zendesk

Scrum in the Real World

Dina Friis

Scrum Master, ZENDESK

Scrum is:

- Lightweight
- Simple to understand
- Difficult to master

Challenges for a Scrum Master

- The definition of Done is when the code is in production
- Developers takes responsibly for the quality of the code base
- The Backlog Items are very ready
- Number of bugs increase
- Team feels that problems are out of their control zone
- Daily Stand up ends up being reporting
- The experienced developers are supporting the whole organisation
- Team overcommits

zendesk

Definition of Done

Code is reviewed and then put in production on the same day.

But from experience experience we know that when a story is estimated to 13 story points or more it means we do not know.

But if we break them further down a part of the story can not go into production alone.

zendesk

Definition of Done

A common language referring to the process must be shared by all participants

Those performing the work and those accepting the work product must share a common definition of "Done"

When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment.

If there are multiple Scrum Teams working on the system or product release, the development teams on all of the Scrum Teams must mutually define the definition of "Done."

Definition of Done

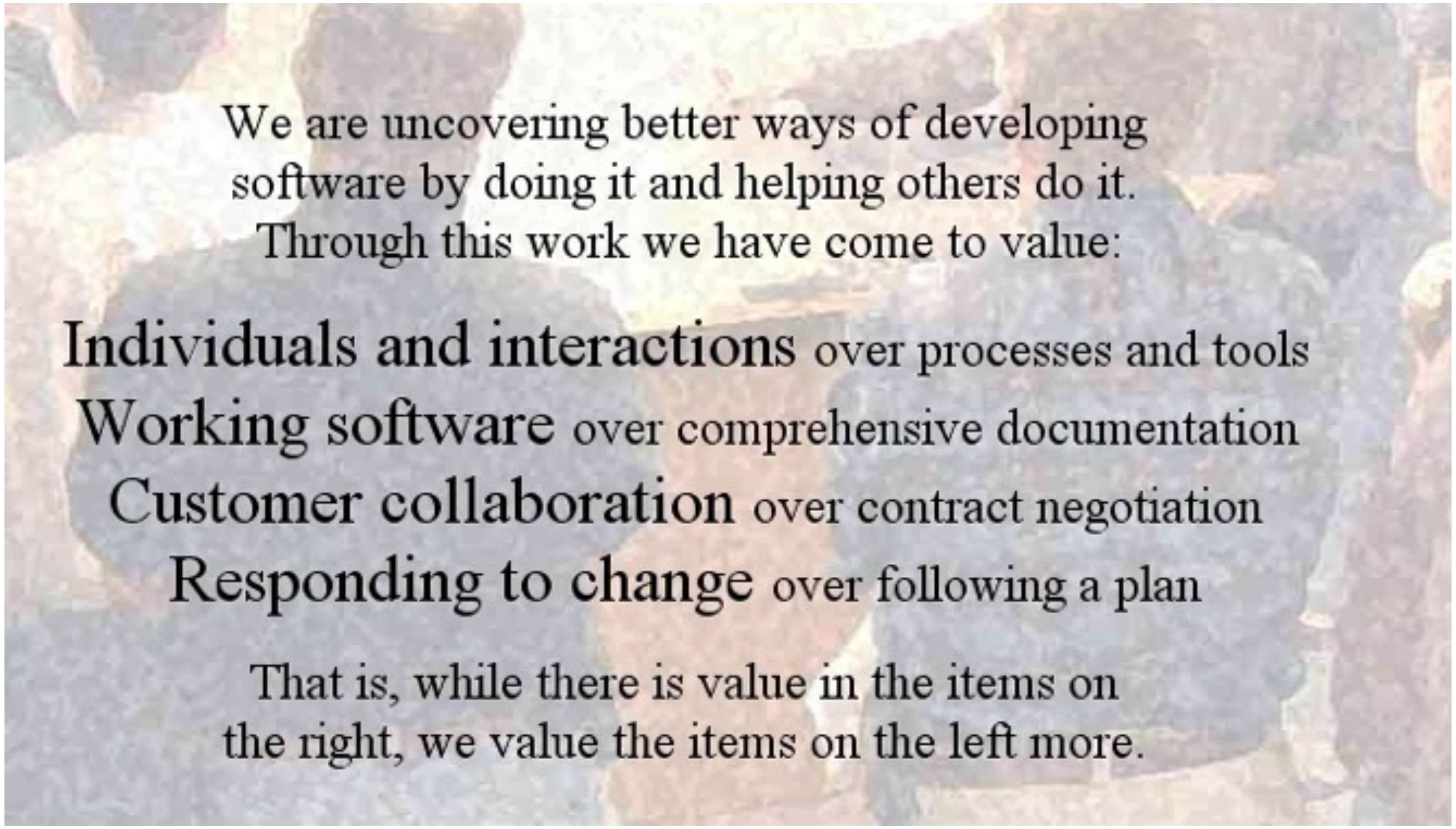
A Contract that secures that a story is finished when there is no more work

zendesk

Definition of Done

Is code acceptance by two developers a god definition of done?

Agile manifesto



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.

Developers takes responsibly for the quality of the code base

=>

Refactoring

Refactoring

- Takes time
- Can introduce errors
- Benefits is hard to measure

Refactoring

The Product Owner decides!

Development Team is stakeholders.

Principles behind the Agile Manifesto

Continuous attention to technical
excellence
and good design enhances agility.

zendesk

Principles behind the Agile Manifesto

Agile processes promote sustainable
development.
The sponsors, developers, and users should be
able
to maintain a constant pace indefinitely.

zendesk

The Backlog Items are very ready

- Boring for developers
- Inhibits new ideas
- Hard to set goals
- Motivation goes down

Principles behind the Agile Manifesto

Build projects around motivated individuals.

Give them the environment and support they need,
and trust them to get the job done.

zendesk

The Backlog Items are very ready

- Put a background story on items and roadmaps
- Explain why and not just what
- Cut down on details
- Inspect and adapt until the right level is found

Number of bugs increases

- Product Owner owns the product and all bugs in it
- Team is responsible for the quality they deliver

Team feels that problems are out of their control zone

Use “The Soup”

http://www.innovationgames.com/circles-and-soup/

Daily Stand up

The Team reports to Scrum Master

The Team reports out in the air

The Team leaves before everyone is finished

The experienced developers are supporting the hole organisation

- They are overworked
- Work is not visible
- New developers never learns
- Velocity is random

The experienced developers are supporting the whole organisation

- The team is responsible
- Make work visible
- Enforce the firewall
- Take stuff out of the Sprint

Team overcommits

- Progress is not predictable
- Estimates are not taken seriously
- Team is under constant pressure
- Team never experiences success

Team overcommits

Use yesterdays weather

Set goals

Try things

Inspect and Adapt