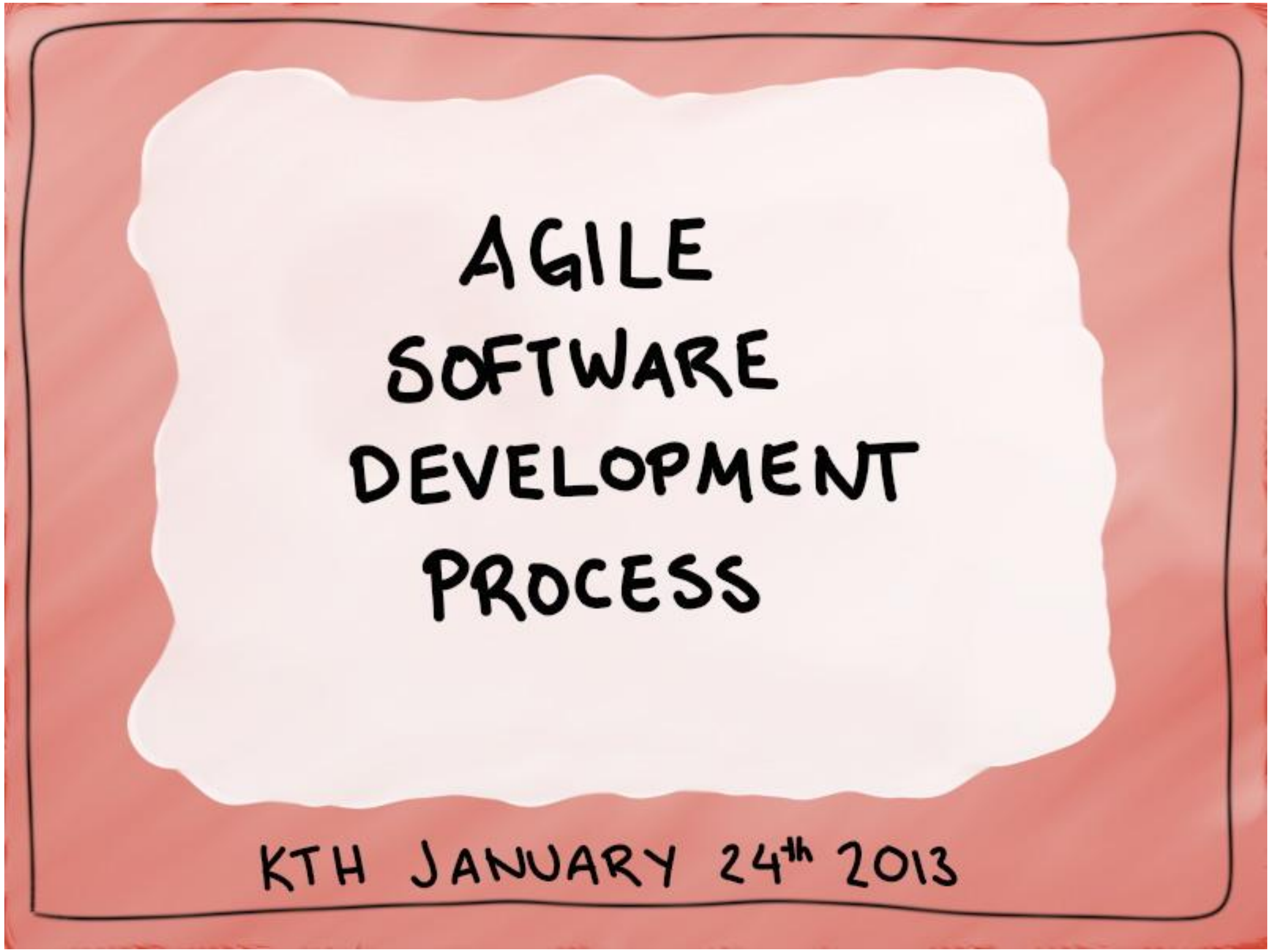
KTH January 24, 2013

How does Agile software development work?

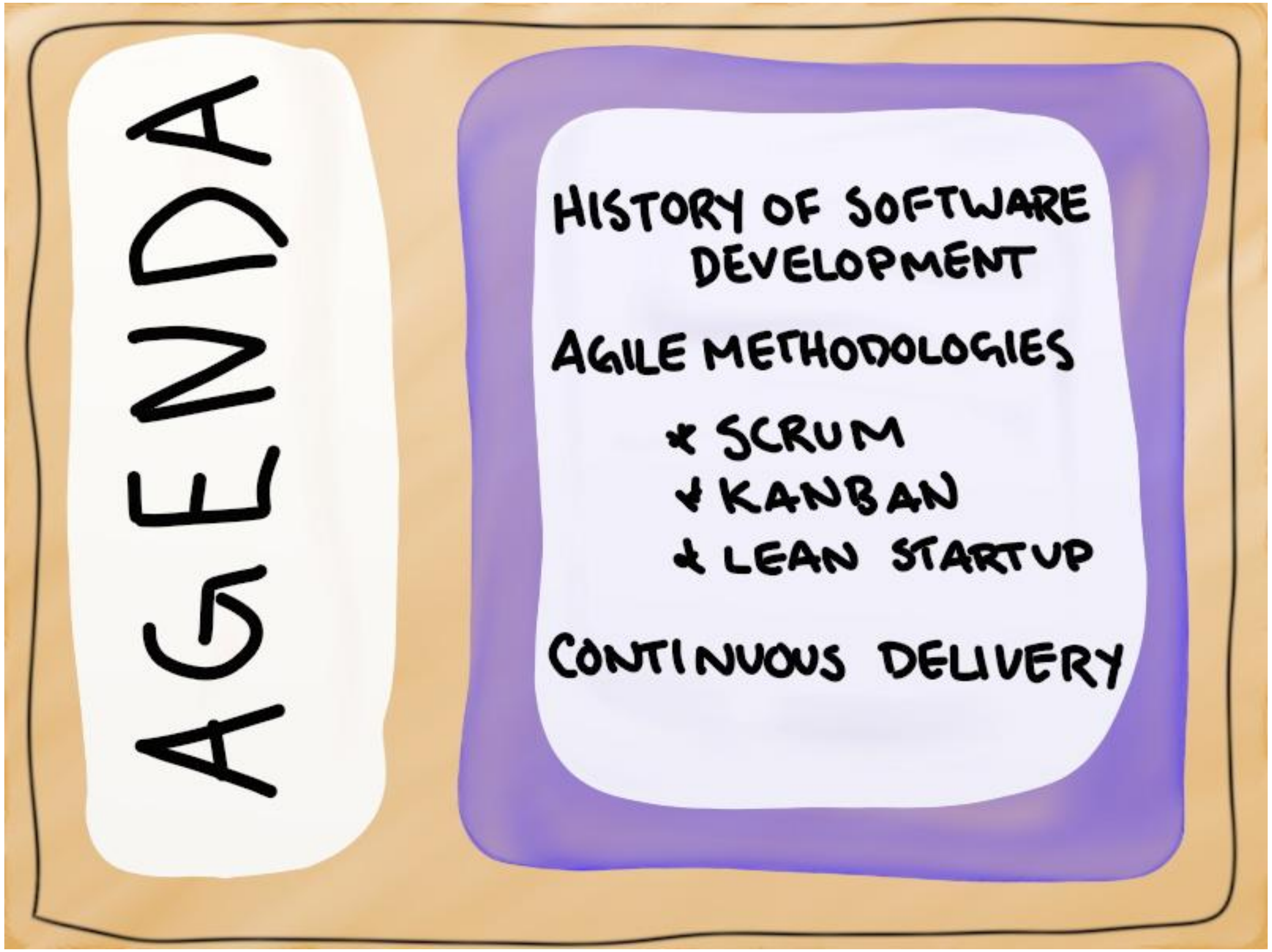


crisp

Yassal Sundman

Agenda

Let's have some fun :)



AGENDA

HISTORY OF SOFTWARE DEVELOPMENT

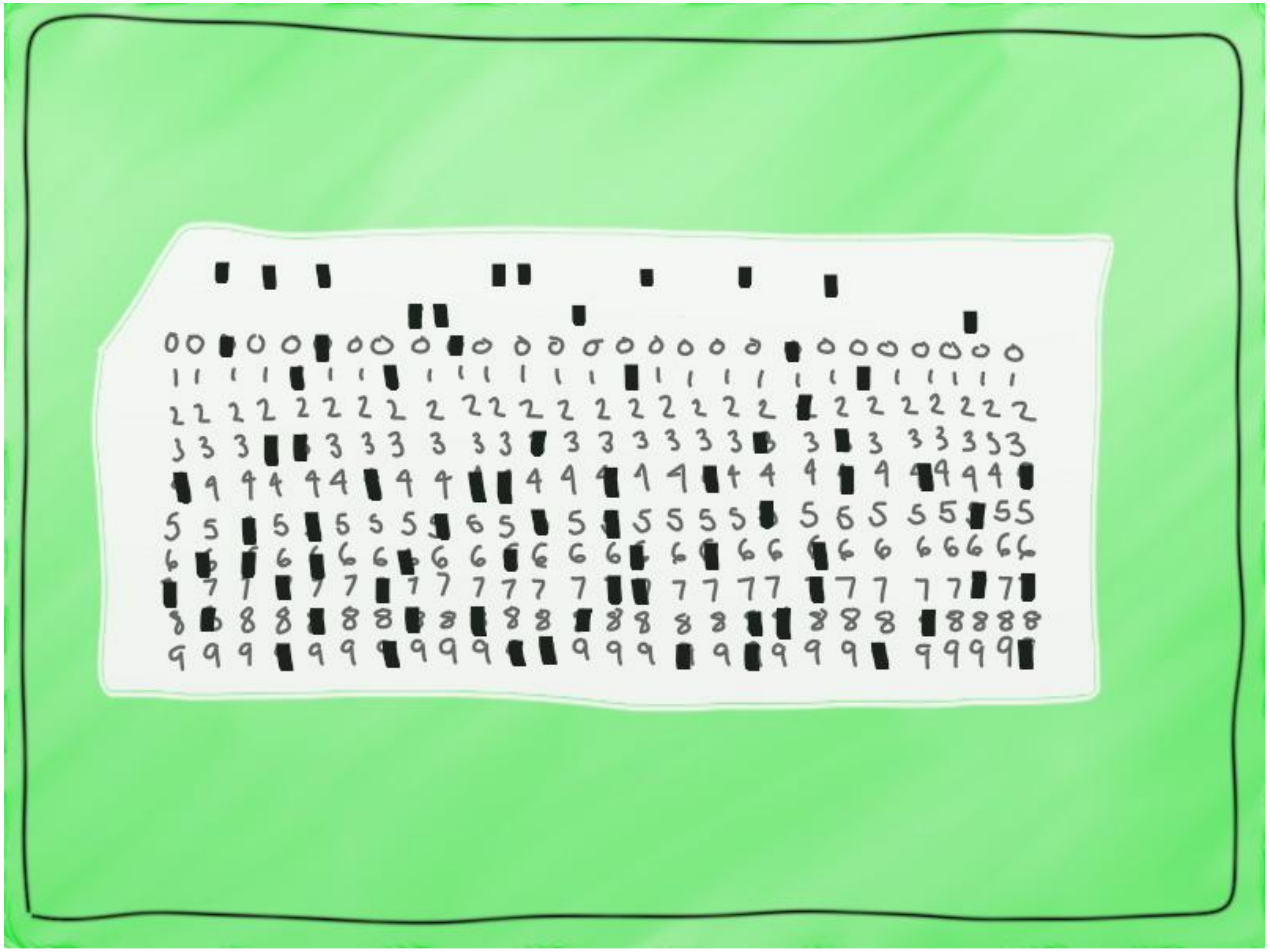
AGILE METHODOLOGIES

- SCRUM
- KANBAN
- LEAN STARTUP

CONTINUOUS DELIVERY

crisp

A Condensed History of Software Development



crisp

Yassal Sundman

1950s to 1960s
Computers are **expensive**, **single purpose** and have a long **turnaround time.**

150 CARDS/MIN !!!
IBM 407 ACCOUNTING MACHINE

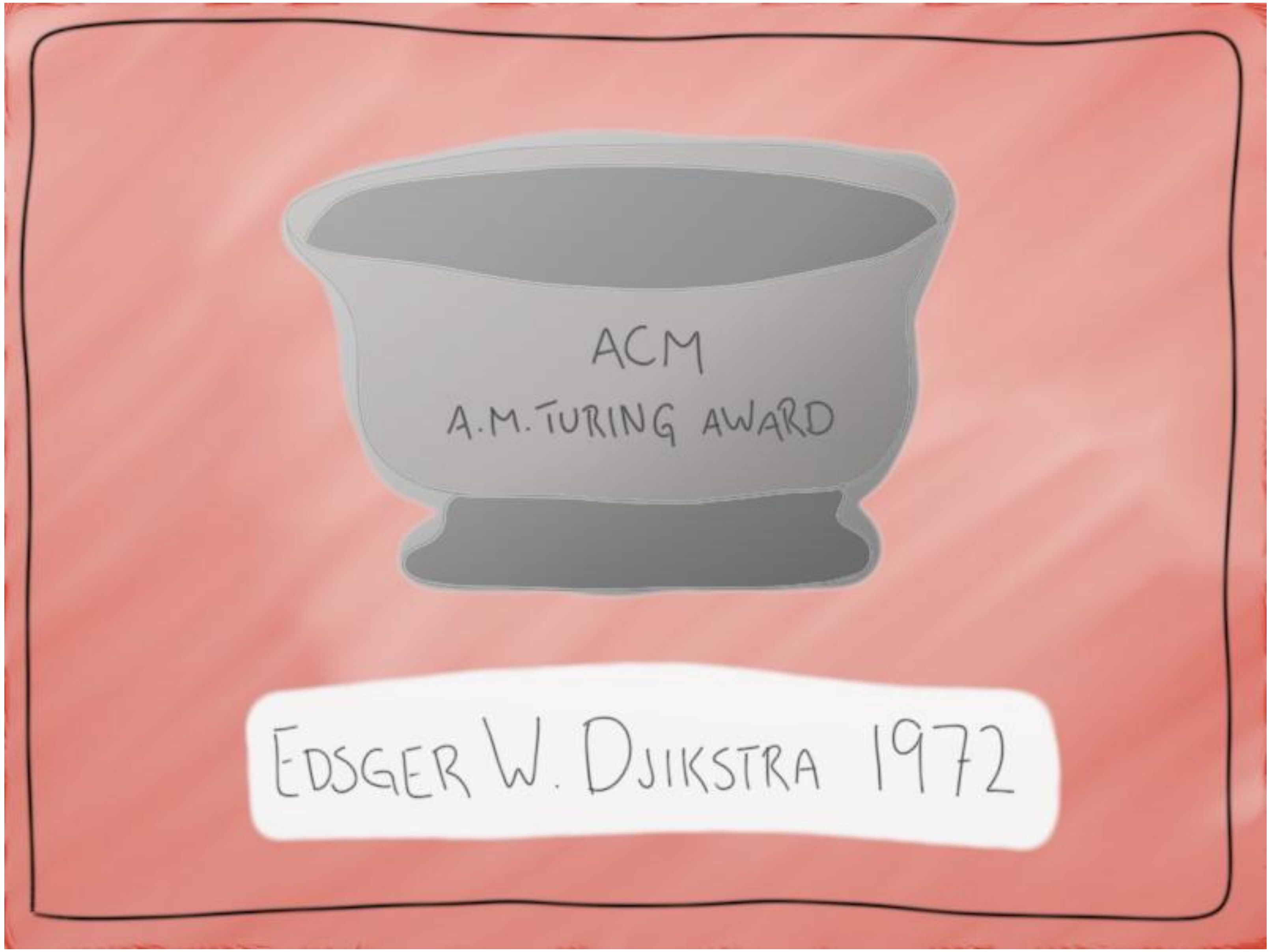
crisp

Yassal Sundman

1970s to 1980s

Edsger Dijstra's Turing Award Lecture in 1972 characterizes the **Software Crisis**:

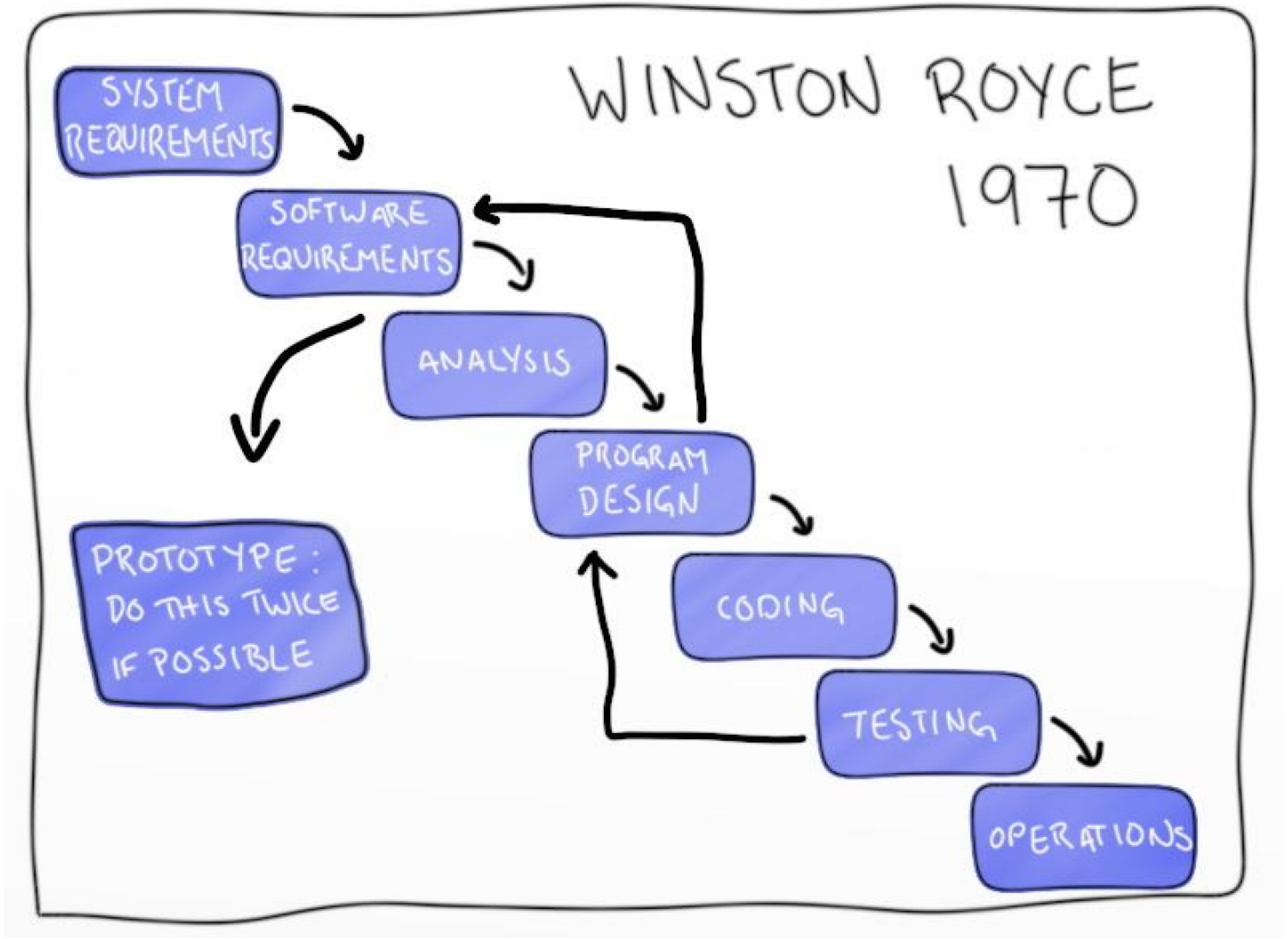
- Development is expensive and slow
- Programs are too complex and difficult to maintain
- Software is unreliable



1970 - Waterfall

1970, Winston Royce describes the waterfall model.

- Complete program design before analysis and coding begins
- Documentation must be current and complete
- Do the job twice if possible
- Testing must be planned, controlled and monitored
- Involve the customer
- Focus on documentation, as little movement possible between the different stages. Recognizes importance of the customer, introduces the idea of a prototype.



crisp

Yassal Sundman

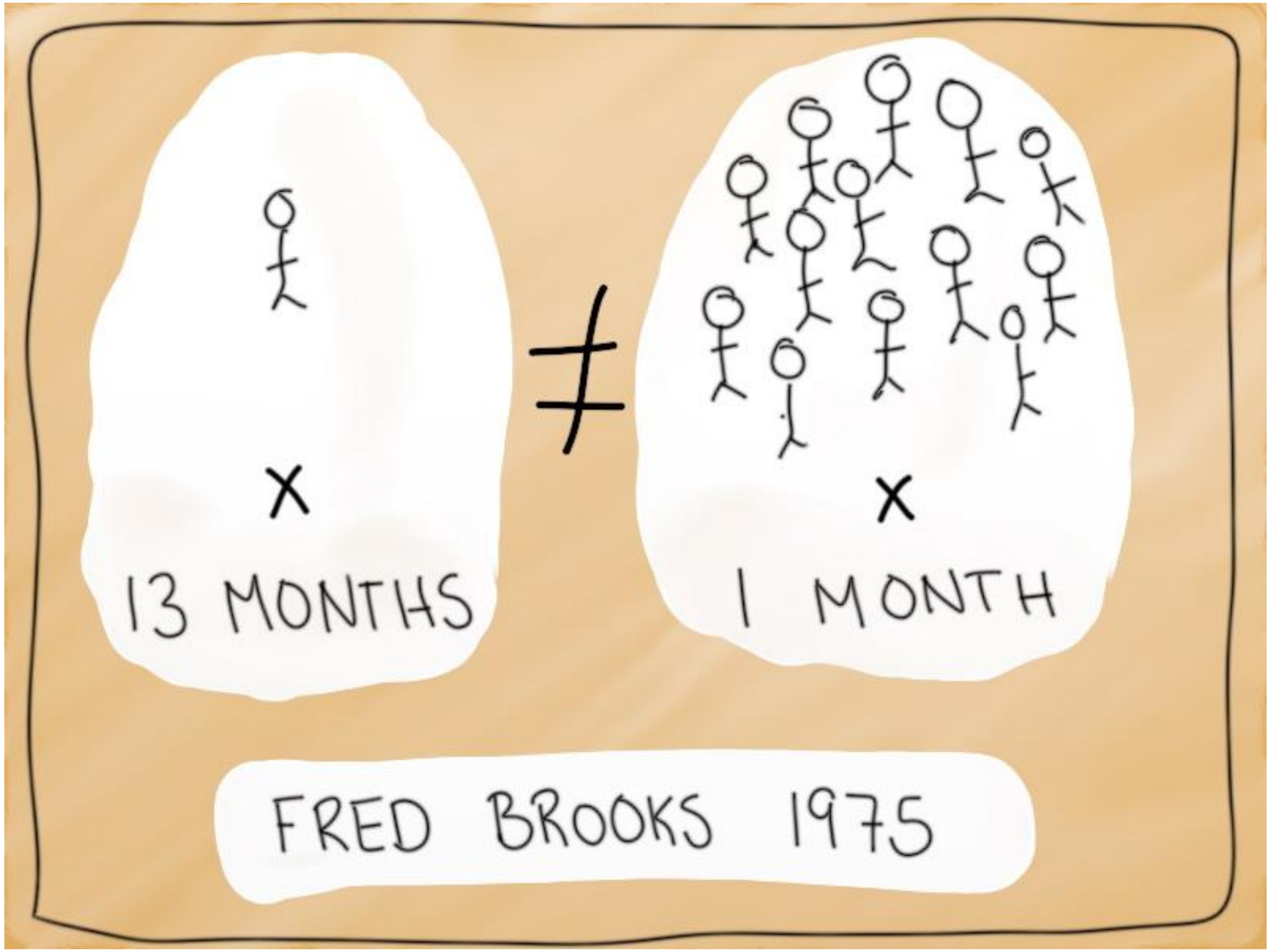
1975 - Mythical Man Month

Fred Brooks revolutionary book comes out. Most important takeaway: **Men and months are not interchangeable**.

Also coined **Brooks law**: adding manpower to a late software project makes it even later

Solutions:

- Documentation
- Specialized team members
- Separation of responsibilities

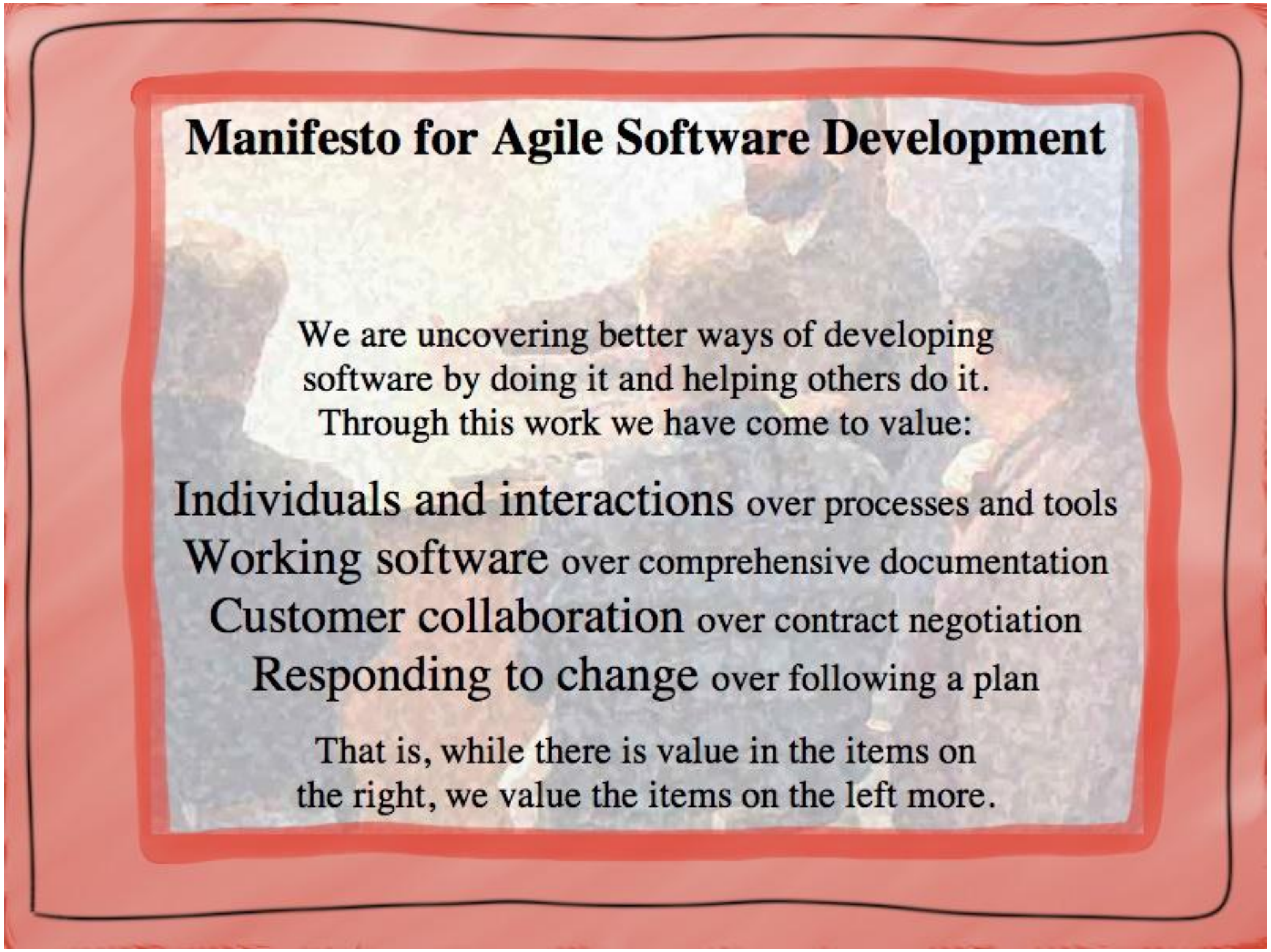


1990s - 2000s

The 90’s are a time when new methodologies are created and tested. Lean software development is being practiced with methodologies taken from product development.

In the mid 1990's Kent Beck formalizes XP, and Jeff Sutherland and Ken Schwaber formalize Scrum. In 1995 Fred Brooks revises his book – an iterative approach is better than waterfall.

In February 2001 the Agile Manifesto is written and signed by the newly formed Agile Alliance.



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

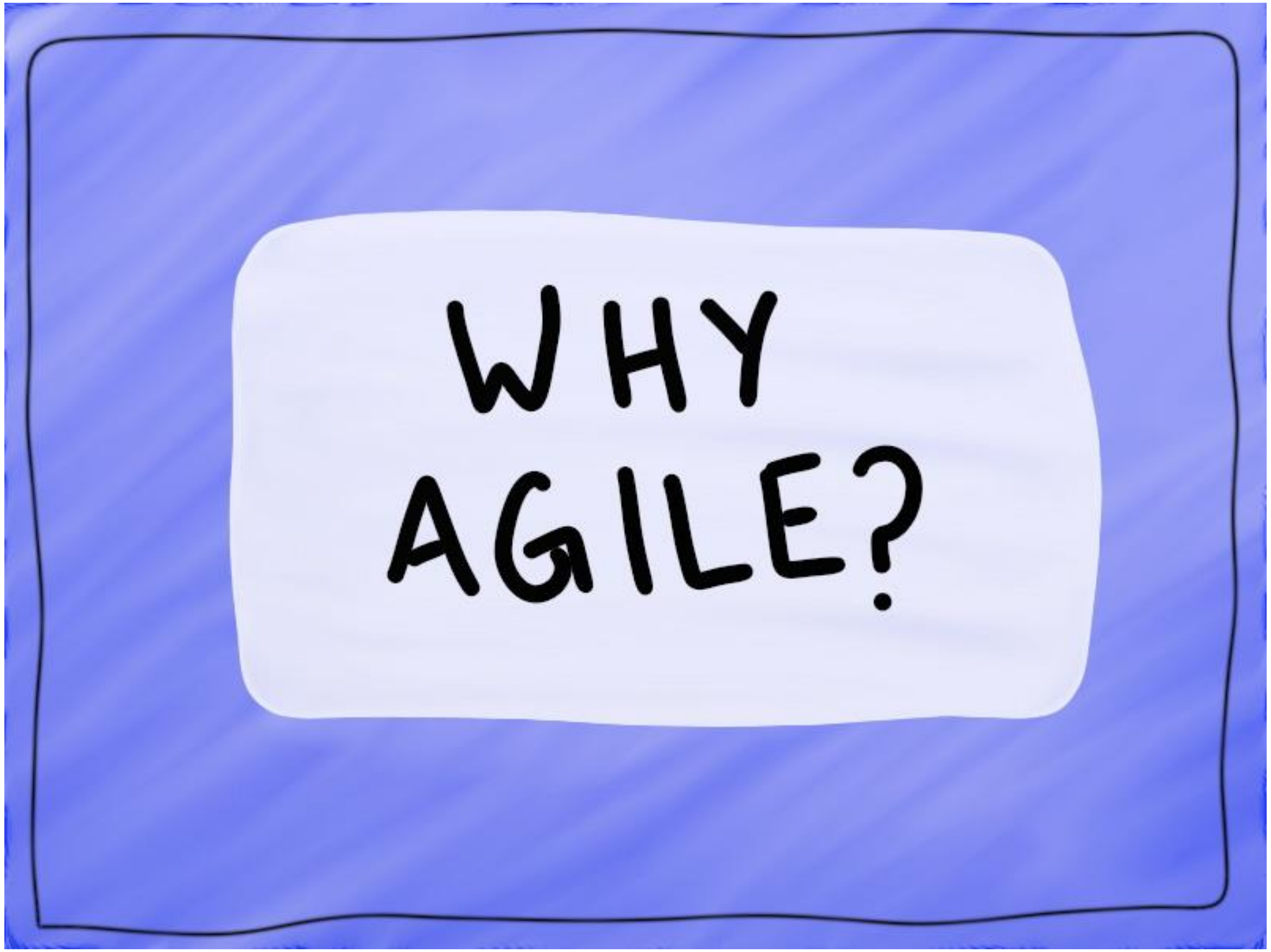
That is, while there is value in the items on the right, we value the items on the left more.

Why Agile?

Agile methodologies address the three main problems that characterize the ongoing **Software Crisis**:

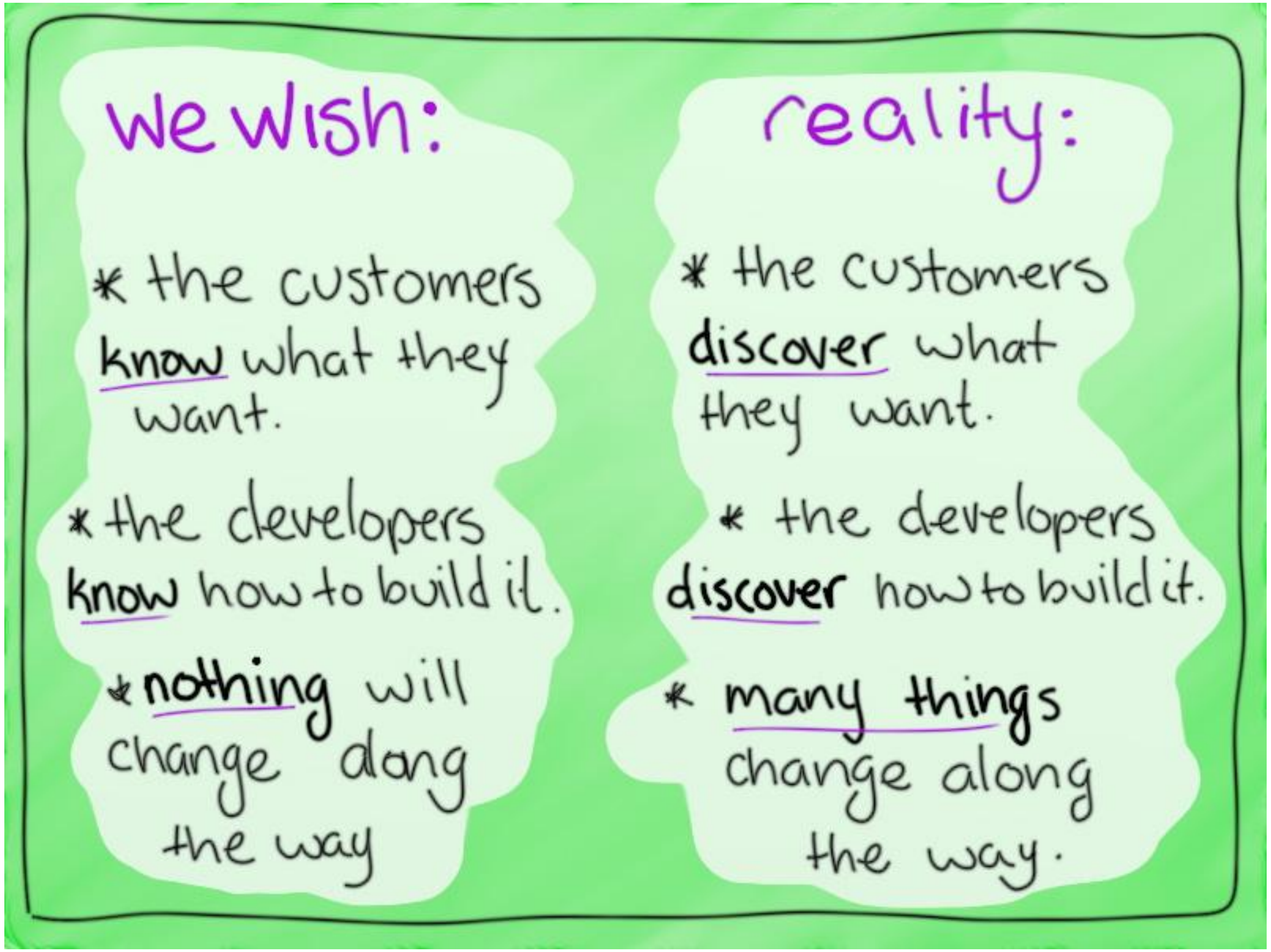
- Development is expensive and slow
- Programs are too complex and difficult to maintain
- Software is unreliable

Martin Fowler has a great blog entry explaining agile methodologies.



What We Wish vs. Reality

Our expectations are not in sync with reality. Waterfall builds on what we wish reality was, agile addresses the way software development usually is.


we wish:
* the customers know what they want.
* the developers know how to build it.
* nothing will change along the way
reality:
* the customers discover what they want.
* the developers discover how to build it.
* many things change along the way.

Yassal Sundman

Deliver Fast and Frequently

The first and third principle of agile are:

- Customer satisfaction by rapid delivery of useful software
- Working software is delivered frequently (weeks rather than months)

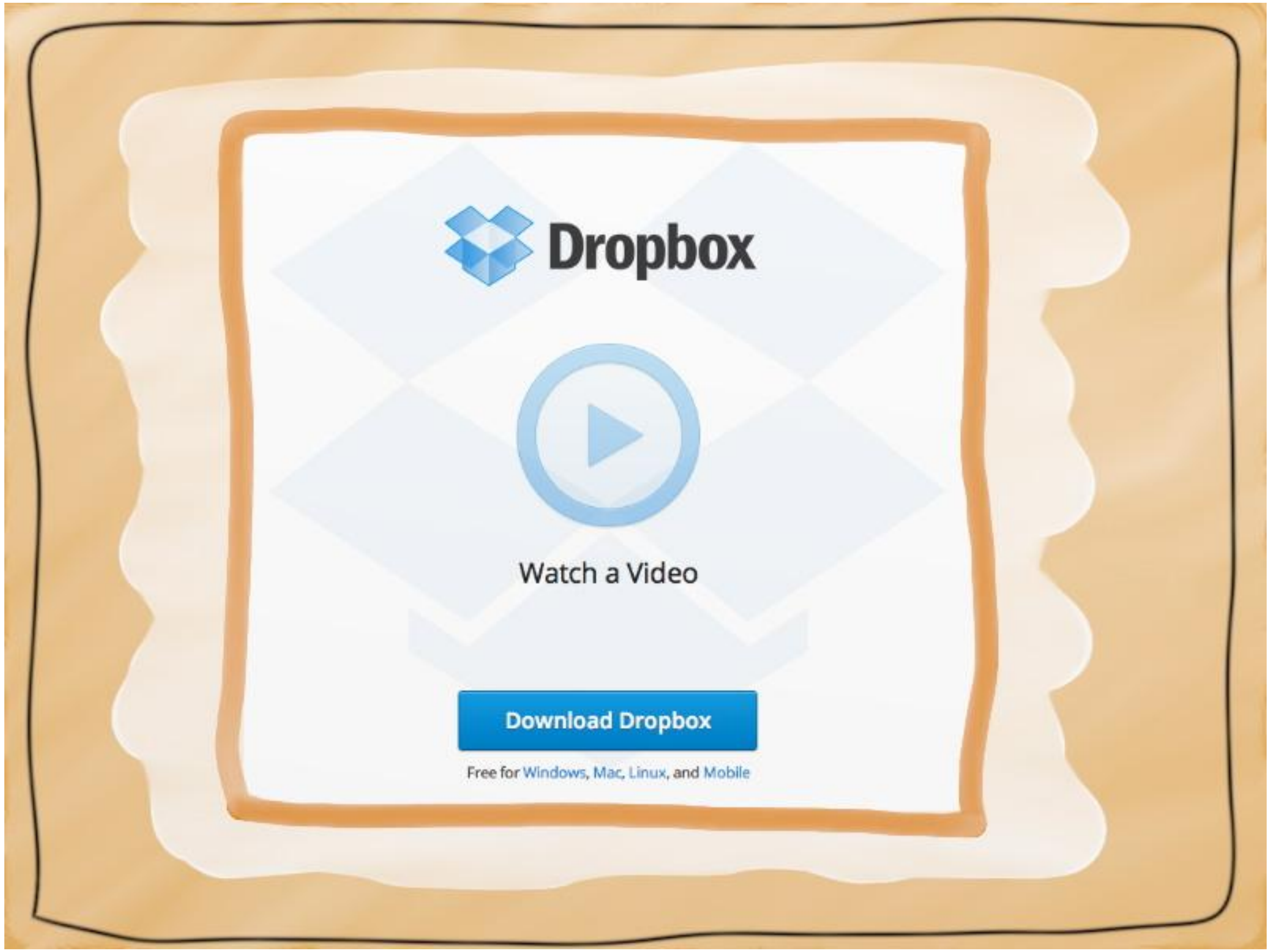
A list of all the agile principles

Yassal Sundman

Simplicity

The tenth principle of the agile manifesto is:

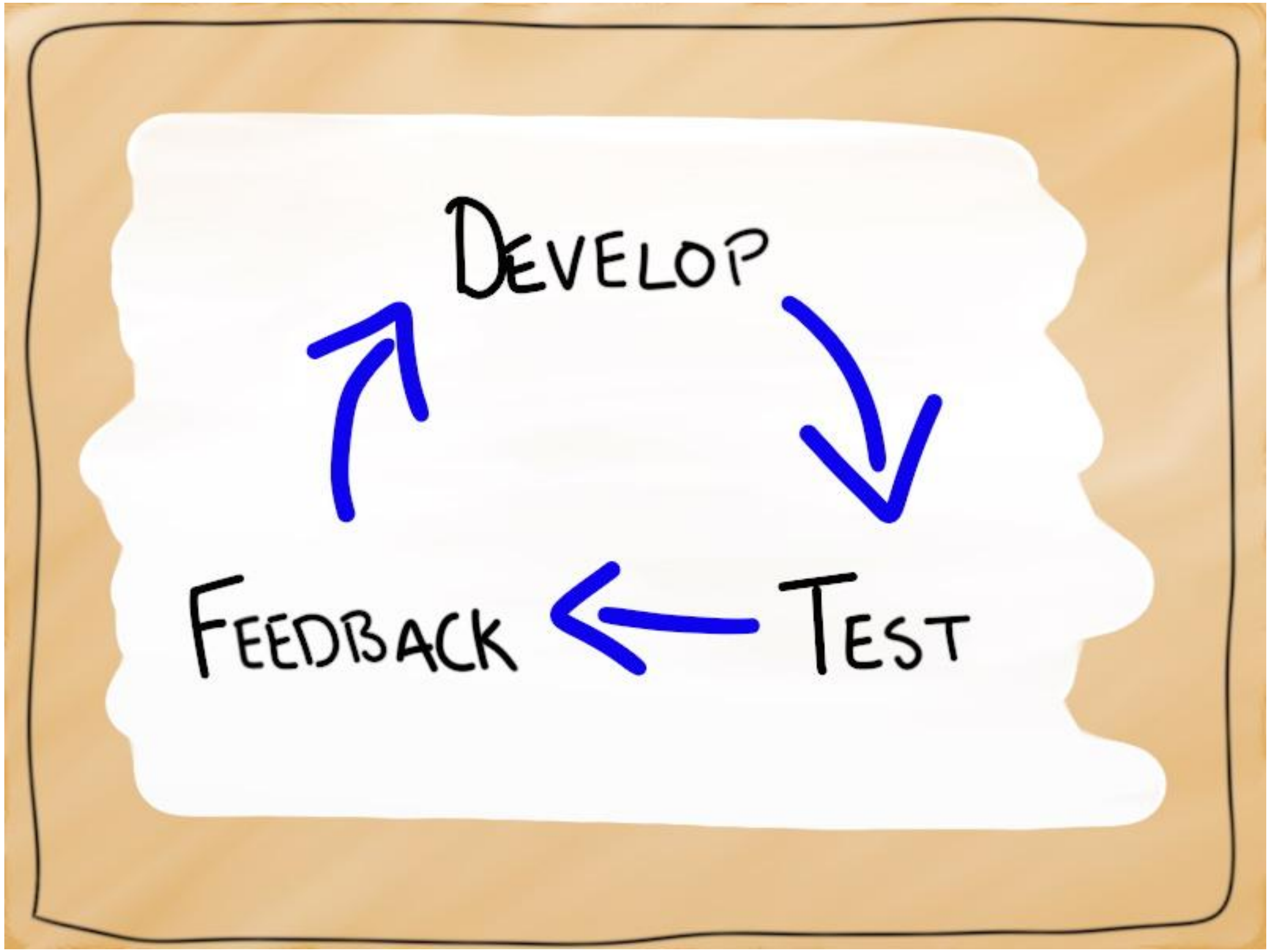
Simplicity- The art of maximizing the amount of work not done - is essential



Yassal Sundman

Iterate

Short cycles that deliver value to the customer, and give feedback to developers.



Quality

Test early, test often. Delivering fast means integrating frequently, problems are found earlier and fixed faster.

Embrace Change

The second principle of the agile manifesto is:

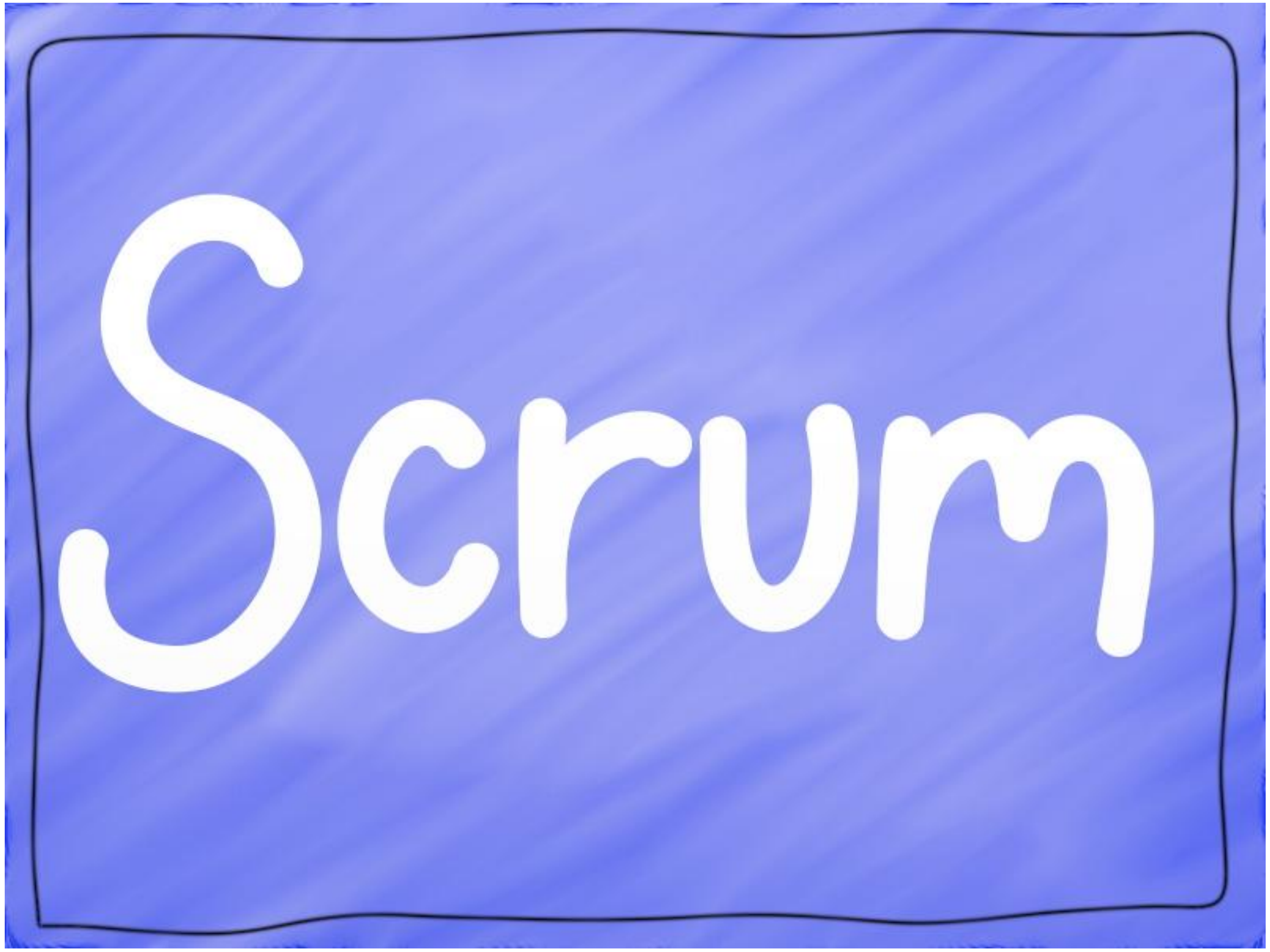
Welcome changing requirements, even late in development

crisp

Yassal Sundman

Scrum

What is Scrum, and how does it address the shortcomings of traditional software development?



crisp

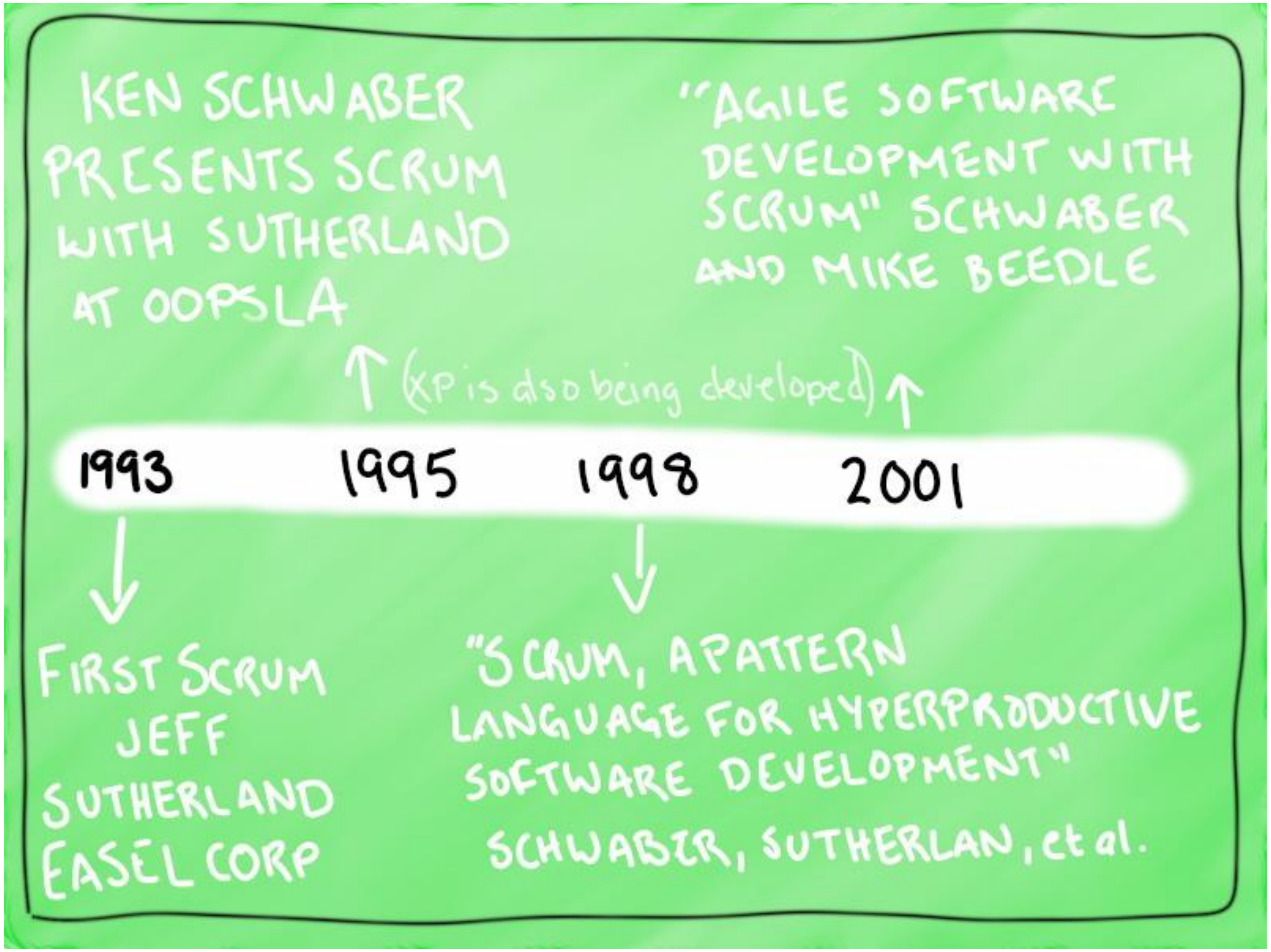
Yassal Sundman

History of Scrum

From 1993 - 1998 two papers are written on Scrum. In 2001 the first book is published.

XP is being developed in the mid 1990s by Kent Beck who collaborates with Jeff Sutherland.

Jeff Sutherland's Scrum Papers provides detailed information about Scrum.

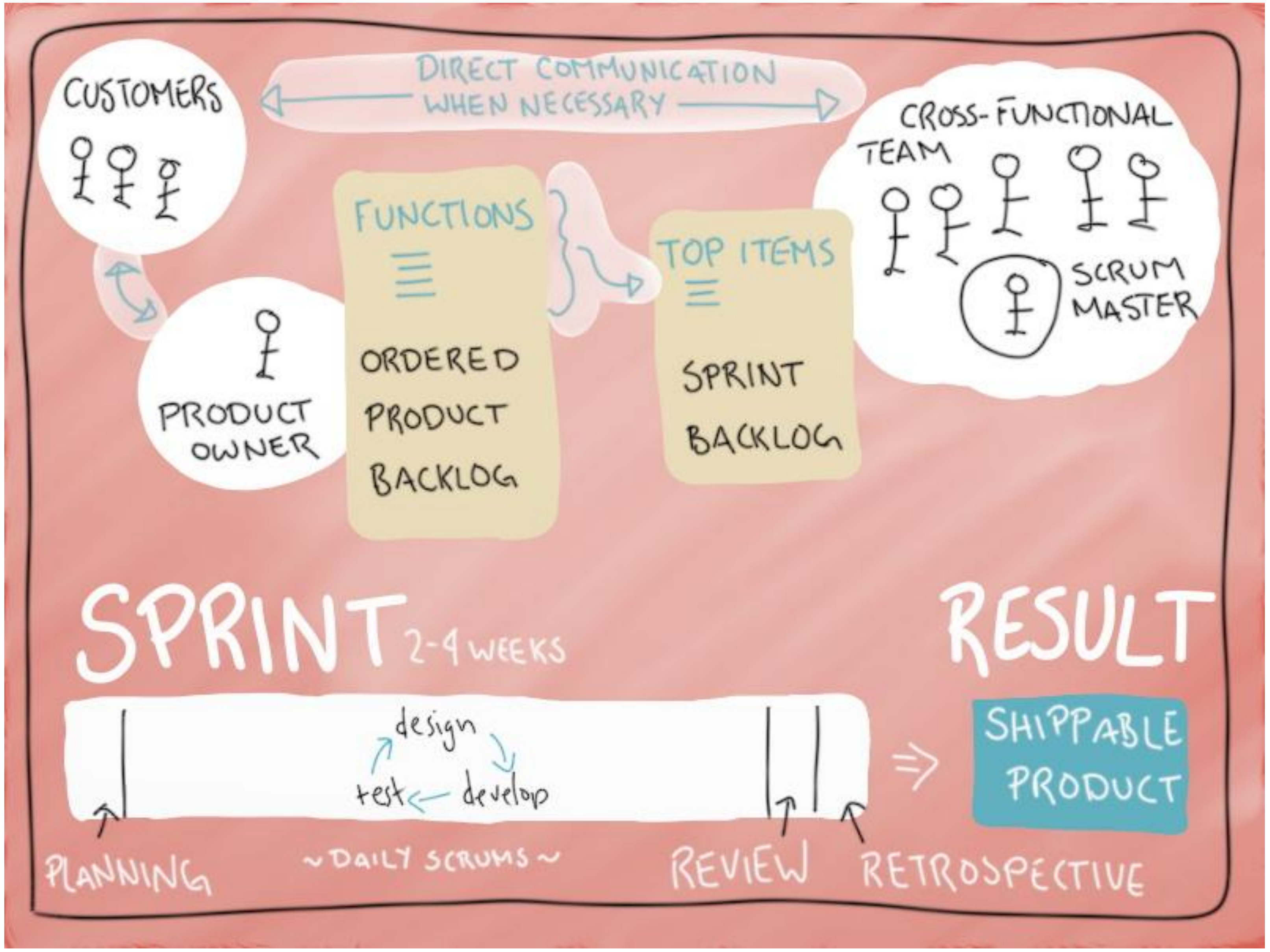

KEN SCHWABER PRESENTS SCRUM WITH SUTHERLAND AT OOPSLA
"AGILE SOFTWARE DEVELOPMENT WITH SCRUM" SCHWABER AND MIKE BEEDLE
(XP is also being developed)
1993
1995
1998
2001
FIRST SCRUM JEFF SUTHERLAND EASEL CORP
"SCRUM, A PATTERN LANGUAGE FOR HYPERPRODUCTIVE SOFTWARE DEVELOPMENT" SCHWABER, SUTHERLAN, et al.

Yassal Sundman

Scrum Is:

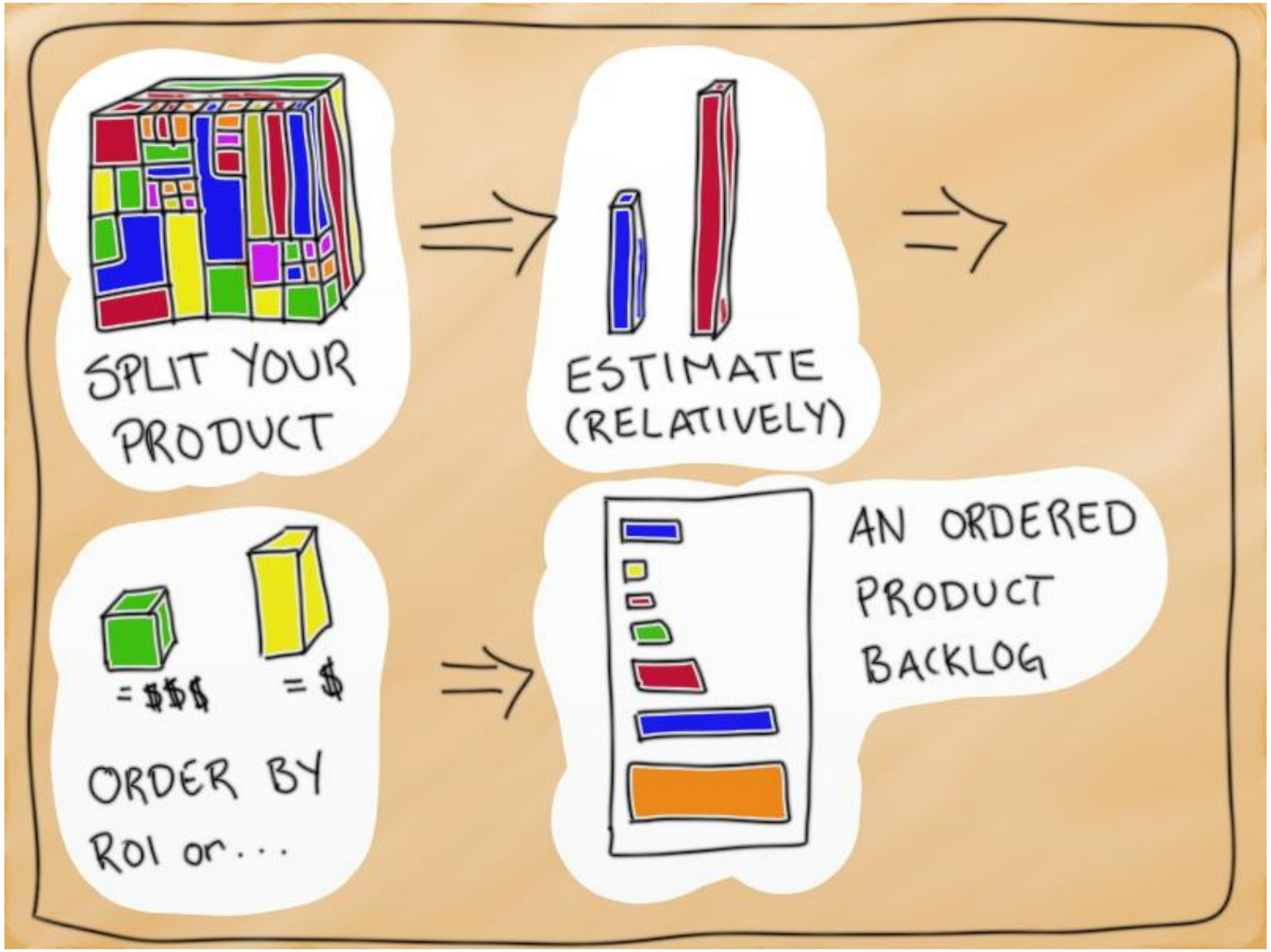
From the Scrum Alliance

- A product owner creates a product backlog.
- The team creates the sprint backlog during the sprint planning
- 2-4 week sprint with daily scrums, that produces a potentially shippable product
- The sprint ends with a sprint review and retrospective.



Creating a Backlog

The product owner breaks down the product with the team's help. The team estimates functional items, relative to each other. The value of each item is calculated. A backlog is created. More information is known, and the items are more detailed higher on the list.

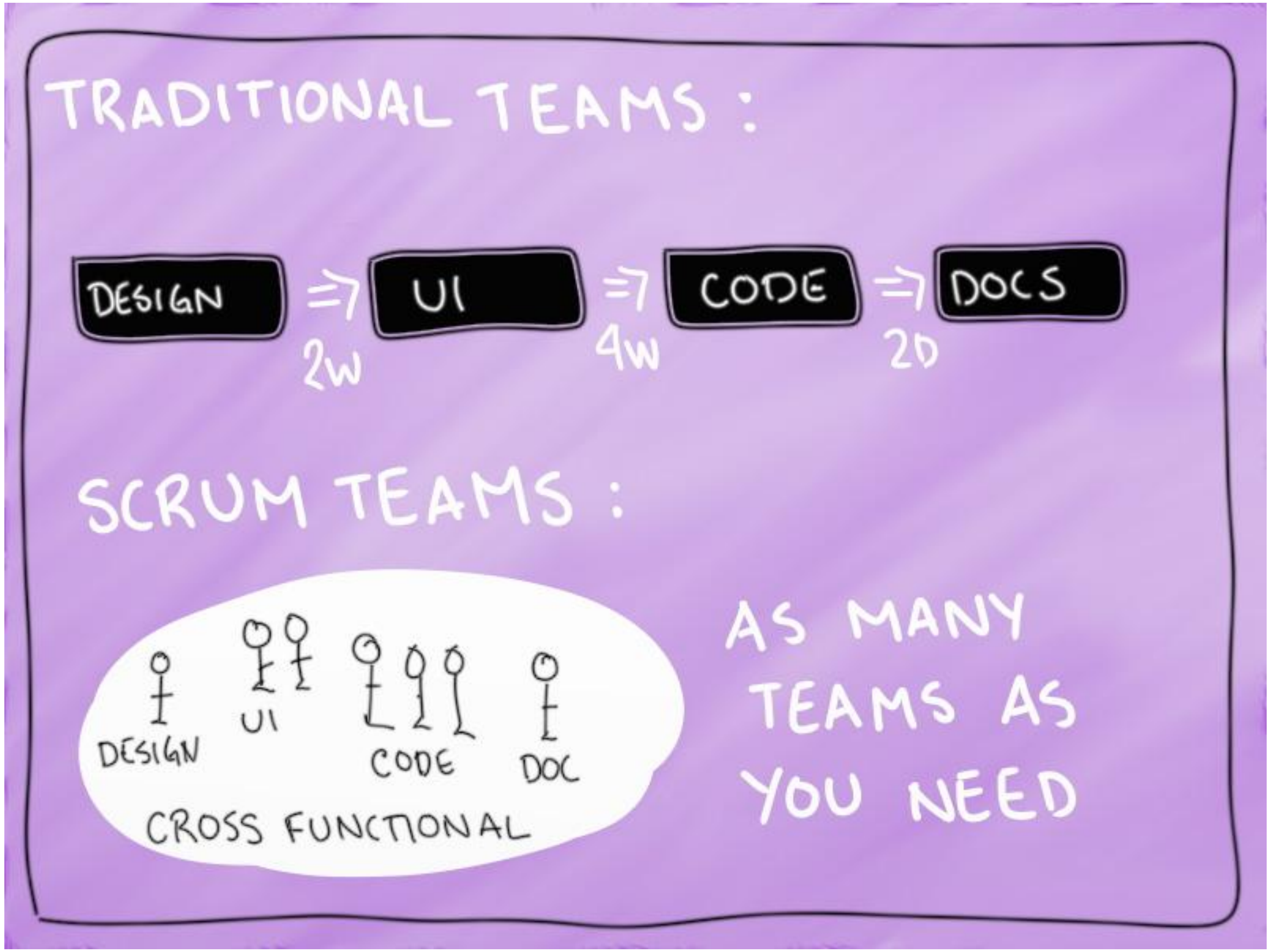


crisp

Yassal Sundman

Scrum Team

Each team has all of the areas of expertise it needs to be able to deliver. The team members each have a primary skill, but they also have overlapping skills. The goal is to share knowledge so that anybody in the team can help out with any task.

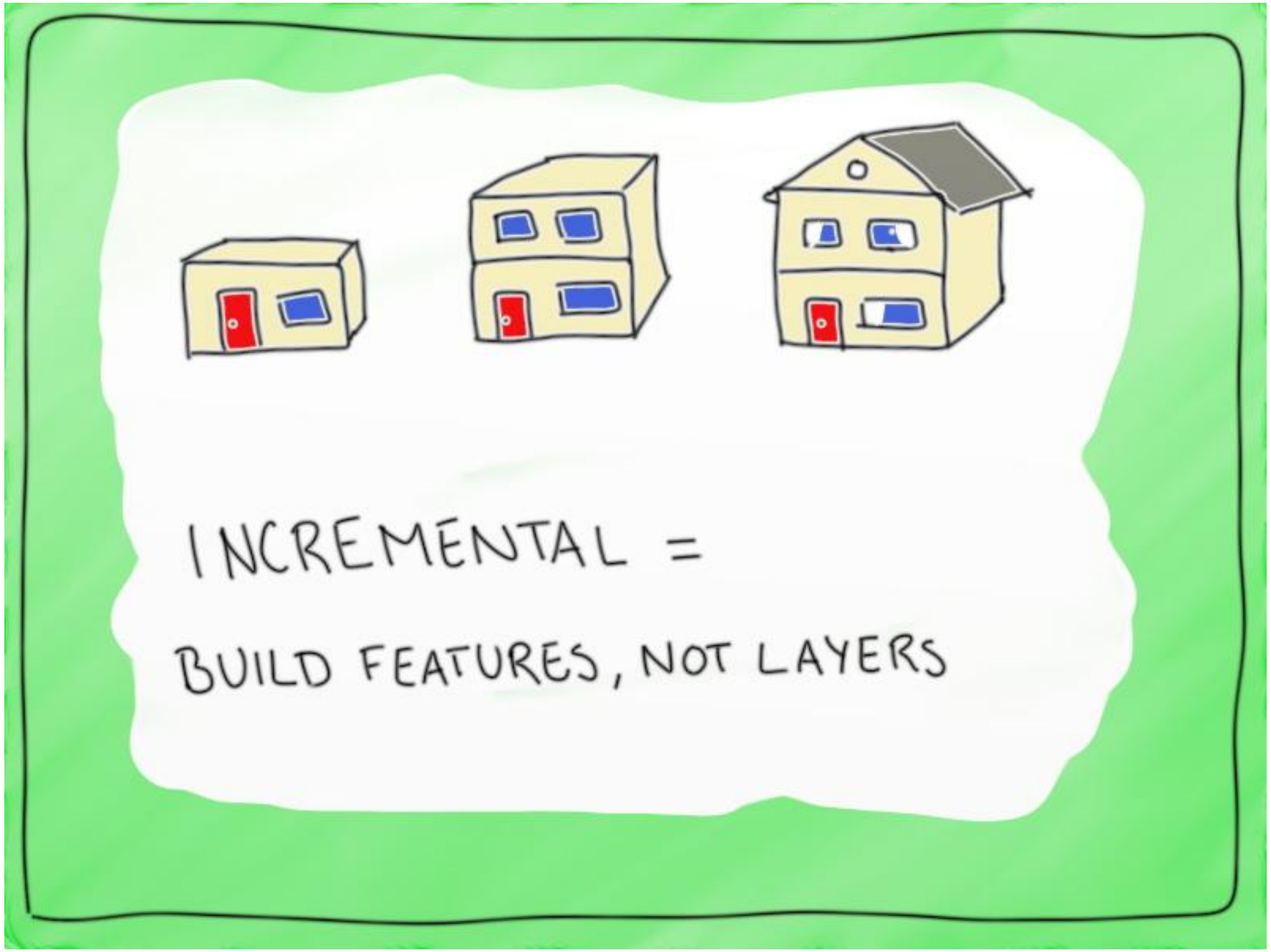


crisp

Yassal Sundman

Incremental, Iterative, Development

Don't expect to get the product right the first time. Deliver a working product to get feedback, and keep adding features.

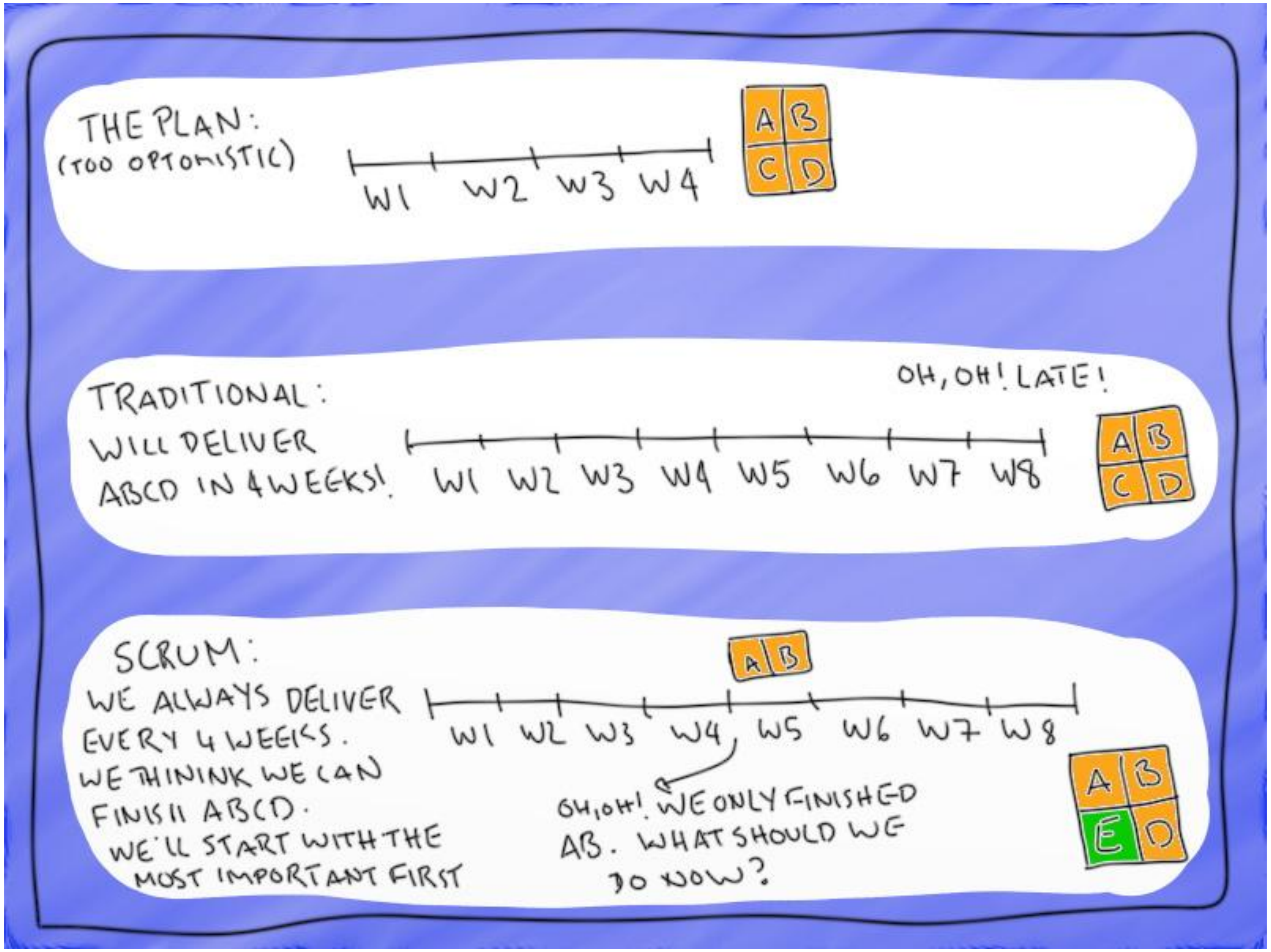


crisp

Yassal Sundman

Timeboxing

Sprint lengths are not variable. The goal of each sprint is a potentially shippable product. The delivery date is always known.

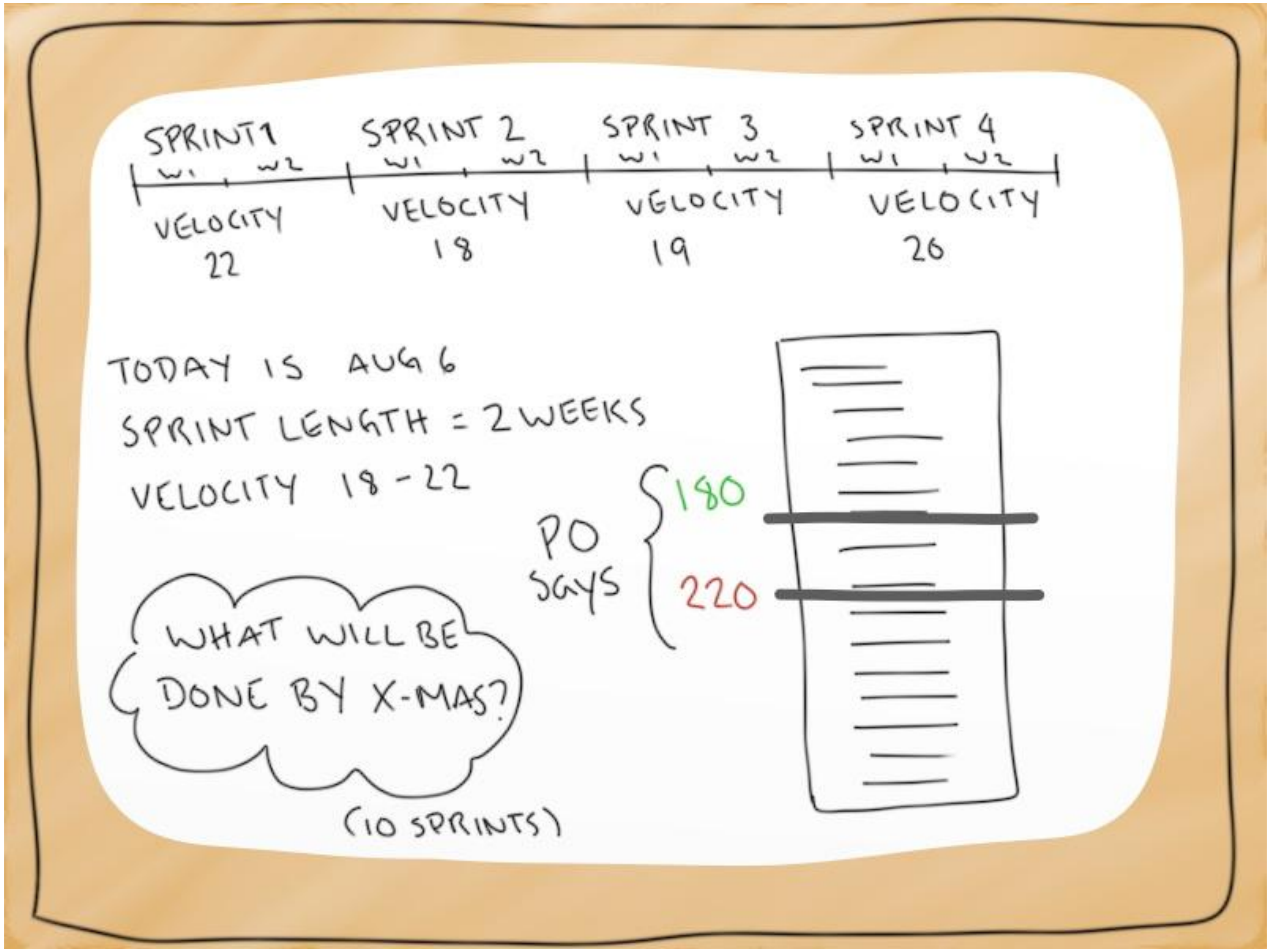

THE PLAN:
(TOO OPTOMISTIC)
W1 W2 W3 W4
A B C D
TRADITIONAL:
WILL DELIVER ABCD IN 4 WEEKS!
W1 W2 W3 W4 W5 W6 W7 W8
OH, OH! LATE!
A B C D
SCRUM:
WE ALWAYS DELIVER EVERY 4 WEEKS.
WE THINK WE CAN FINISH ABCD.
WE'LL START WITH THE MOST IMPORTANT FIRST
W1 W2 W3 W4 W5 W6 W7 W8
A B
OH, OH! WE ONLY FINISHED AB. WHAT SHOULD WE DO NOW?
A B E D

crisp

Yassal Sundman

Release Planning

The team's velocity is calculated based on past performance, and a release plan is communicated based on the velocity. Changes in the velocity alert early that the plan will be delivered with more functionality, or will need to cut functionality.

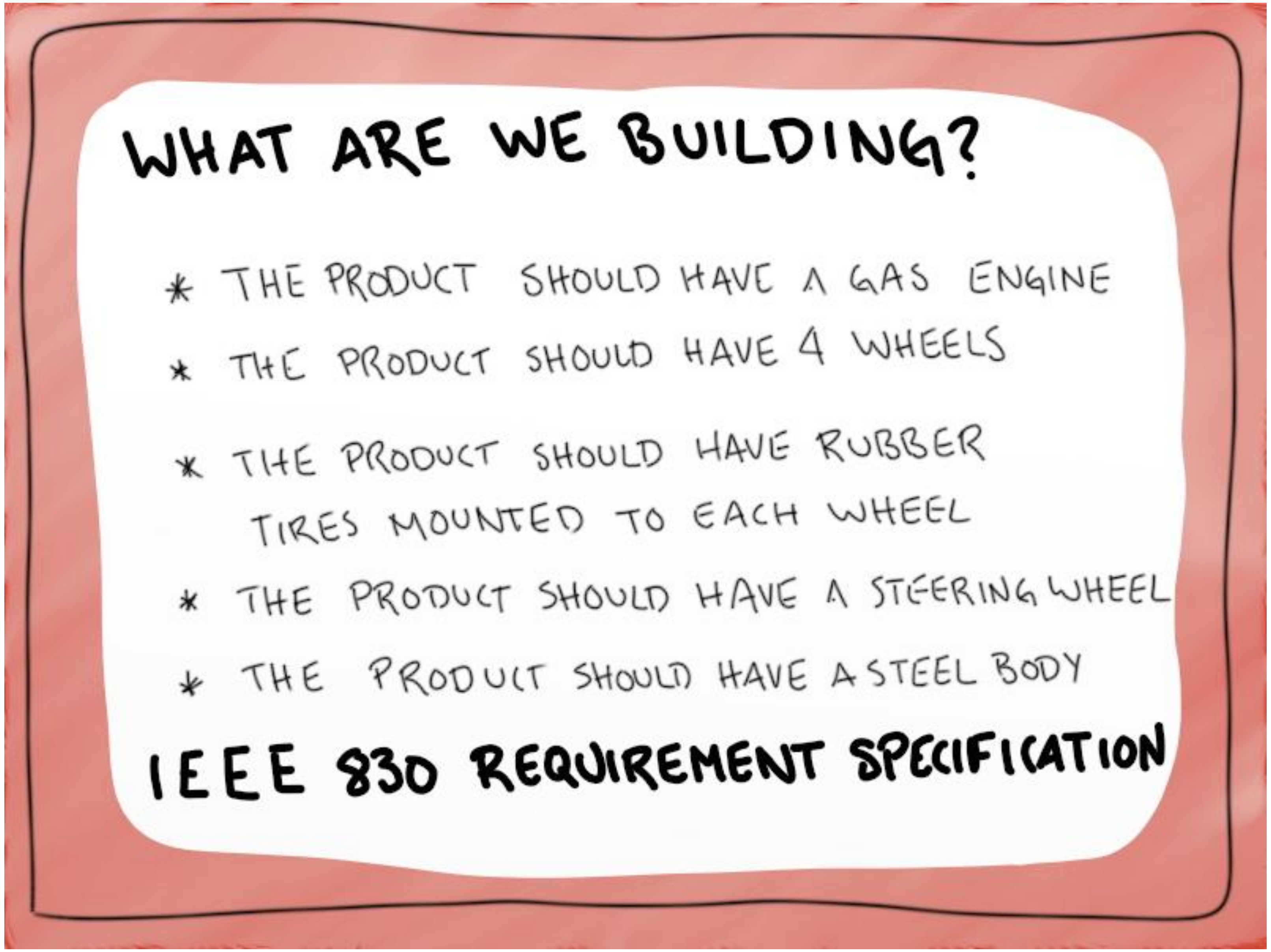


crisp

Yassal Sundman

Requirements

What are we building?



WHAT ARE WE BUILDING?

* THE PRODUCT SHOULD HAVE A GAS ENGINE
* THE PRODUCT SHOULD HAVE 4 WHEELS
* THE PRODUCT SHOULD HAVE RUBBER TIRES MOUNTED TO EACH WHEEL
* THE PRODUCT SHOULD HAVE A STEERING WHEEL
* THE PRODUCT SHOULD HAVE A STEEL BODY

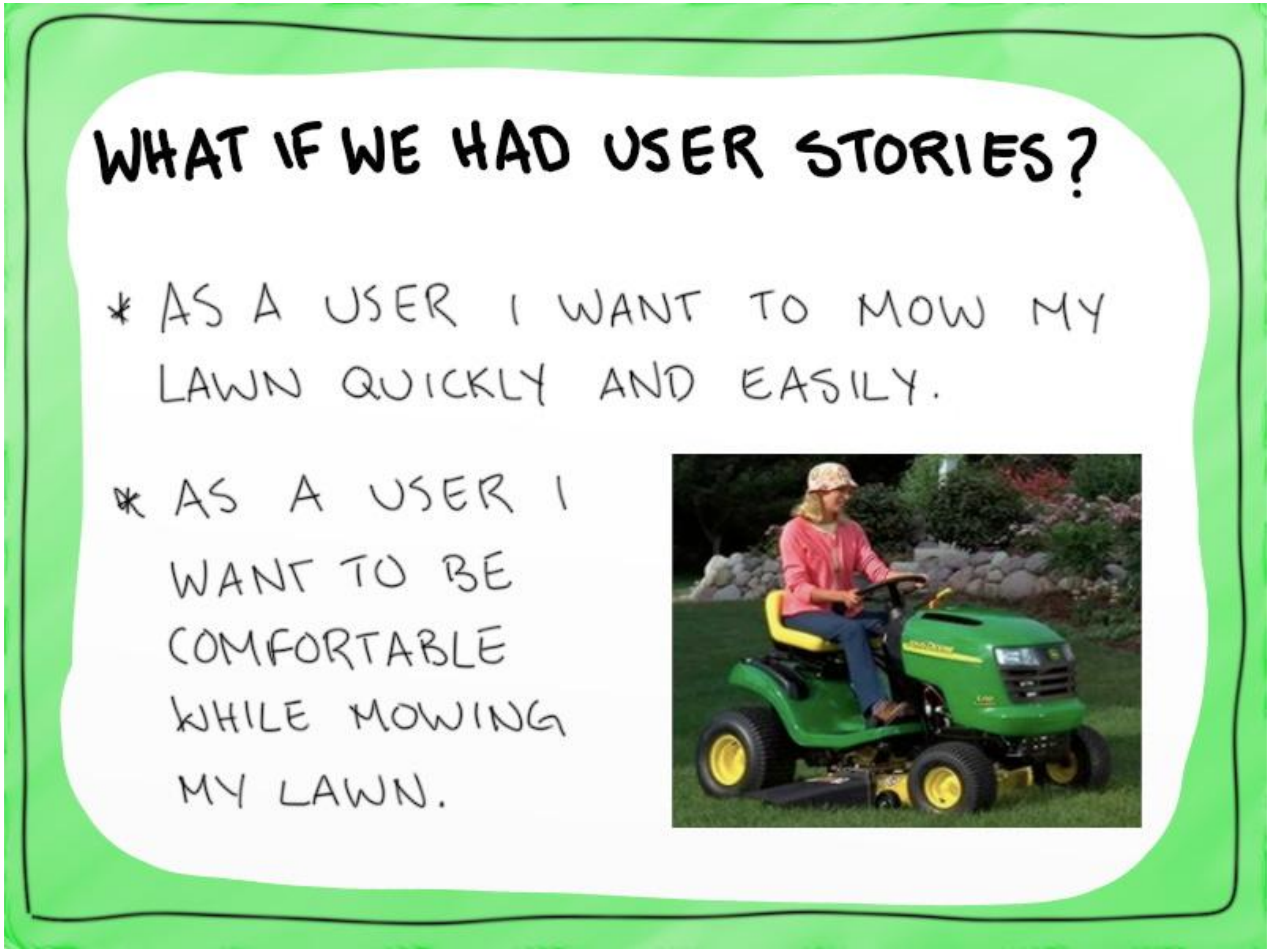
IEEE 830 REQUIREMENT SPECIFICATION

crisp

Yassal Sundman

User Stories

Myth: If you write down the requirements the users get what they want!



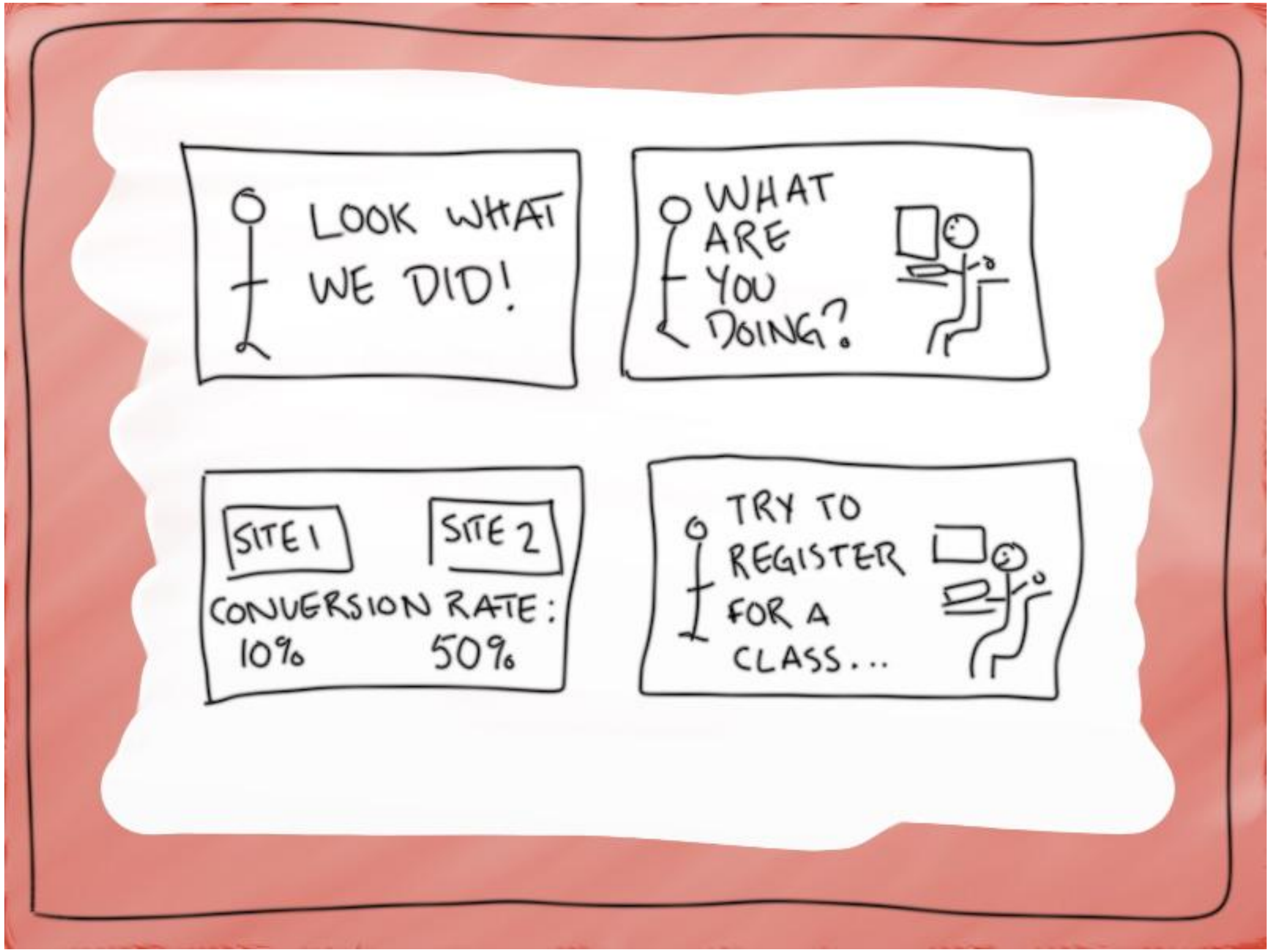
WHAT IF WE HAD USER STORIES?

* AS A USER I WANT TO MOW MY LAWN QUICKLY AND EASILY.
* AS A USER I WANT TO BE COMFORTABLE WHILE MOWING MY LAWN.

User Feedback

It's important to get feedback throughout the development process.

- Demo for real users.
- Follow users at their jobs
- A/B testing
- User testing



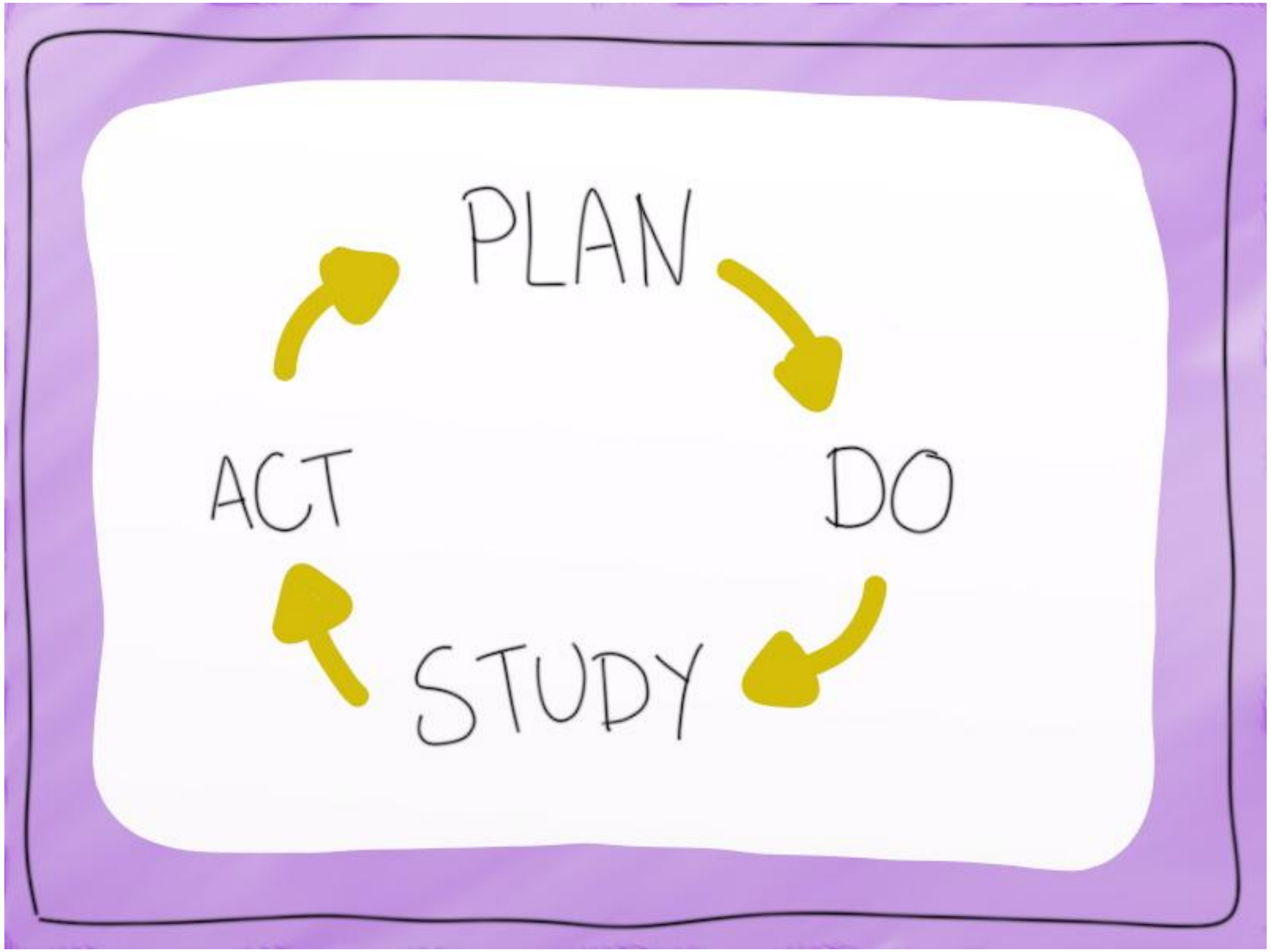
Yassal Sundman

Lean and Kanban

Yassal Sundman

Deming

In the 1950s W. Edwards Deming goes to Japan and works with product development and manufacturing. He's interested in improving processes. His focus is on quality.



crisp

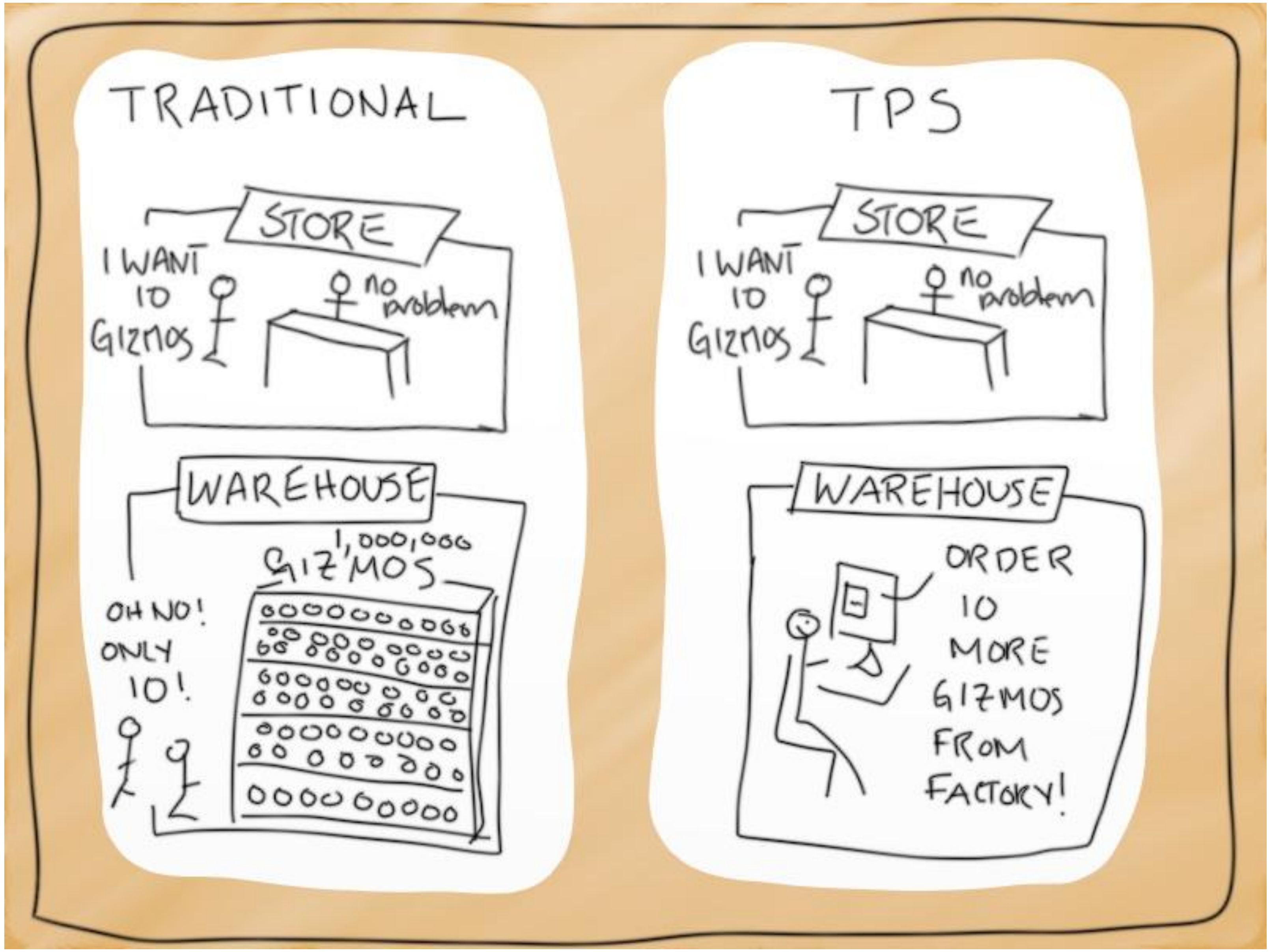
Yassal Sundman

Toyota Production System

Spearheaded by Taiichi Ohno in the late 1950's. The tenets are:

- Just in time: make the right thing at the right time and the right amount
- Jidoka - build quality in to the process
- Kaizen - continuous improvement

The goal is to increase income by eliminating waste. Provide the highest quality product at the lowest possible cost.



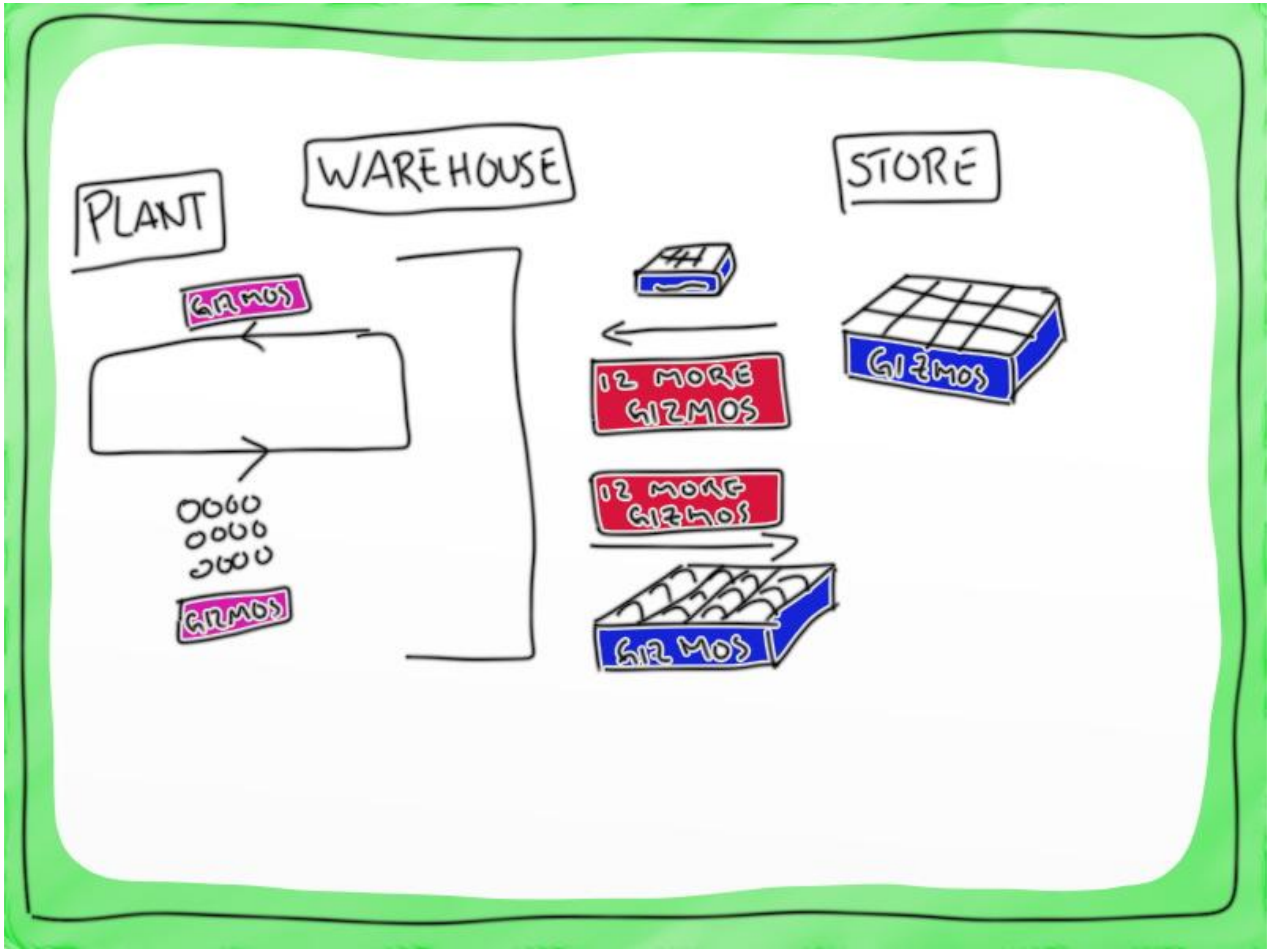
crisp

Yassal Sundman

Kanban in Product Development

Six rules of Kanban:

- Customer(Downstream) processes withdraw items in the precise amounts specified on the Kanban.
- Supplier(Upstream) produces items in the precise amounts and sequences specified by the Kanban.
- No items are made or moved without a Kanban.
- A Kanban should accompany each item, every time.
- Defects and incorrect amounts are never sent to the next downstream process.
- The number of Kanbans is reduced carefully to lower inventories and to reveal problems.

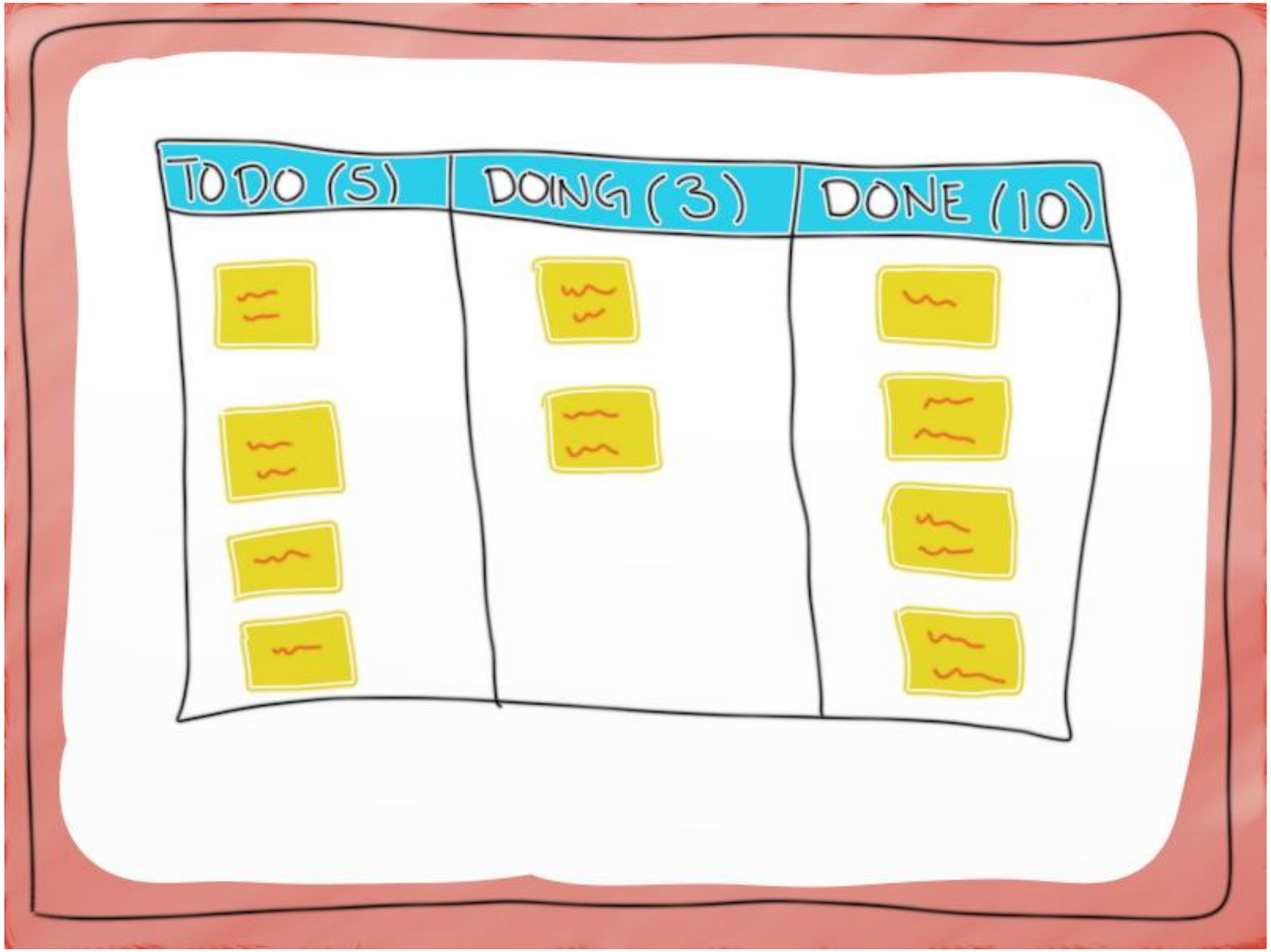


crisp

Yassal Sundman

Kanban in Software Development

Visualize and limit work in progress.

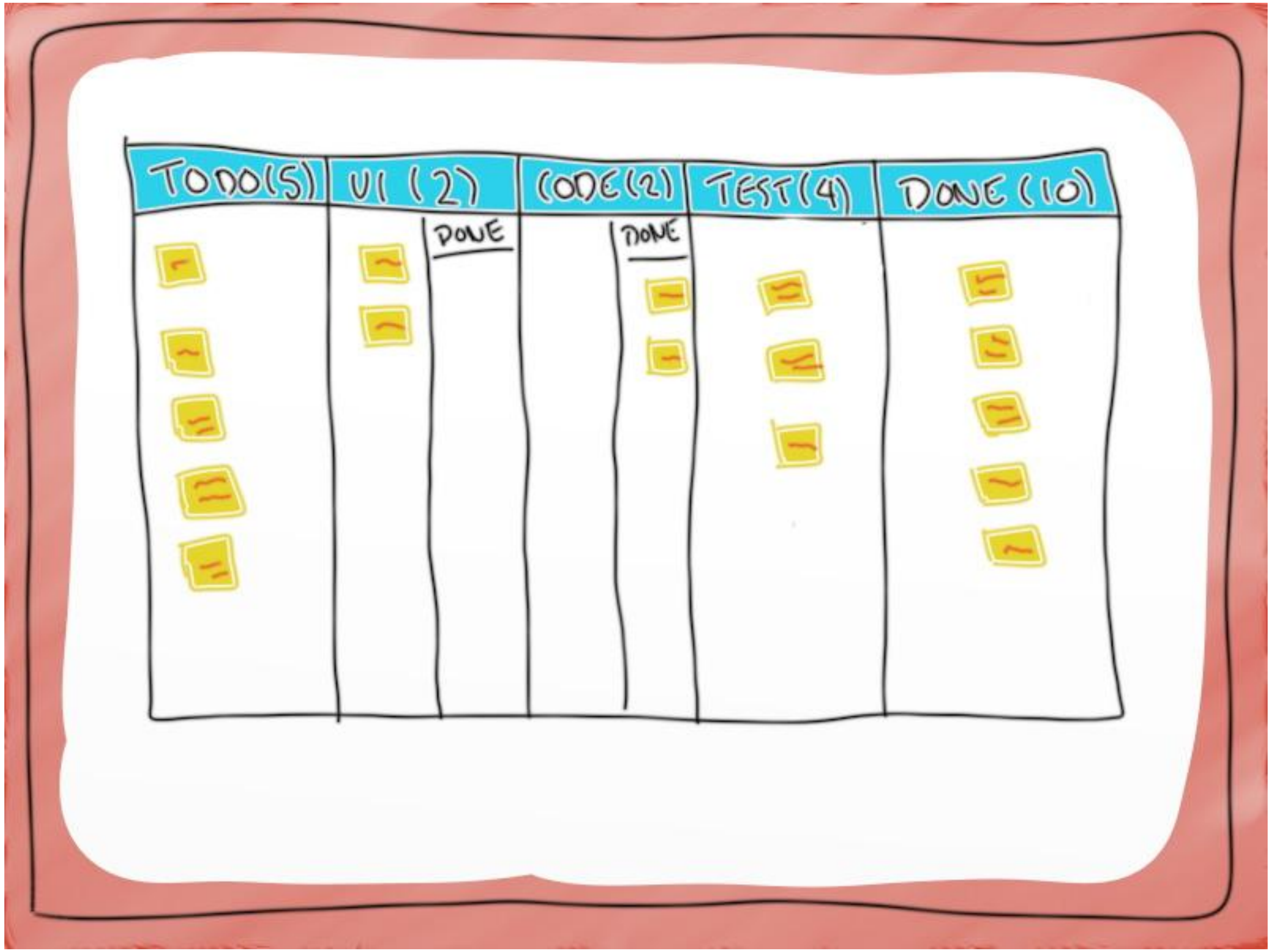


crisp

Yassal Sundman

Kanban Pull

The downstream job has to pull from the upstream.



crisp

Yassal Sundman

Kanban Practices

Six core practices:

- Visualise
- Limit WIP
- Manage Flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

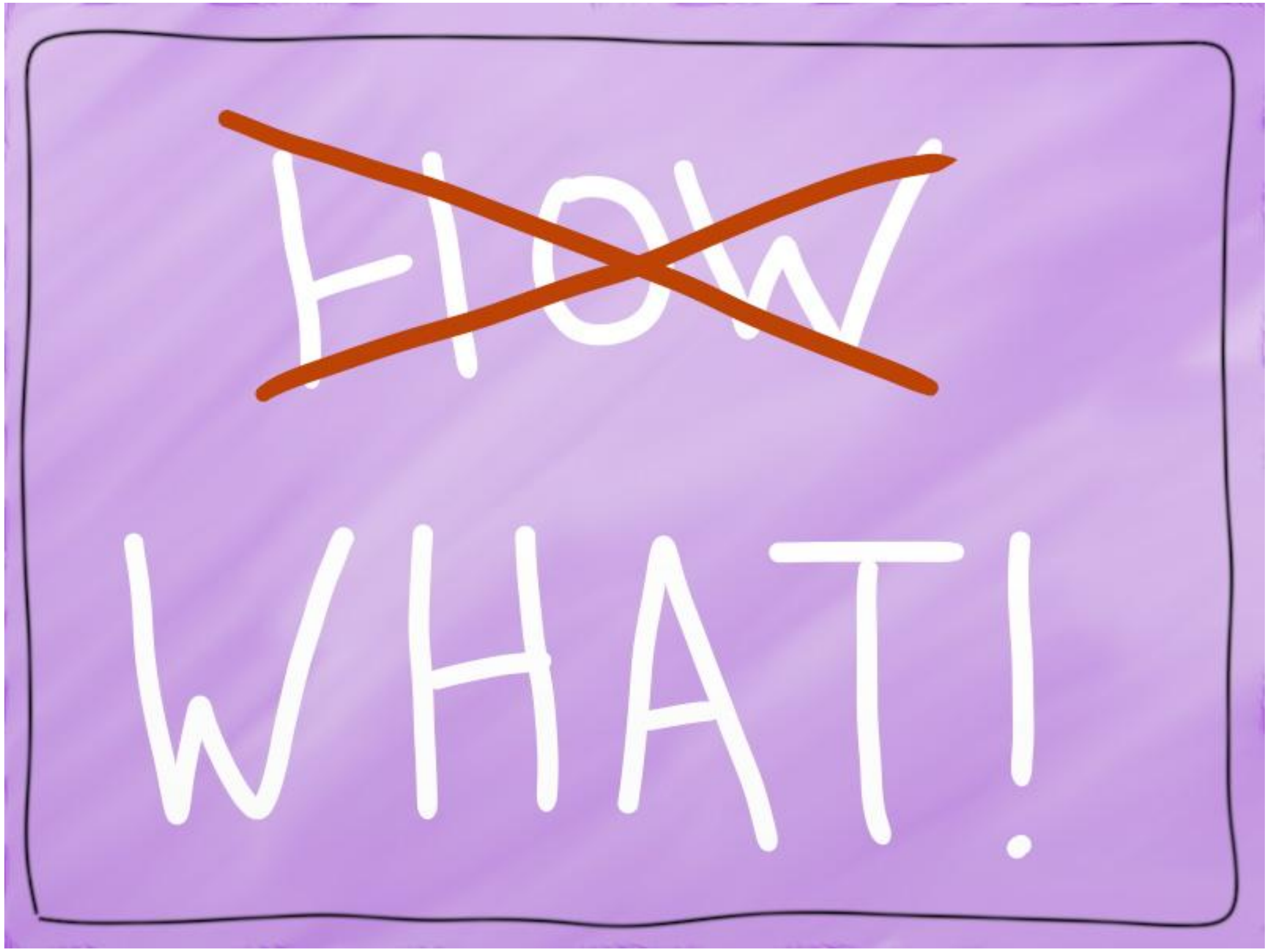
HOW LONG DOES IT TAKE FOR A STORY TO GET TO DONE?
HOW CAN WE REDUCE THE TIME?
WHEN IS A STORY DONE?
HOW IS OUR WORK AFFECTING THE BOTTOM-LINE?

crisp

Yassal Sundman

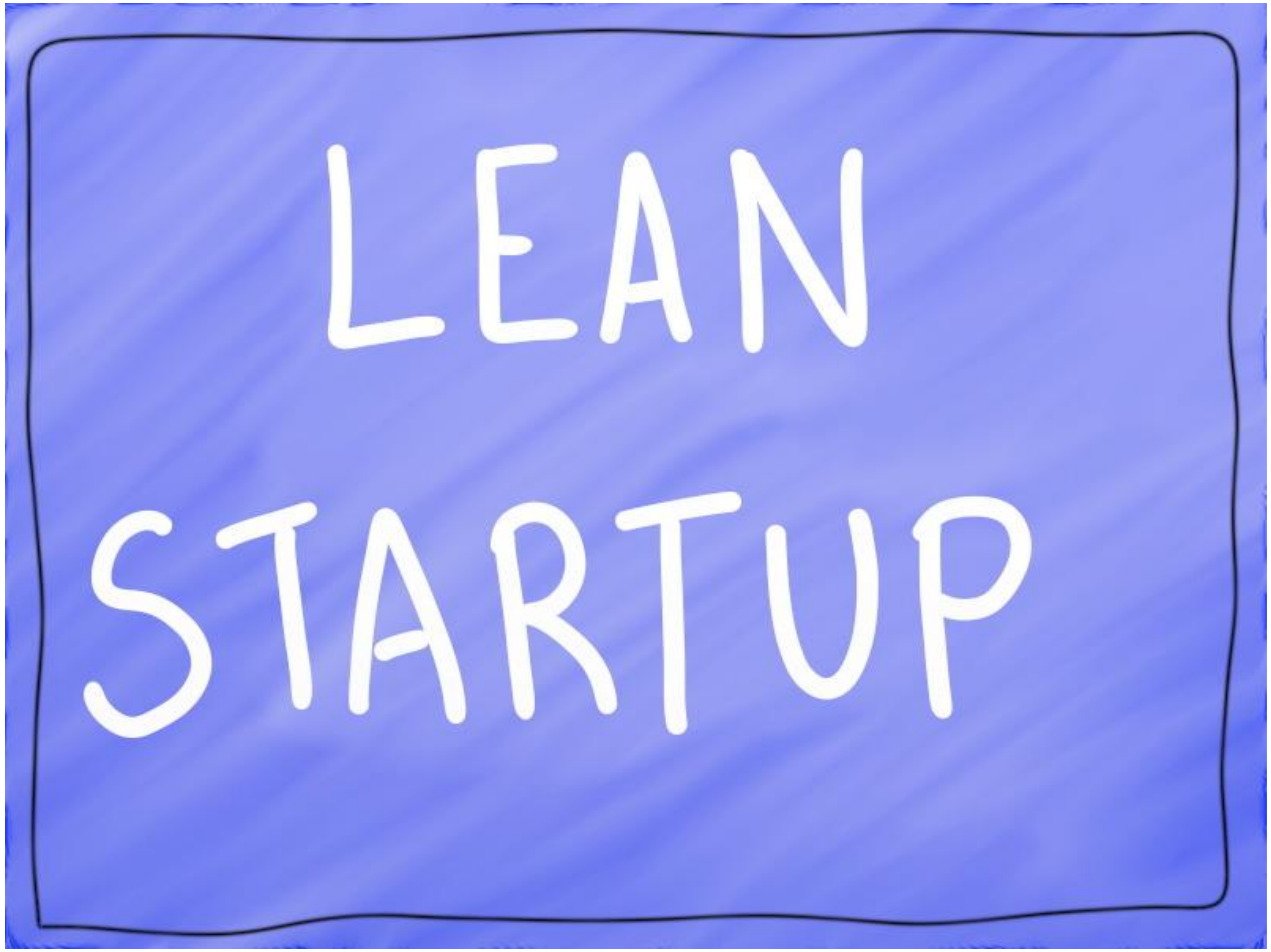
What Not How

Kanban in software development focuses on **what** the organization needs to do, not **how**. The how will be specific to each organization's needs and structure, and should be decided by the members.



Lean Startup

How does an agile startup company do software development? Lean Startup, developed by Eric Ries starting in 2008, is a business strategy to help startups get up and running in today's market.

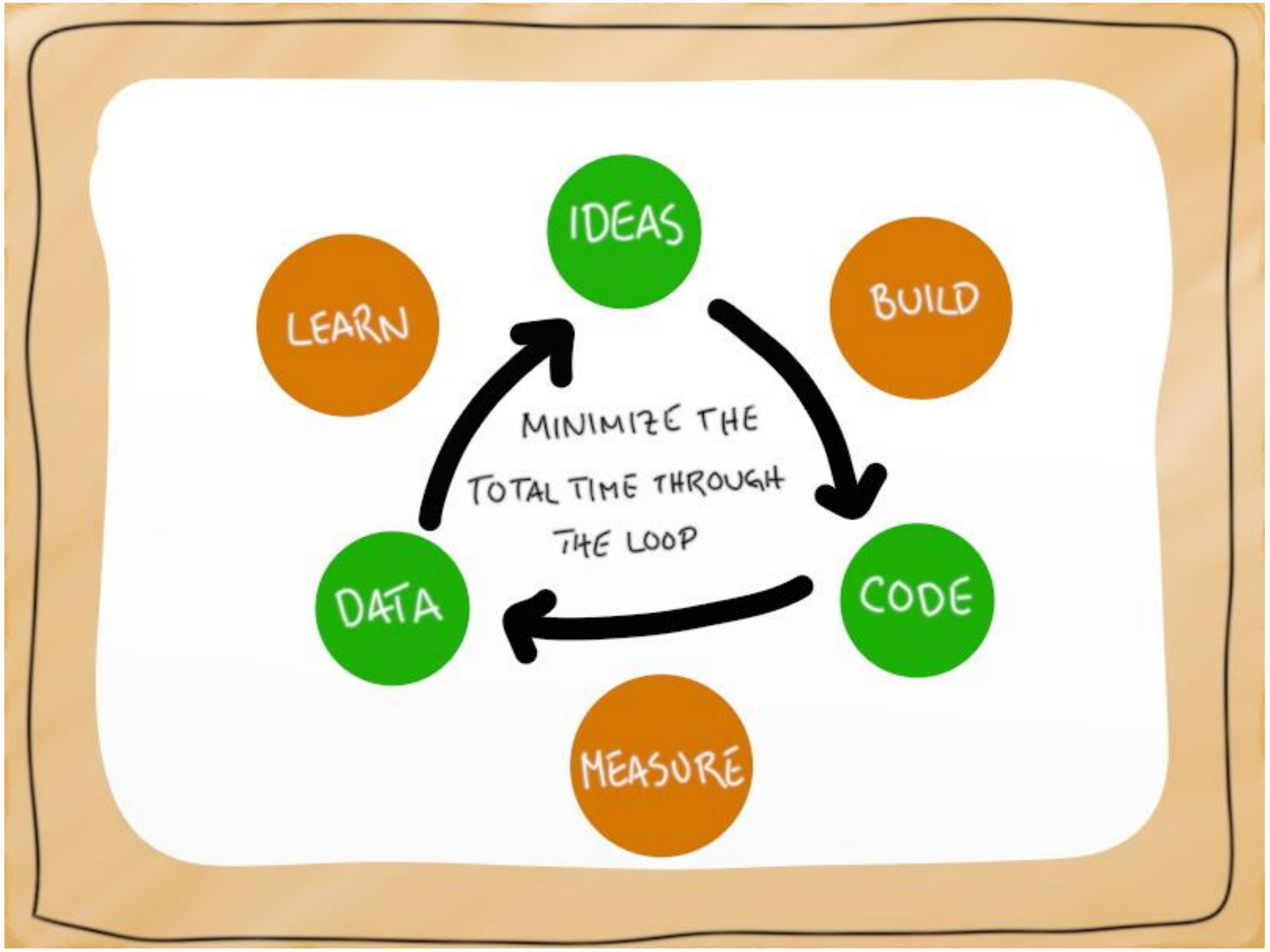


crisp

Yassal Sundman

Build Measure Learn

Starting to look familiar? The fifth principle is the one people think of when they hear Lean Startup for software teams: build - measure -learn. Figure out what your MVP is (minimum viable product), get your MVP out fast and measure your customers' reaction. Now iterate forward by changing your product to match what the customers want.

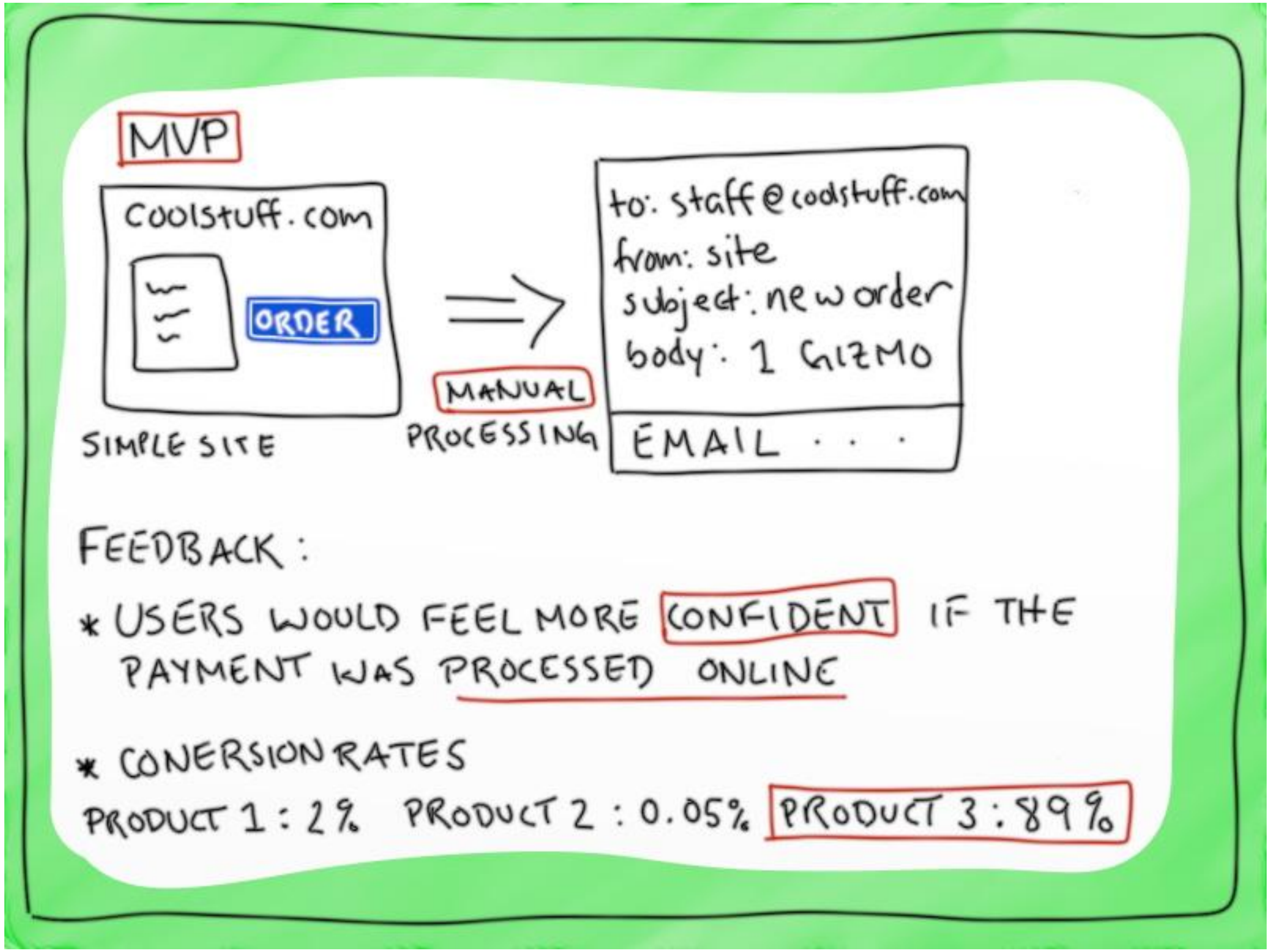


Example Lean Startup

Even for a tech venture you can start really simple. Manual steps the whole way to test out your idea. The goal is not having happy users, but rather high conversion rates. To have a successful startup you need:

- Continuous customer interaction
- Revenue goals from day one
- No scaling until revenue
- Assumes customer and features are unknowns
- Low burn by design - not crisis

Slides on Lean Startup.

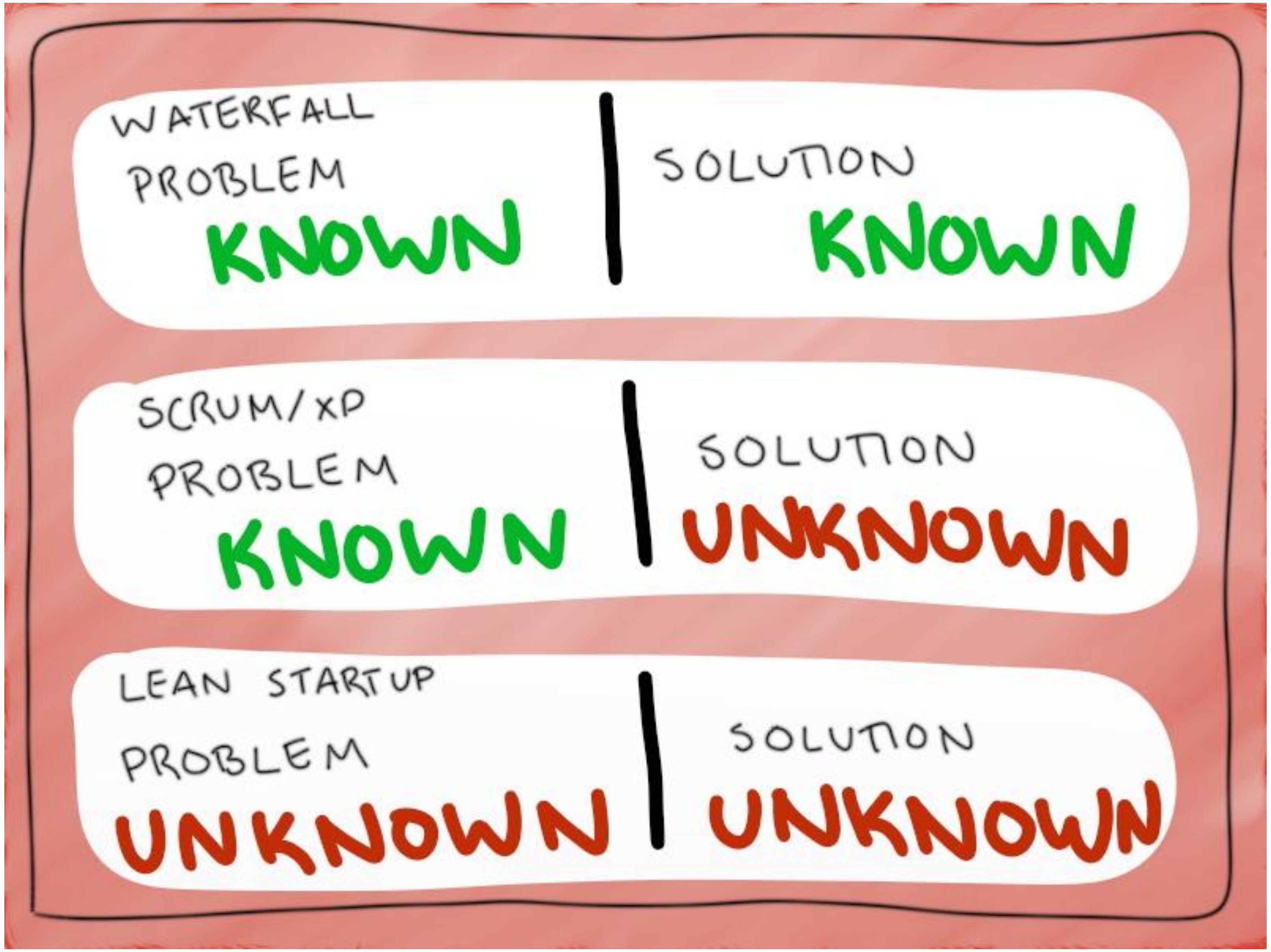


crisp

Yassal Sundman

Problem Unknown

Lean Startup explicitly states that the problem that the organization is trying to solve is unknown. You will discover what the market and your customers want as you build it.



crisp

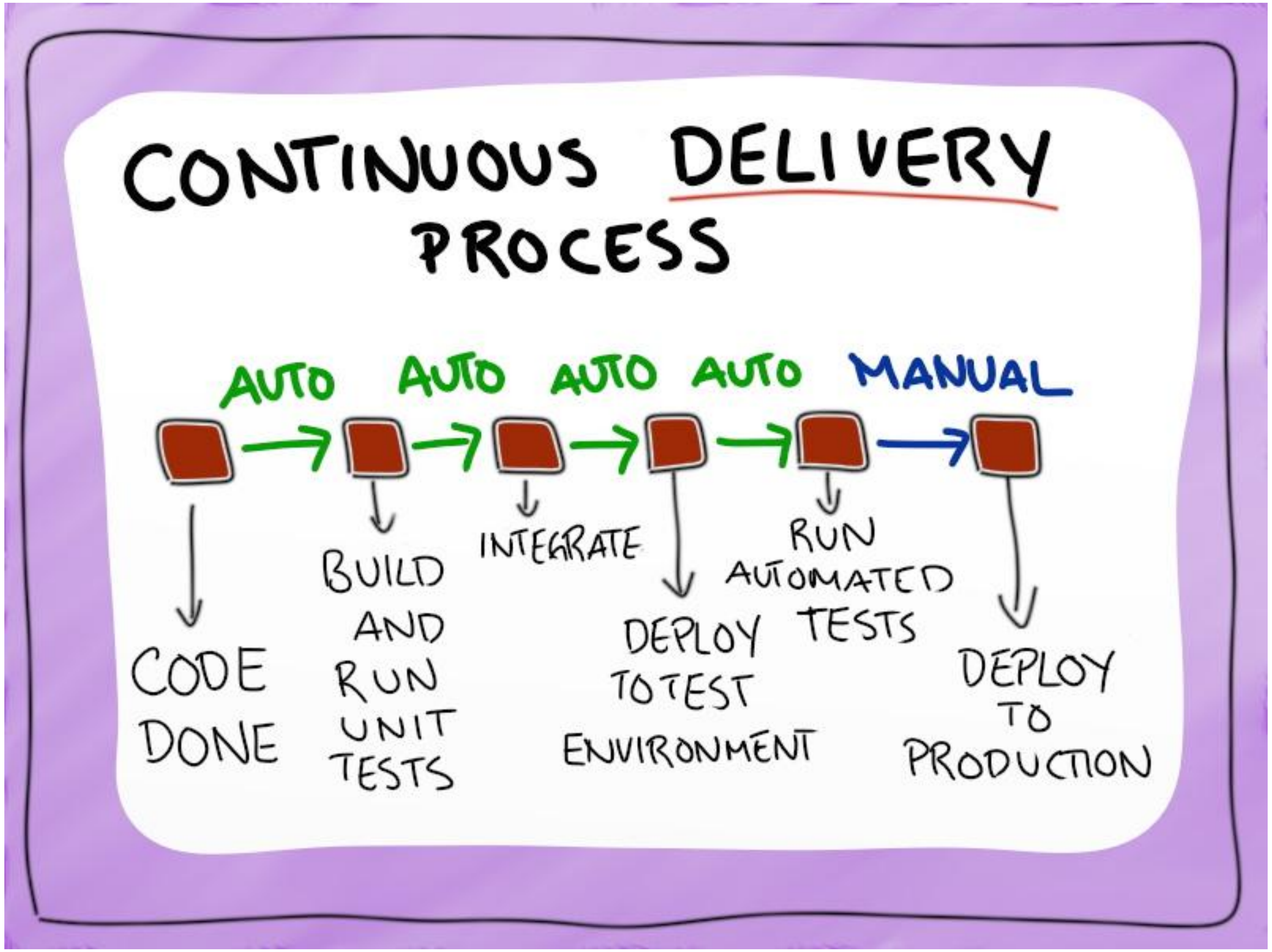
Yassal Sundman

Continuous Delivery

Deliver when you want to as often as you want to. Jez Humble's book Continuous Delivery came out in 2010, and there's a great blog entry describing continuous delivery.

Continuous Delivery Pipeline

Integrate automatically, deploy to test environments automatically, test automatically. Deploy to production **on demand**.

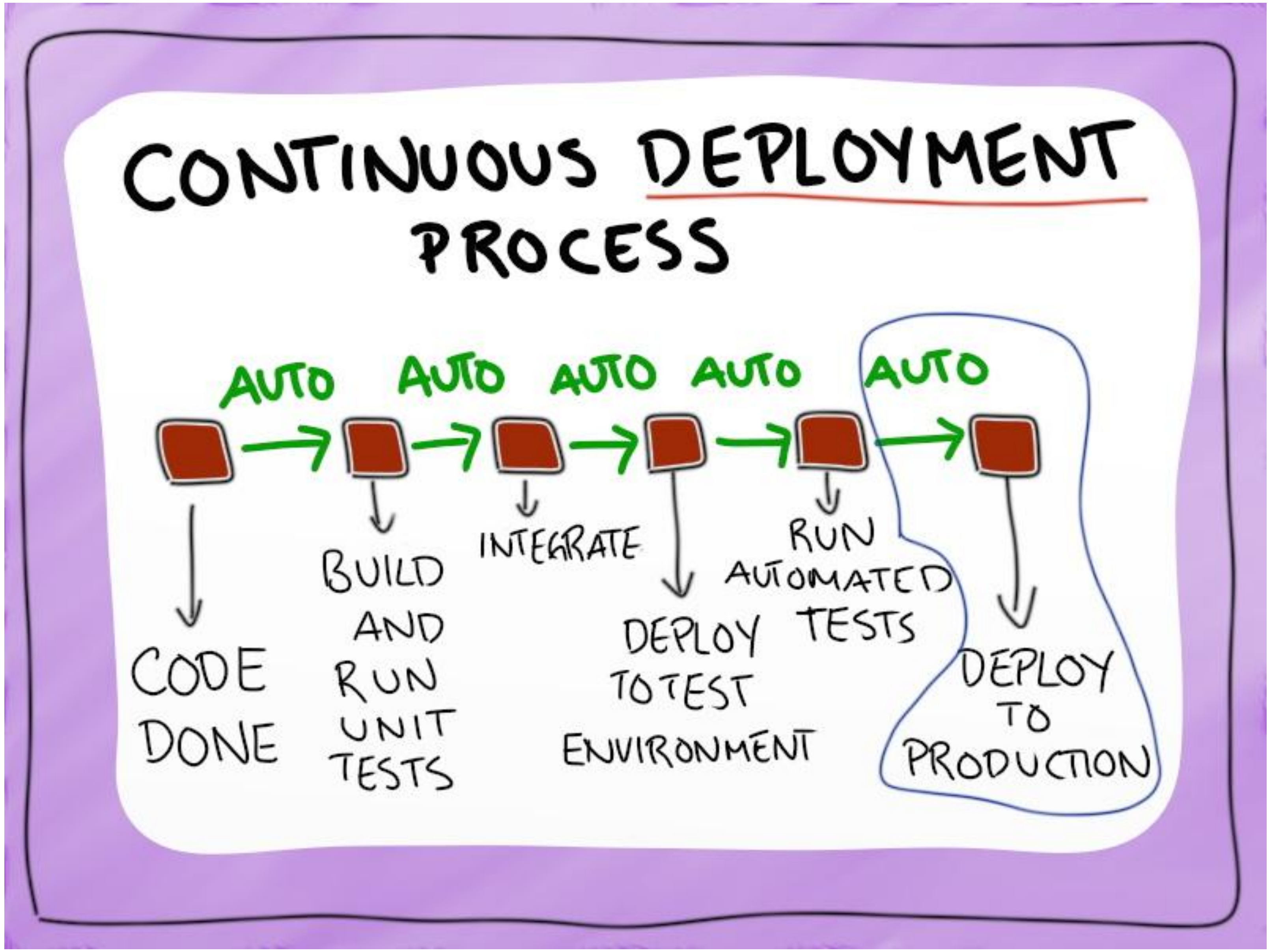


crisp

Yassal Sundman

Continuous Deployment Pipeline

Integrate automatically, deploy to test environments automatically, test automatically. Deploy to production **automatically**.



crisp

Yassal Sundman

Thank You!

Thank you for having me here! And a huge thanks to all the Crisp consultants whose material I've used in making these slides: Hans Brattberg,Thomas Björkholm and Henrik Kniberg.