# User Stories

Thursday, September 27

Oregon State University

# Announcements

First interview is scheduled

You can now form partial teams

Sprint 1 has been released

Sprint 0 and teams are due tomorrow @ 5PM.

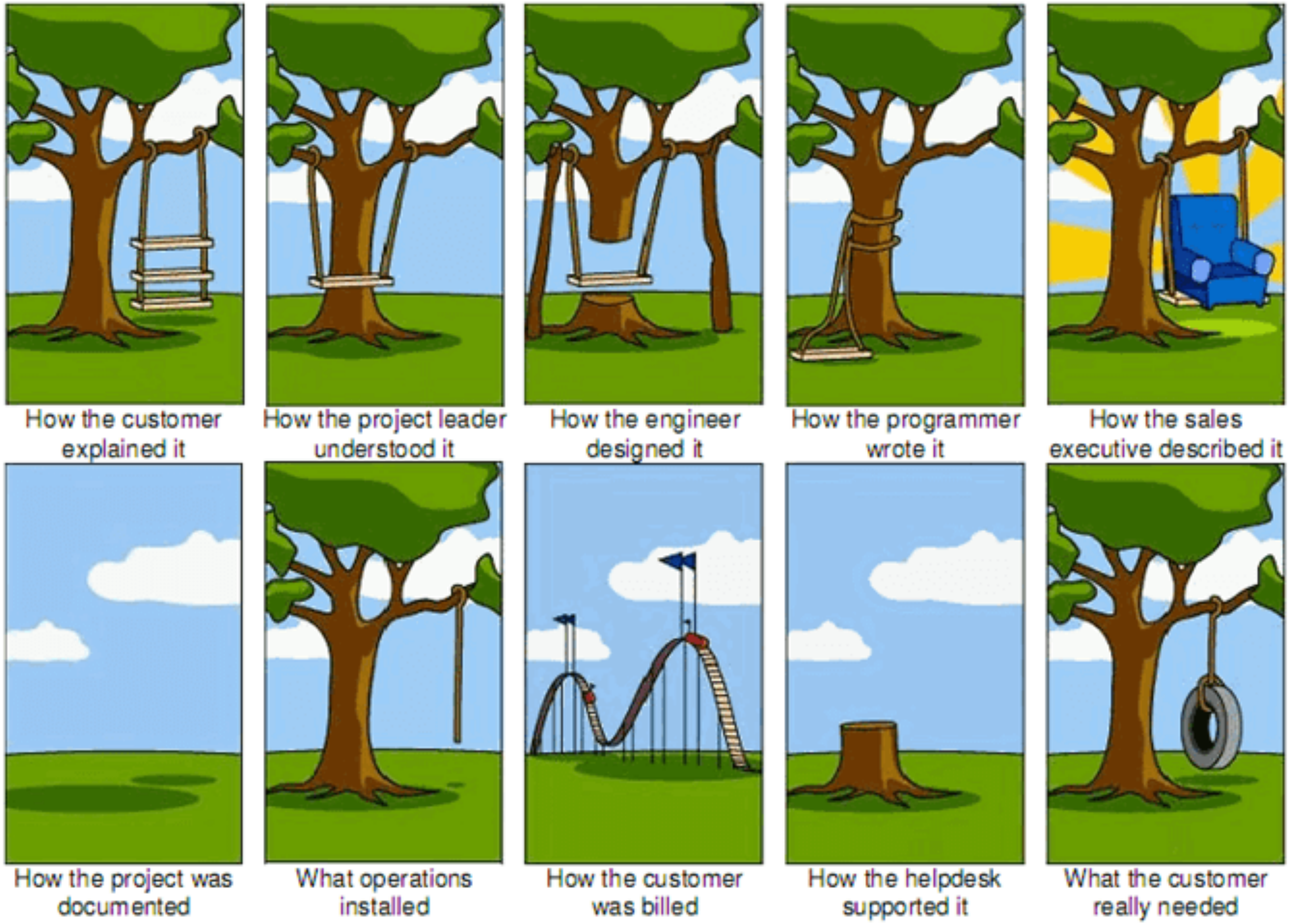
# Christian Romming

Christian Romming founded Etleap in 2013. Before founding Etleap, Romming was the CTO of an ad-tech company, where he recognized the available solutions for building data pipelines required monumental engineering resources to implement, maintain, and scale.

Their mission is to make data analytics teams more productive. Their ETL solution lets analysts build data warehouses without internal IT resources or knowledge of complex scripting languages. This reduces the time of typical ETL projects from weeks to hours, and takes out the pain of maintaining data pipelines over time.

# Overview of Sprint 1

# Requirements



How the customer explained it

How the project leader understood it

How the engineer designed it

How the programmer wrote it

How the sales executive described it

How the project was documented

What operations installed

How the customer was billed

How the helpdesk supported it

What the customer really needed

# What are requirements?

A requirement is a service, function or feature that a **user needs.**

There are 2 types:

**Functional:** Describe the behavior of a system

**Non-functional:** Specifies a perfomance characteristic.

# Non-functional requirements

Also referred as **quality attributes** or **characteristics**

Reliability

Robustness

Efficiency

Usability

Maintainability

Testability

Flexibility

Portability

Interoperability

Reusability

# Non-functional requirements

The system…

*…must handle a peak load of 1000 requests/second*

*…must be available 99.9% of the time*

*…must answer requests in max 500 ms.*

SLA's (Service Level Agreement)

# Functional requirements

What the system must do.

They describe the system behavior when the user **interacts** with it.

# User Stories

Describes a small piece of system functionality, in a simple and easy to read sentence.

It's written in **plain english,** so that the end user can understand.

# The Card

*As a [role], I want [some functionality] so that [I get some benefit].*

Written on 3x5" index cards

You stick them to a wall, or whiteboard

# The Conversation

You use the card to start a **conversation** with the product owner.

This helps you refine the user story

# The Confirmation

How do you know when you are **done** with a story?

This is **negotiated** with the product owner.

Can be written on the back of the card.

Should be automated, as much as possible

*Acceptance Test.*

# The definition of *Done*

A clear and concise list of requirements that *[software]* must adhere to for the team to call it **complete.**

*[software]* = user story / milestone / release / feature

# Example Definition of *Done*

All tests are passing

Unit tests added for new features

Code has been peer reviewed

User Documentation has been updated

Accepted by Product Owner

# Writing good user stories

Don't go into too many details when writing the card

That's what the conversation is for

Technical tasks are not user stories

Don't skip the conversation

# INVEST

- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimable
- **S**mall (Sized appropriately)
- **T**estable

# Independent

User stories must stand on their own

**No dependencies** between stories

Not always possible, but worth trying

You should be able to schedule them in any order.

# Negotiable

Details can be negotiated during development

The conversation is critical here

# **V**aluable

The user story needs to bring **value** to the user

# Estimable

You should be able to provide a time-frame for completion

This guarantees that you can finish during a sprint

# Small

Here's where the 3x5" card come in handy

A user story needs to fit into a sprint (2 weeks in this class)

A user story should fit in a week, at most

Too big == !estimable

# Testable

Part of the definition of done & the acceptance criteria

If we can't test, can we be sure we understood the task?

# Your turn

Write down 1 user story for the banking app

https://vimeo.com/41800652