



# **Object-oriented Programming Project**

**User Stories** 

Dr. Jan-Philipp Steghöfer & Dr. Alex Gerdes TDA367/DIT212 - HT 2018



- Organisation:
  - Make groups, deadline Wednesday 5 Sept. at 13:00
  - Start thinking about project idea (lagom!)
  - Needs to have some model, not only view for external back-end



# Summary previous lecture



- Software development
  - Different models
  - Our approach: iterative
  - Support for development









- · JPS:
  - User Stories
  - Task breakdown
  - Backlog (scrum-ish)
  - Exercise
- AG:
  - User Stories in our course







# A Primer to Agile Requirements Engineering

Jan-Philipp Steghöfer September 4, 2018



### Scrum





## **Backlog Items**

#### **Epics**

Large stories that can't be delivered within one sprint "I want to feed the world an earn money doing it"

#### **User stories**

As an X I want Y since Z

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

**Product Backlog** 

**Product Backlog** 

Sprint Backlog

Sprint Backlog

#### Tasks

Small work units that can usually be delivered by one person within a short period "Create a sign that can be erased to show the offers and prices and that can be attached to the portable booth."





#### **User Story Cards**

- Describe 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





## **INVEST Criteria**

I	Independent	User stories should not overlap and they should be formulated so they can be implemented in any order.
Ν	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!
Т	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] http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

UNIVERSITY OF GOTHENBURG

## **Acceptance Criteria**

- Valid for a specific user story
- Describes when value is realised for customer
- Should be testable
- Basis for estimations
- Can change over time
- Has to be fulfilled before story is moved to testing stage

## DoD – Definition of Done

- Valid for all user stories
- A checklist which usefully guides discussion, estimation and design
- Should be helpful, not an obstacle
- Team level PO Other teams
- Can change over time
- Has to be fulfilled before story is moved to done stage

#### Backlog

As a terminal worker, I want the berth view to show the estimated time of the next event of a vessel (either ETB, ETC or ETD, instead of it always showing time of arrival at harbour), so it can give me relevant information without having to click on the vessel  $\equiv$   $\bigtriangledown$  0/5

As a terminal worker, I want the Portcall view to only shows vessels that my terminal has been assigned to because I am not interested in the other vessels

=

As a terminal worker, I want to have some preset Favorite States that are relevent for me so I dont manually have to choose them myself (the list is really long)

≣ ⊠ 0/4

As a terminal worker, I want to be able to add a new cargo type and its loading speed in the calculator view so that I can use the calculator for all cargo types

#### + Add another card

ToDo

As a terminal worker, I want the Portcall view to not show vessels that have already left my terminal because I am no longer interested in those vessels

≣ ⊠ 0/4

As a terminal worker, I want to get a suggested ETC by combining ETB and duration of cargo operations, so that I don't have to calculate it myself.  $\equiv \Box 0/8$ 

+ Add another card

#### In Progress

JS 1 2

3 As a terminal worker, I want to be able to automatically calculate how long cargo operations will take for Gasoline depending on the volume, so that I don't have to calculate it myself.

≣ ⊠ 5/7

+ Add another card



\_\_\_\_

Done

**⊠ 8/8** 

As a team, we want everyone to have the same repo of the app so that we can later merge easily

& Port Calls Qsewich 18

1. As a terminal worker, I want the Portcall view to show vessels together with ETA at pilot station (sorted by NOR or ETA), so I can get an overview of work to come

≣ ⊘ 1 🖸 10/10

As a team, we want to merge our branches into a single repository so that we can properly show our features on Wednesdays presentation

≣ 🖸 1/1

As a terminal worker, I want to be able



#### User stories are usually "too big" to tackle as one





#### Horizontal vs. vertical user stories



http://www.deltamatrix.com/horizontal-and-vertical-user-stories-slicing-the-cake



### **Vertical Slicing/Laminating**





## **SMART Criteria for Tasks**

Specific	A task needs to be <i>specific</i> enough that everyone can understand what's involved in it. This helps keep other tasks from overlapping, and helps people understand whether the tasks add up to the full story.
Measurable	The key <i>measure</i> is progress till completion. That means that it should be clear when the task can be moved to the next column. To determine whether the task is finished, the Definition of Done should be applied.
Achievable	The task owner should expect to be able to <i>achieve</i> a task. XP teams have a rule that anybody can ask for help whenever they need it; this certainly includes ensuring that task owners are up to the job.
Relevant	Every task should be <i>relevant</i> , contributing to the story at hand. Stories are broken into tasks for the benefit of developers, but a customer should still be able to expect that every task can be explained and justified.
Time-Boxed	A task should be <i>time-boxed</i> : limited to a specific duration. This doesn't need to be a formal estimate in hours or days, but there should be an expectation so people know when they should seek help. If a task is harder than expected, the team needs to know it must split the task, change players, or do something to help the task (and story) get done.

https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/





## **Elephant Carpaccio**

- Vision: Retail calculator calculate prices for delivery to other countries Slice user stories as small as possible!
- Three Inputs:
  - 1. How many items
  - 2. Price per item
  - 3. 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.

Amount	Discount	
10.000 SEK	2%	
50.000 SEK	3%	
100.000 SEK	5%	
200.000 SEK	8%	
500.000 SEK	10%	

Country	VAT
Belgium (BE)	21%
Germany (DE)	19%
Hungary (HU)	27%
Sweden (SE)	25%
United Kingdom (UK)	20%





- Split "5 states, 5 discounts" into as many slices as possible
- Each slice must have UI, input & output, and be visibly different from last slice





- Split "5 states, 5 discounts" into as many slices as possible
- Each slice must have UI, input & output, and be visibly different from last slice

UNIVERSITY OF GOTHENBURG





**Definition of Done!** 

Difference between Product and Sprint Backlog!

# **User Stories in our course**

- Documentation of User Stories
  - In Requirements and Analysis Document (RAD)
- Workflow:
  - Project idea: have this really clear
  - Define User Stories
  - Prioritise the User Stories
  - Break down the User Stories in tasks
  - Make rough estimation of User Stories (in person-days)
  - Make a selection for 2 to 3 weeks
  - Design (next lecture)
  - Implement (start with defining tests, TDD)
  - Check acceptance criteria
  - Reflect and iterate

