate it, release it
Step 2: build the second feature …
and so on

1 user story ~ 1 feature ~ 1 software release

Definition

User story : a high-level definition of a requirement, an elemental task that a user can do with the software

➡ It should focus on who, what, why but not how

"As `[persona]` (a `[role]`), I want to `[goal]`, so that `[benefit]`"

this part could be optional
but highly recommended

INVEST in good user stories

- **Independent :** a user story should not be (too) dependent on another one
- **Negotiable :** a user story should be as concise and as precise as possible. Yet, they are not contracts and can hide some non trivial details
- **Valuable :** a user story should enhance the user experience of the targeted persona
- **Estimable :** developers should have a rough estimate of the amount of time and effort that will be required to implement the user story
- **Small :** but yet not too small by the time you will add them to your sprint backlog
- **Testable :** developers should have a good idea about how to test whether the product implements the user story

Refining user stories

Users Stories can be **refined to capture more details**

> *As a shopper, I would like to have a receipt for my purchase*

Could become

> *As Jerry (a budget shopper), I would like to have a receipt for my purchase that lists the price and any discounts, so that I have a record of my purchases. The receipt should also have a date and the name and price of all articles purchased.*

Breaking down user stories

Some user stories are too big by themselves and should be broken down into several, more specific, user stories

> *As a store manager, I want to offer items on sale for a reduced price for a limited time, so that I can increase traffic in the store. The price reductions can be a percentage reduction or a straight reduction to a lower price.*

Could become

- *As Bina (a store manager), I want to offer a flat rate discount on items between Dec 26 and Dec 31.*
- *As Bina (a store manager), I want to offer a percentage discount to senior citizens on Tuesdays.*

Dealing with conflictual user stories

- *As a customer, I want to have a help button that summons a cashier if I'm having difficulty.*

- *As a retailer, I don't want to have a help button that summons a cashier because it wastes employee time.*

Identify bad user stories

- *As a user, I want the application to do* [everything that it is supposed to do]

- *As a user, I want to be able to login*
- *As a user, I want the application to work on iOS*
- *As a user, I want the application to work well*
- *As a user, I want the application UI to be user friendly*

Identifying "epic" user stories

As a student, I want to be able type my solution online, test it and submit it so that it can be graded and the score sent back

Triaging user stories

Defining what should build first in terms of importance of the feature

- *As Alice (student), I want to see my grade for a given course*
- *As Alice (student), I want to see the grade distribution for a given course*

Sometimes, you need to think about user stories dependencies
and their corresponding software release

- *As Alice (a student), I want to browse the course catalogue*
- *As Bob (an administrator), I want to add new courses to the course catalogue*

Exercises

- Analyzing user stories
  `userstories-exercise.pdf`

- Writing user stories
  `grademywork.pdf`