# User Stories

- Appeared in the context of eXtreme Programming (XP), 1998
- Written descriptions of a user making use of a system to reach some goals
- Extremely important for any Agile process, framework and method

# User Stories

- Key ideas:
  - Written descriptions of a user making use of a system to reach some goals
  - Usually, written in *story cards*
  - Describe functionality that will be valuable to either a user or purchaser of the software
  - Conversations between developers and users are written as *Details* and Confirmations are recorded as *Acceptance Tests*
  - Not exactly requirements; user stories are best understood as *pointer to requirements* (Cohn, 2004)
  - It's the user (or customer) who writes the story, not developers
- Simple heuristic for determining quality of a user story:
  - If a user cannot establish some priority to a user story, then probably we are facing a bad user story (rationale: if he or she cannot establish priority, then probably there is no value for the user)

# User Stories

- **User Story example:**
  - **Story:** ***"As a student, I can request all grades of all approved courses, so I can see my overall performance"***
  - **Conversations:** ***"Need to see my grades with no decimal point, grades below 55 with red, grades in range 0 to 100".***
  - **Acceptance Tests:** ***Test with grade 110, Test with grade 55, Test with grade 49.***
- **Key aspects:**
  - **Role:**
    - **Student**
  - **Functionality:**
    - **Requesting grades of all approved courses**
  - **Value:**
    - **Students will see their own performance in the program**

# User Stories

- **Key question: Who *writes* the card?**
  - **"User writes the story" may sound simple, but…**
  - **… in practice, users and/or customers are busy enough for writing stories**
  - **By "User writes the story" we mean the intellectual process of writing the story**
  - **There is no problem with having one or more developers physically write in paper (or computer) what the user/customer says**

## Evaluating User Stories

- **A User Story is good if it is (Wake, 2003):**
  - **Independent**
  - **Negotiable**
  - **Valuable (to users or customers)**
  - **Estimable**
  - **Small**
  - **Testable**
- **How to remember these characteristics?**
  - **Bill Wake suggested the acronym INVEST**

## Estimating User Stories

- **We can estimate User Stories by using Story Points**
  - **Each team define what a Story Point means**
    - **For one team, a story point could be an ideal day of work (i.e., a day with no interruptions of mails, phone calls, meetings…)**
    - **For others, a story point could be a measure of the complexity of the story**
  - **What happens if the team uses *pair programming*?**
    - **The team can use *"ideal pair days"* or something similar**
  - **Some teams use Fibonacci sequence for assigning the numeric value to story points (rationale: more points, more complexity)**
    - **However, this is not recommended as this means the team have not passed the INVEST test**
- **Why to estimate User Stories?**
  - **Estimating User Stories allows the team to decide how much stories could be developed in an iteration**
  - **As the project goes on, the team can decide more precisely how many stories could be developed in each iteration**

# Estimating User Stories: Planning Poker

- **Five steps for estimating user stories:**
  - **Step 1: bring together user/client and developers (team)**
  - **Step 2: user/client takes one random User Story and reads it to developers. If questions arise, developers can ask user/client for more details.**
  - **Step 3: each developer writes a secret value for estimating the User Story (with no intervention between developers)**
  - **Step 4: developers show their estimations**
  - **Step 5: if the values are very different, developers discuss the differences**
    - **Go back to Step 3 until reaching some convergence in the values**
- **Key aspect:**
  - **The owner of a User Story is the user/client, but the owner of the estimation process is the team**
    - **Interventions from user/client are not allowed**

# Triangulating User Stories

- **We can check if a story is a well-estimated one by triangulating them**
  - **Create several columns; one for each story point value (e.g., 1-2-3-4…)**
  - **Put each estimated story in the corresponding column (classify them by using the story point value)**
  - **The team has to check if there are stories wrongly estimated**

This story is clearly wrongly estimated as developing a paying module is much more complex than developing a log-in module

| 1 [sp] | 2 [sp] | 3 [sp] |
|---|---|---|
| As a student I can log-in the system | - | As a student I can request a list of books requested by year |
| As a student I can pay all my debts in the library | - | |