

User stories

Antonio Brogi

Department of Computer Science
University of Pisa



Agile software development

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

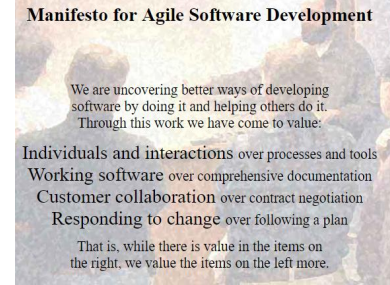
Customer collaboration over contract negotiation

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.

(2001)

Twelve principles of agile software



1. Our highest priority is to satisfy the customer through **early and continuous delivery of valuable software**.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. **Working software is the primary measure of progress**.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. **Simplicity**--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Agile software development
User requirements elicitation





How the customer explained it.



How the project leader understood it.



How the analyst designed it.



How the programmer wrote it.



What the customer really wanted.



Agile software development
User requirements elicitation
User stories
Example stories





«Working software is the primary measure of progress.»

«Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.»

«Simplicity--the art of maximizing the amount of work not done--is essential.»



- organize and chunk work into units that represent value to the customer
- distinguish more valuable from less important
- visualize units of work, through simple stories
- template: **As a** (who) + **I want to** (what) + **So I can** (why)
- increase quality of stories makes rest of development process more efficient
- user stories create business value for the users
- build software from the users perspective



Agile software development
User requirements elicitation
User stories
 Example stories
 Splitting stories

YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.



www.dilbert.com scottadams@aol.com

DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?



4/14/01 © 2001 United Feature Syndicate, Inc.

GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.











- Think about application the way the user thinks about it
- User thinks of an application in terms of the behaviour she finds valuable, in terms of «slices of functionality»
- Build software incrementally, as slices of functionality
- A good story should take 1 or 2 days of work
- Split long stories into several shorter stories, which can be prioritized



Agile software development
User requirements elicitation
User stories
 Example stories
 Splitting stories
 User story cards

User story cards

User stories often written on **physical cards**, with the format:

As a *<role>*
I want to *<requirement>*
So that *<value>*

As a
I want to
So that

Produce stakeholders include an indication of the **priority** of the story (e.g., [1,10],{H,M,L},...)

Developers (may) include an **estimate of effort** needed to implement the story

User stories are sorted according to priority



Agile software development
User requirements elicitation
User stories
 Example stories
 Splitting stories
 User story cards
 Workshop: Tic-tac-toe



I hear. *I forget.*



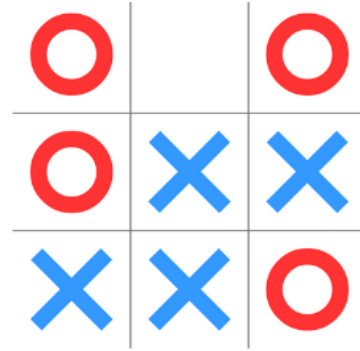
I see. *I remember.*



I do. *I understand.*

CHINESE
PROVERB

Tic-tac-toe



Class divides into groups

Each group writes user stories on an app for playing [Tic-tac-toe](#)

/* Tic-tac-toe user stories generated in class with highest priority */

As a	User
I want to	Have a 3x3 board
So that	I can play my turns by inserting crosses or circles
Priority	1.0
Effort	30 min
As a	User
I want to	Alternate myself with opponent in different turns
So that	I can react to my opponent moves
Priority	1.1
Effort	60 min
As a	User
I want to	Occupy a free cell with my symbol
So that	I can play my turn
Priority	1.1
Effort	45 min
As a	User
I want to	Win if there are three of my symbols in a row/column/diagonal
So that	The game ends
Priority	1.3
Effort	32 min
As a	User
I want to	See when all cells are occupied
So that	I See the end of the game
Priority	1.4
Effort	32 min

/* Other Tic-tac-toe user stories generated in class */

As a User
I want to Play a match on the same device
So that I can have fun with my friend
Priority 5
Effort

As a User
I want to Play remotely with other people
So that I can improve my skill and have fun
Priority 6?
Effort

As a User
I want to See my game statistics
So that I can keep track of my previous matches
Priority 7?
Effort

As a User
I want to Have a list of friends
So that I can play with my friends
Priority 9
Effort

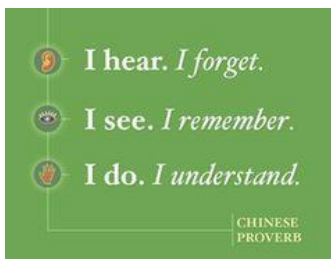
As a User
I want to Play with an AI opponent
So that I can play alone and off-line
Priority 5??
Effort

As a User
I want to Choose at beginnning between cross and circle
So that Choose my favourite icon
Priority 10
Effort



Agile software development
User requirements elicitation
User stories
 Example stories
 Splitting stories
 User story cards
 Workshop: Tic-tac-toe
The grand view





Testing

User stories

Monolith

Microservices

Improving service interactions

Monitoring

Securing

Packaging

Deploying



* If time allows

Beep Beep



- Toy application for runners
- Runners can see their runs, training plans, and get periodic activity reports
- Beep Beep accounts hooked to **STRAVA** (social fitness network)



Agile software development

User requirements elicitation

User stories

- Example stories

- Splitting stories

- User story cards

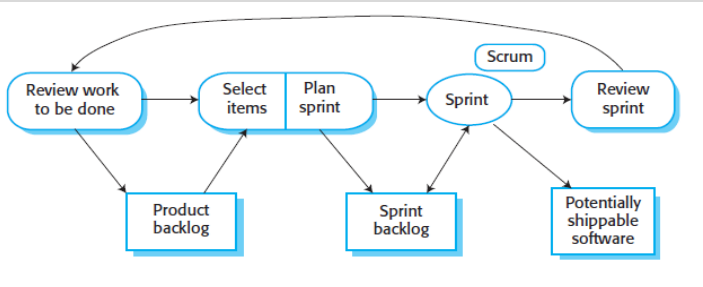
- Workshop: Tic-tac-toe

The grand view

Beep Beep user stories (workshop)

Agile processes: e.g., Scrum

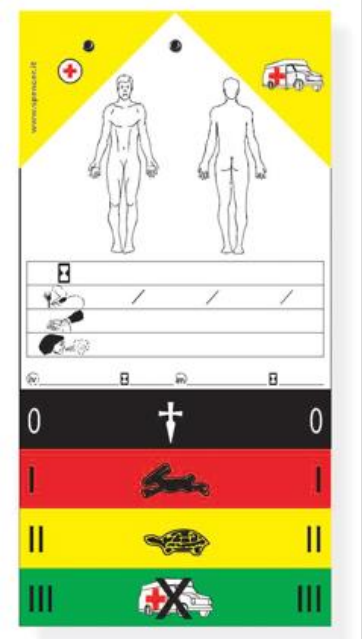
Scrum term	Definition
Development team	A self-organizing group of software developers, which should be no more than seven people. They are responsible for developing the software and other essential project documents.
Potentially shippable product increment	The software increment that is delivered from a sprint. The idea is that this should be “potentially shippable,” which means that it is in a finished state and no further work, such as testing, is needed to incorporate it into the final product. In practice, this is not always achievable.
Product backlog	This is a list of “to do” items that the Scrum team must tackle. They may be feature definitions for the software, software requirements, user stories, or descriptions of supplementary tasks that are needed, such as architecture definition or user documentation.
Product owner	An individual (or possibly a small group) whose job is to identify product features or requirements, prioritize these for development, and continuously review the product backlog to ensure that the project continues to meet critical business needs. The Product Owner can be a customer but might also be a product manager in a software company or other stakeholder representative.
Scrum	A daily meeting of the Scrum team that reviews progress and prioritizes work to be done that day. Ideally, this should be a short face-to-face meeting that includes the whole team.
ScrumMaster	The ScrumMaster is responsible for ensuring that the Scrum process is followed and guides the team in the effective use of Scrum. He or she is responsible for interfacing with the rest of the company and for ensuring that the Scrum team is not diverted by outside interference. The Scrum developers are adamant that the ScrumMaster should not be thought of as a project manager. Others, however, may not always find it easy to see the difference.
Sprint	A development iteration. Sprints are usually 2 to 4 weeks long.
Velocity	An estimate of how much product backlog effort a team can cover in a single sprint. Understanding a team’s velocity helps them estimate what can be covered in a sprint and provides a basis for measuring improving performance.



Kanban boards



Requirements triage



Learn by **DOING**.





- Toy application for runners
- Runners can see their runs, training plans, and get periodic activity reports
- Beep Beep accounts hooked to **STRAVA** (social fitness network)

User stories

- Class divides into groups
- Each group writes user stories for Beep Beep
 - runs
 - reports
 - training plans
 - *<your creativity>*

Tip: Keep an eye on STRAVA API constraints

<http://developers.strava.com/docs/reference/>

As a Unregistered User
I want to Register
So that I can use app functionalities
Priority 1.1

As a Registered User
I want to Authenticate
So that I can hook to Strava & access my data
Priority 1.2

As a Connected User
I want to Logout
So that I can disconnect myself
Priority 1.2

As a Connected User
I want to See all my previous runs on a list
So that I can keep track of them
Priority 1.3

As a Connected User
I want to See average speed of all my runs
So that I can improve myself
Priority 1.4

As a Registered User
I want to Delete my account
So that All my data will be deleted
Priority 1.5

As a Connected User
I want to Click on a previous run
So that To see speed & distance of a single run
Priority 2.1

As a Connected User
I want to Set a training objective
So that I can plan my running
Priority 2.1

As a Connected User
I want to Challenge a previous run
So that I can challenge myself
Priority 2.2

As a Connected User
I want to Compare different run statistics
So that I can improve myself
Priority 2.3

As a Connected User
I want to See distance to my set training objective
So that I know how much to run
Priority 2.4

As a Registered User
I want to Get a configurably periodic report via email
So that I can get them when I feel so
Priority 3.1

As a Registered User
I want to Get tips on when to run (based on my current training objective and weather forecast)
So that I can run in my ideal conditions
Priority 3.2