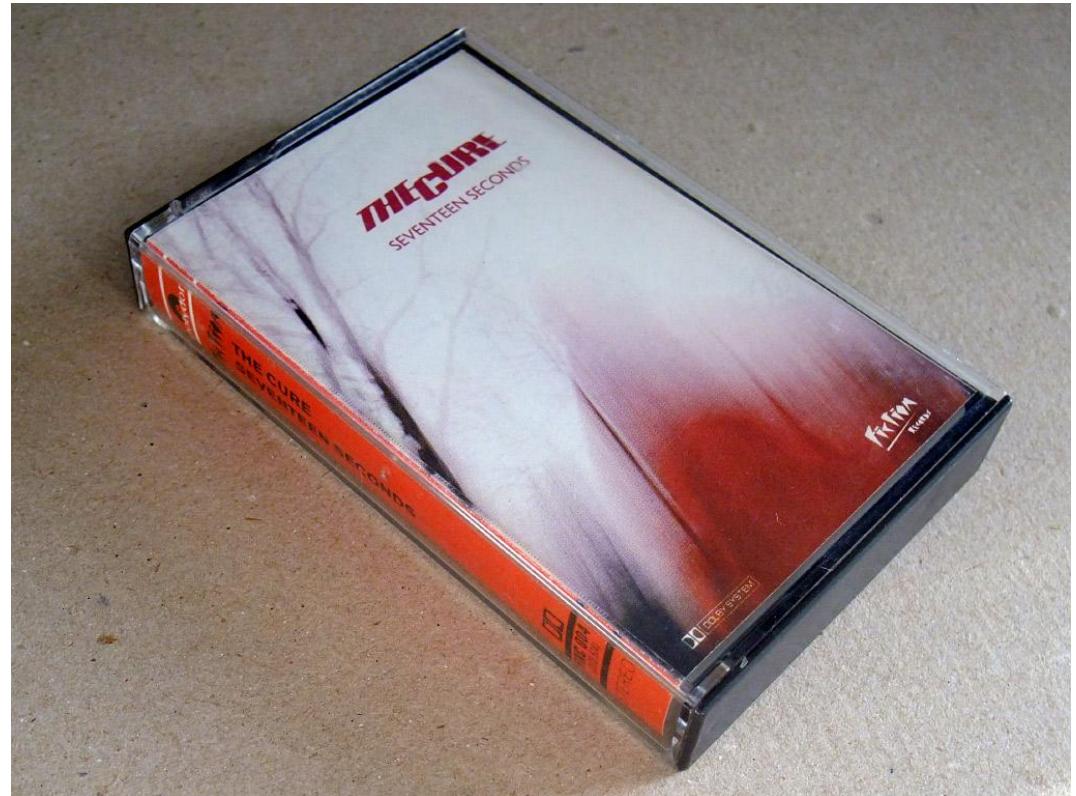


# DAT255 / DIT543 SOFTWARE ENGINEERING PROJECT



A photograph of a man with dark hair, wearing a white dress shirt and a dark tie. He is looking upwards and slightly to his right with a thoughtful expression. The background is a bright, cloudy sky.

TODAY

Reflections

Scrum

Practical Stuff

# REFLECTIONS



- 3: No remarks
- 2: Minor remarks
- 1: Major remarks
- 0: Failed delivery

# NO REMARKS

“We assumed many things”

- Regular meetings with stakeholders
- Allow for change



# MINOR REMARKS

Loss in detail

E.g. no roles in planning phase



# MAJOR REMARKS



ONLY MENTIONS PROBLEM

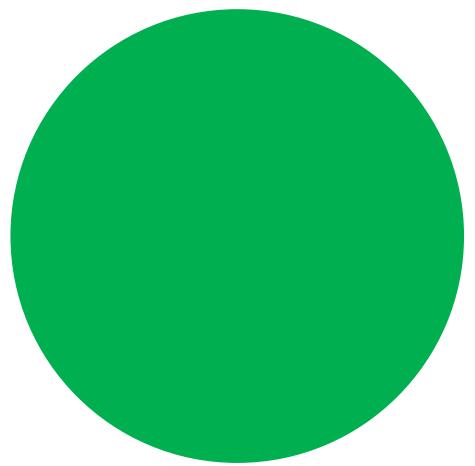
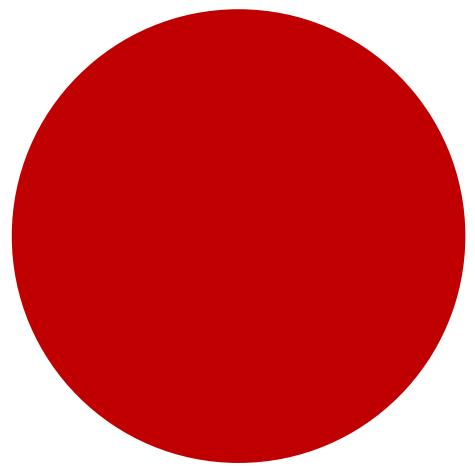
“THAT IS ONLY TO BE EXPECTED”

# FAILED DELIVERY



“THE SPRINTS WHERE TOO SHORT”

“HAVE A COMPLETE  
UNDERSTANDING UP-FRONT”



Vid produkt- eller tjänstutbyte mellan producerande bolag och kund är kunskapsnivåerna ofta varierande. I regel sitter det producerande bolaget på mer kunskap rörande den produkt eller tjänst som ska produceras. Av naturliga skäl eftersom att kunden är i färd med att köpa in möjligheten att tillhandahålla produkten eller tjänsten. Här uppkommer en asymmetri eftersom att aktören som ska beställa produkten eller tjänsten då besitter mindre kunskap än den aktör som ska leverera produkten eller tjänsten. Följden blir då att kunden relativt ovetande lägger fram ett förslag rörande t.ex. budget, tid och omfattning utan att egentligen veta förutsättningen för hur rimlig förfrågan är. Lärdomen här är att det producerande företaget då måste vara medveten om denna aspekt och förhandla med kunden för att på så sätt designa efterfrågan så att den matchar vad producenten kan leverera.

What: för vår grupp var 2-våningshuset ett för stort projekt, borde ha släppt det till en annan grupp

Why: anpassa projekt efter gruppens förmåga

How: Acceptera att det är en för stor uppgift och leta upp en mer lämplig

- Vad? Givet de önskemål och den prioriteringsordning produktägaren har, dvs. user stories, är det viktigt att utvecklingsteamet förstår varför önskemål och prioriteringsordning faktiskt ser ut som de gör.
- Varför? Om utvecklingsteamet förstår den grundläggande orsaken till varför ett specifikt produktönskemål förts fram, kan det ibland finnas andra lösningar än den av produktägaren föreslagna. Genom att kunna svara på frågan varför, kan följaktligen ibland andra, mindre komplexa lösningar, identifieras och utföras istället för det som initialt specificerades av produktägaren. När frågan om varför har besvarats har de olika parterna sannolikt, i större utsträckning, en gemensam läges- och problembild. Det vill säga, utvecklingsteamet och produktägaren delar samma syn avseende vilket problem som faktiskt åsyftas.
- Hur? Genom att kommunicera och ställa rätt frågor till produktägaren under sprintplaneringen har utvecklingsteamet, i samråd med produktägaren, möjlighet att identifiera potentiellt bättre lösningar som är mer rimliga att genomföra.

- Ta tid för kontinuerlig reflektion

Vi vill sträva efter att ta sig tid inom gruppen för kontinuerlig reflektion. Detta för att underlätta arbetet inom gruppen och förbättra det under projektets gång. Tas tid för kontinuerlig och gemensam reflektion kan olika förbättringar identifieras och börja användas redan under projektets gång. Dessutom ökar detta förståelsen mellan olika funktioner i gruppen och därmed kan vi förebygga potentiella konflikter inom gruppen och vi kan arbeta för att gemensamt optimera hela gruppens arbete. Detta vill vi göra genom att varje schemalagt möte inleds med ett möte där saker som funkar bra såväl som saker som kan utvecklas diskuteras.

## Take priorities into account

In order to meet the clients demands one needs to take the priorities of the user stories in account while choosing assignments. For example, at the Lego exercise the project owner prioritized roads for the city above all else. Despite that, no one took it upon themselves to construct roads to the city. No one even consulted the project owner. During yesterday's Lego exercise it became apparent that if you use scrum for a project you must always communicate with the project owner to find out the thoughts and needs of the customer. To make a customer happy you not only have to finish the tasks, the customer might also want them in a specific order.



# REFLECTION

“WHAT IS  
IN RELATION TO  
WHAT MIGHT OR SHOULD BE AND  
INCLUDES FEEDBACK TO REDUCE THE GAP”

R. SMITH, 2001



REALISTIC

# QA

'Questions don't have to make sense, Vincent', said Miss Susan.

'But answers do'

Terry Pratchett  
*Thief of Time*, 2001

# SCRUM



## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



**Product Owner**



**The Team**



**Product Backlog**

Team selects starting at top as much as it can commit to deliver by end of Sprint

**Sprint Planning Meeting**



**Sprint Backlog**



# Working with the backlog

Differentiation product backlog/sprint backlog

Product backlog: Prioritised list of **all** product requirements

Product backlog can be continuously updated by the Product Owner

Sprint backlog: List of requirements selected **for the current sprint**

Sprint backlog contains more detailed information: tasks

Sprint backlog is only updated by Scrum Team

# Backlog Items

## User stories

“As an entrepreneur I want a portable booth to be able to sell sausages to hungry citizens wherever they happen to be”

“Sausages!”

## Epics

Large stories that can't be delivered within one sprint

## Themes

Something that is common to a number of backlog items

Roads and bridges

Work places

Vehicles

# User Story Cards

- Describes the requirements and the acceptance criteria
- Can also hold information about the estimate (from Scrum Team) and the priority (from Product Owner)
- Can be updated continuously

Front of Card	Back of Card
<p>1B</p> <p>As a student I want to purchase a parking pass so that I can drive to school</p> <p>Priority: <del>Must</del> Should Estimate: 4</p>	<p><u>Confirmations:</u></p> <p><del>The student must pay the correct amount</del></p> <p>One pass for one month is issued at a time</p> <p>The student will not receive a pass if the payment isn't sufficient</p> <p>The person buying the pass must be a currently enrolled student.</p> <p>The student may only buy one pass per month.</p>

Copyright 2005-2009 Scott W. Ambler

I	Independent	User stories should not overlap and they should be formulated so they can be implemented in any order.
N	Negotiable	A user story should be an invitation for a conversation. It can be changed, augmented, and redacted; of course, always in dialog with the Product Owner!
V	Valuable	Each user story should deliver value, either to the Product Owner or to Scrum Team.
E	Estimable	It must be possible to assign effort to each user story. A story that can not be estimated is not complete!
S	Small	A user story must be a manageable task. If its completion takes longer than 3 or 4 days, it must be broken down!
T	Testable	There must be clear, testable criteria to define when the story is done in the eyes of the Product Owner and the Scrum Team.

[Buglione & Abran, 2013]

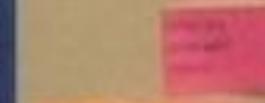
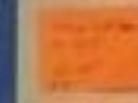
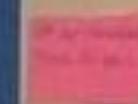
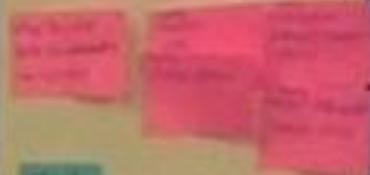
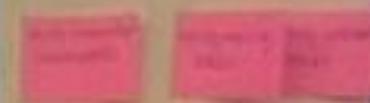
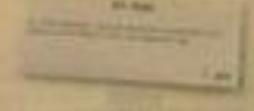
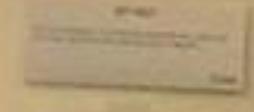
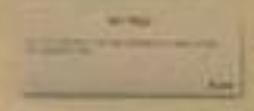
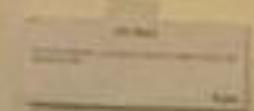
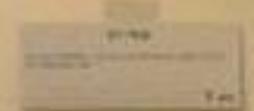
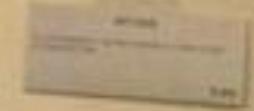
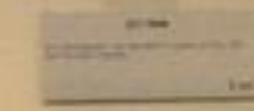
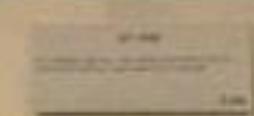
STORIES

TO DO

IN  
PROGRESS

TESTING

SPRINT  
ENDS  
12/22



# DoD - DEFINITION OF DONE

A checklist which usefully guides discussion, estimation and design

Should be helpful and not an obstacle

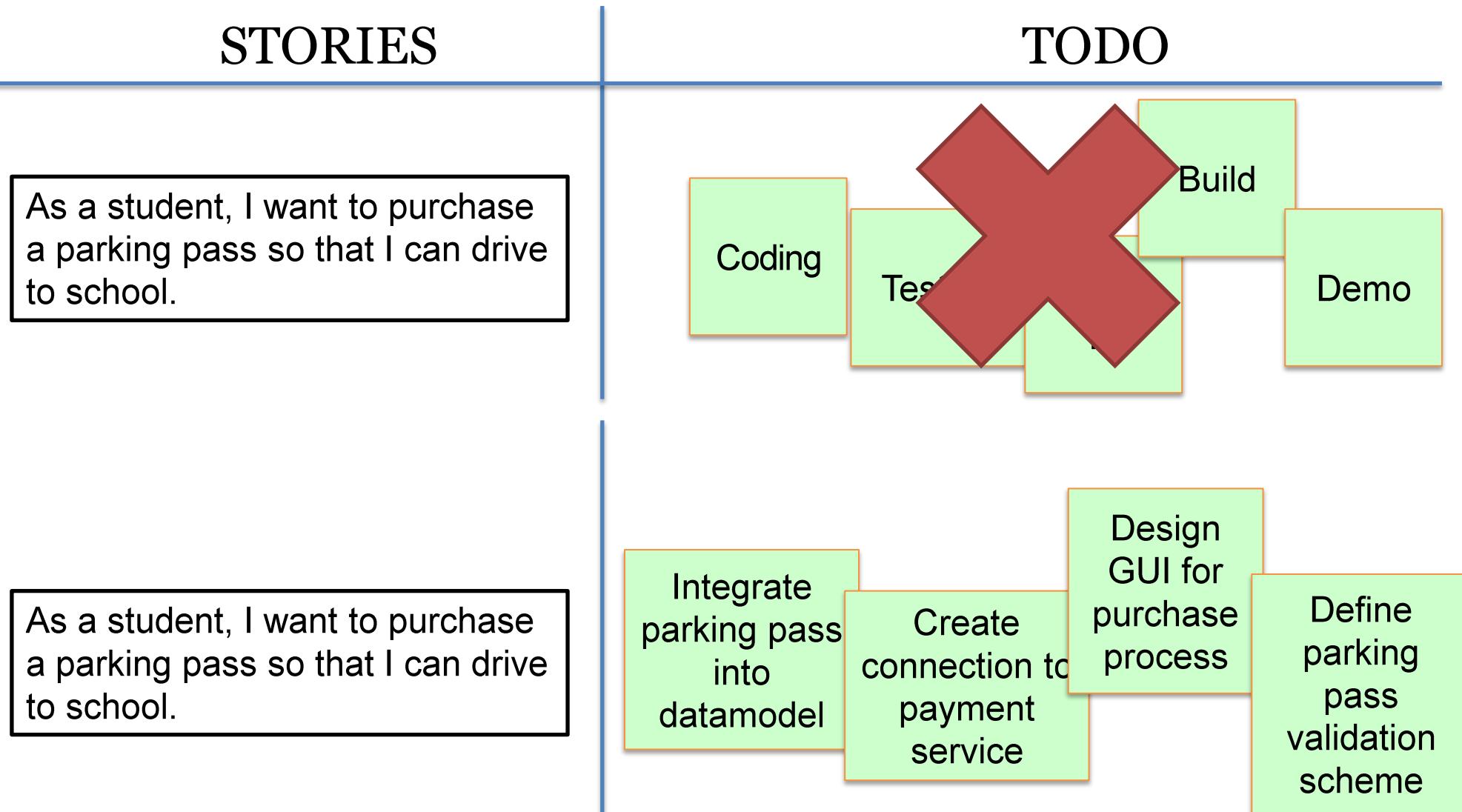
Team level – PO – Other teams

Can change over time

Code – Test – Integrate – Demonstrate – Ship

# Tasks

User stories are usually “too big” to tackle as one



# SLICING

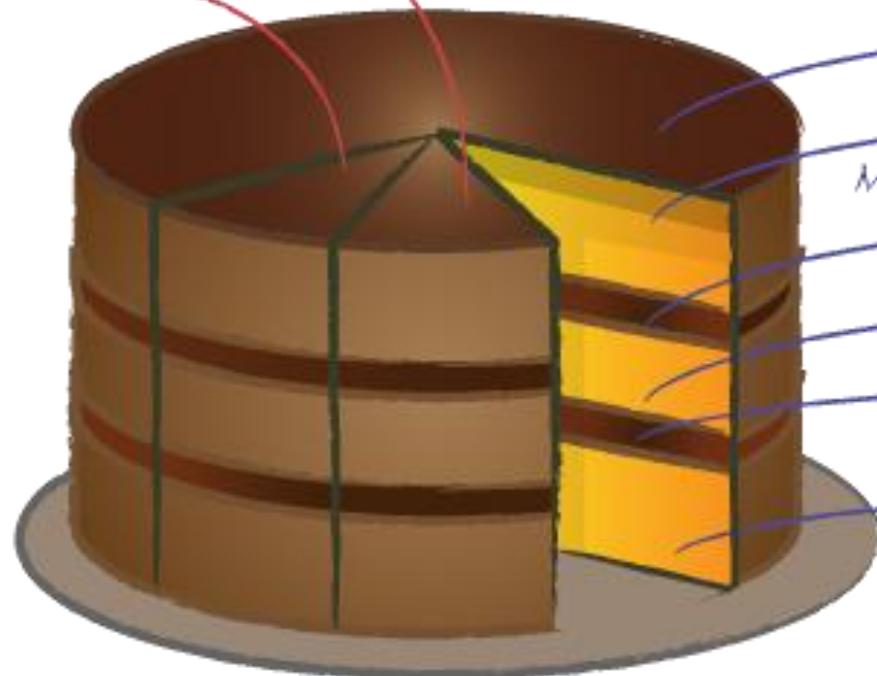
Automated Teller Machine (ATM)

Horizontal and Vertical User Stories - Slicing the Cake

Vertical User Stories

Cash Withdrawal (90% usage)

Bank Statement



Horizontal Stories

UI - PIN and Card Reader

Security Layer

Middleware - Transaction Protocol

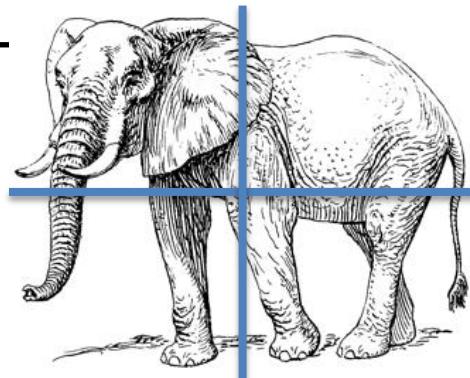
Tuxedo DB Interface

Transport Protocol

Bank Mainframe Database

# ELEPHANT CARPACCIO/LAMINATING

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



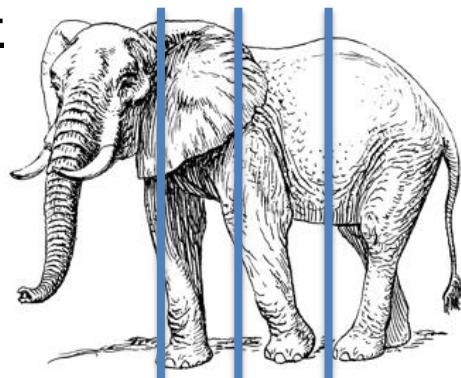
Integrate parking pass into datamodel

Create connection to payment service

Design GUI for purchase process

Define parking pass validation scheme

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



Implement purchase process without payment

Integrate payment into the purchase process

Allow users to see their past orders

Define parking pass validation scheme

## Exercise on Apr 05: Create a backlog

Vision: Retail calculator – calculate prices for deliveries in other countries

Three Inputs:

How many items

Price per item

2-letter country code

Output: total price of the order.

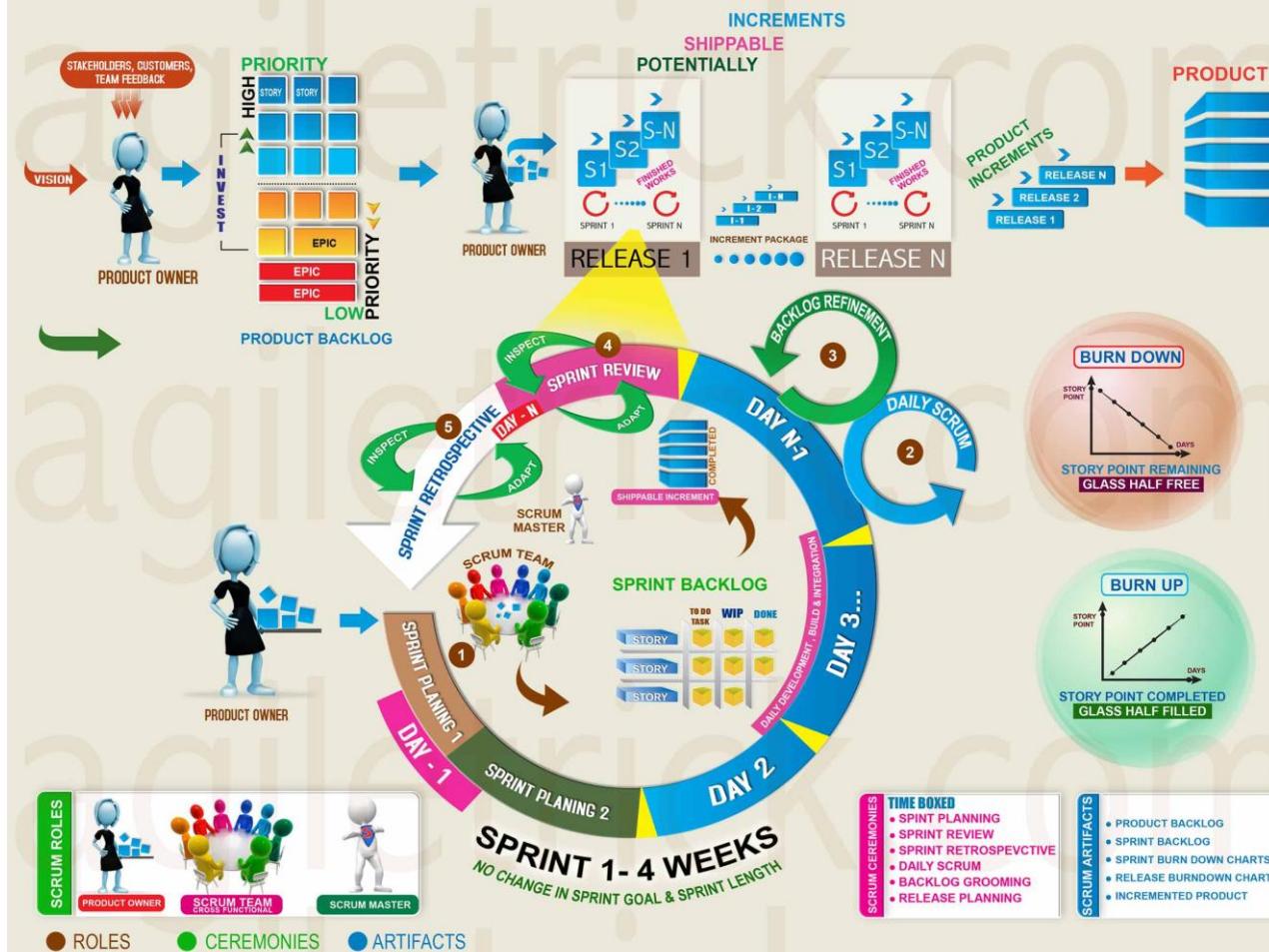
Algorithm: Give a discount based on total price, then add state tax based on country code and discounted price.

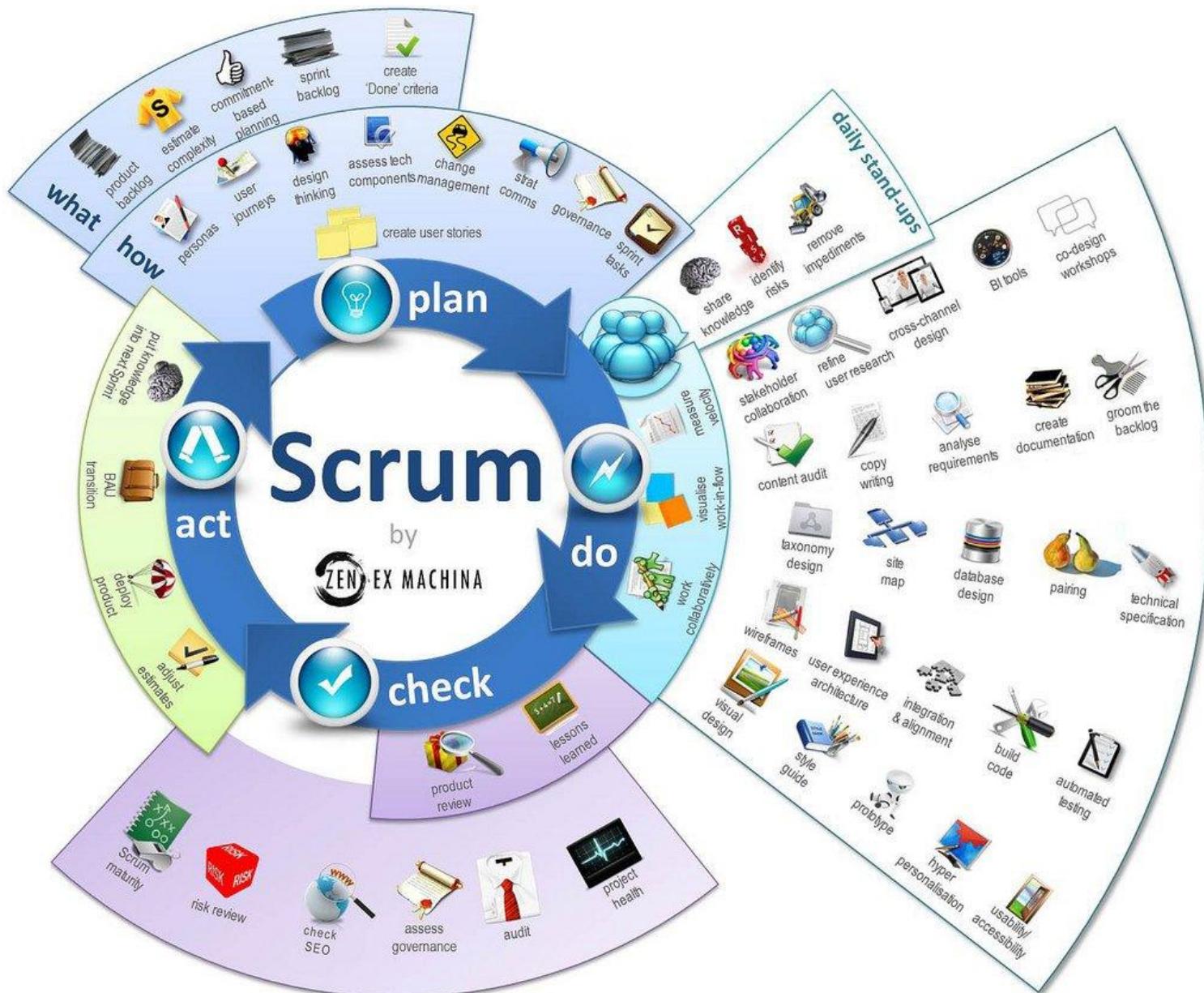
Country	VAT	Amount	Discount
Belgium (BE)	21%	10.000 SEK	2%
Germany (DE)	19%	50.000 SEK	3%
Hungary (HU)	27%	100.000 SEK	5%
Sweden (SE)	25%	200.000 SEK	8%
United Kingdom (UK)	20%	500.000 SEK	10%

Slice user stories as small as possible!

# THE SCRUM FRAMEWORK

agiletrick.com





# WAY OF WORKING

When to meet

How to meet

What to do

Definition of done

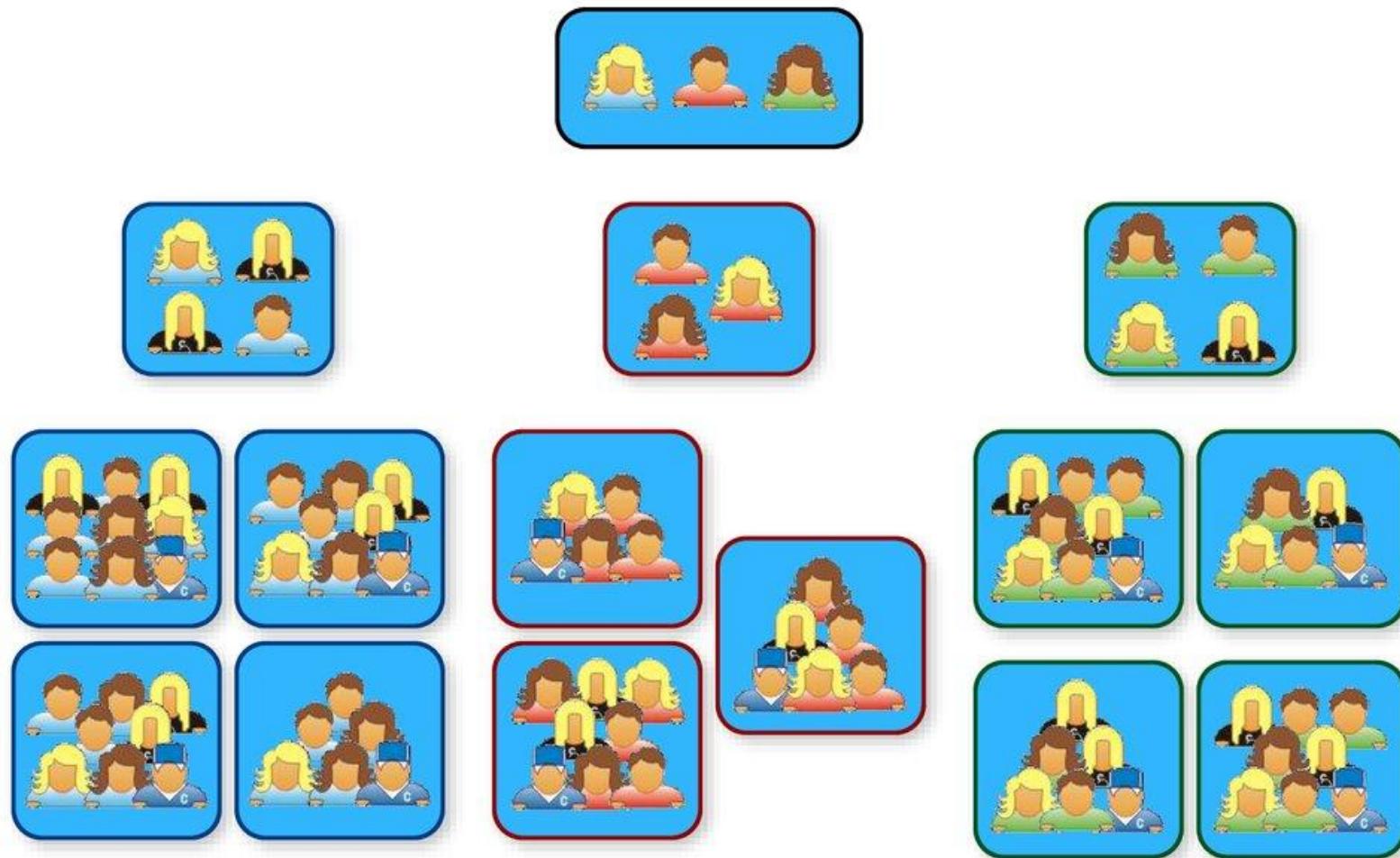
Roles

Best practices

...

Reflection

# SCRUM OF SCRUMS



# SCRUM OF SCRUMS

Regular meetings:

- 1) What has your team done since we last met?
- 2) What will your team do before we meet again?
- 3) Is anything slowing your team down or getting in the way?
- 4) Are you about to put something in another team's way?

Possible tools and practices:

- Slack / Facebook group
- Epic board
- Git repo
- Meet at lectures and supervision
- Rotate representatives
- Social contract
- DoD
- ...

# QA

'Questions don't have to make sense, Vincent', said Miss Susan.

'But answers do'

Terry Pratchett  
*Thief of Time*, 2001