



Fabian Gilson

Software Engineering II

SENG301 - Agile Software Development

Lecture 2 - Semester 1, 2020



In spite of its name, software engineering requires - cruelly - hard science for its support.

- Edsger W. Dijkstra

EWD 498 - How do we tell truths that might hurt? - 1975

A close-up photograph of a person's hand holding a silver pen over a white sheet of paper. The paper contains a printed checklist with several items, each preceded by a small square checkbox. The hand is positioned as if it is about to write or has just finished writing in one of the boxes.

Agile development

Agile core values

Initiated by 17 researchers and IT professionals in 2001

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

- First principle of Agile manifesto

Four core values

- **individuals** and **interactions** over processes and tools
- **working software** over comprehensive documentation
- **customer collaboration** over contract negotiation
- **responding to change** over following a plan

For all authors and signatories, see agilemanifesto.org

Agile principles (I)

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

" Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage. "

" **Deliver** working software **frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale. "

" Business people and developers must **work together daily** throughout the project. "

" Build projects around **motivated individuals**. Give them the environment and support they need, and **trust them** to get the job done. "

Agile principles (II)

" *The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.* "

" **Working software** is the primary measure of progress. "

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

" Continuous attention to **technical excellence** and **good design** enhances agility. "

" **Simplicity** – the art of maximizing the amount of work not done – is essential. "

" The best architectures, requirements, and designs emerge from **self-organizing teams**. "

Main Agile variants

Scrum

- team commitment and **empowerment**
- objectives in sprint deliveries
- improved code quality and documentation
- Beedle et al., "SCRUM: An extension pattern language for hyperproductive software development", 1999

Lean

- aka **optimisation** of processes (more than *agile*)
- limit waste and decide as late as possible
- neither over-production nor over-processing
- Poppendieck and Poppendieck, *Lean Software Development: An Agile Toolkit*, 2003

Kanban

- *lean*-oriented and reuse part of scrum
- no up-front planning (w.r.t scrum)
- **continuous changes** in one global board
- Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*, 2010



Scrum values, actors and tools

Scrum values



Scrum roles

Product owner (PO) *aka customer voice*

- translates user demands into **stories**
- maintains and prioritises the list of things to do (*aka backlog*)
- negotiates content of **releases** and **timing** with the team

Scrum master (SM) *aka process leader and facilitator*

- acts as a **coach**, *i.e.* does not command
- facilitates **communication inside** and **outside** the team
- represents managements, but **protects** team members

Team *aka developers*

- everyone is a **developer** (often even the SM), no hierarchy
- **self-organising** and **cross-functional** (*i.e.* low bus factor)
- **collective responsibility** for achievements (*i.e.* snow ploughing)

Backlog



Backlog items (I)

epic

- a **large piece of work** that may span over multiple sprints
- **abstractly defined** high-level requirement
- may look like a story, but with fuzzier definition
- must be broken down into stories

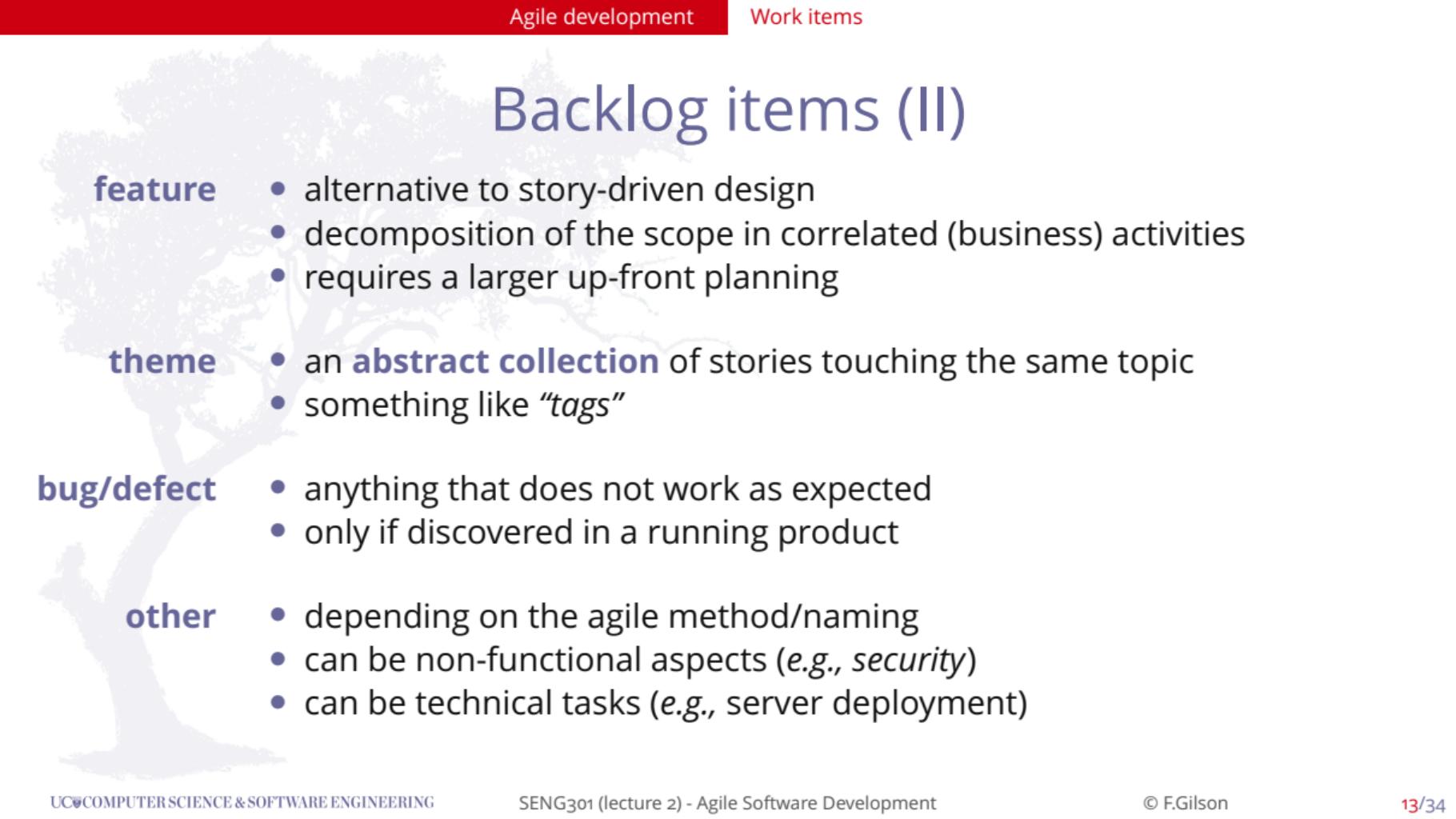
story

- a **small** and **well-defined** piece of work
- **concretely** and **extensively** defined requirement
- must be **handled in one** single sprint

task

- **anything concrete** and **time-boxed** that must be done
- may be split into subtasks
- may only be **assigned to one** (pair of) programmer(s)

Backlog items (II)

- 
- feature**
 - alternative to story-driven design
 - decomposition of the scope in correlated (business) activities
 - requires a larger up-front planning
 - theme**
 - an **abstract collection** of stories touching the same topic
 - something like “tags”
 - bug/defect**
 - anything that does not work as expected
 - only if discovered in a running product
 - other**
 - depending on the agile method/naming
 - can be non-functional aspects (*e.g., security*)
 - can be technical tasks (*e.g., server deployment*)

Product backlog and Sprint backlog

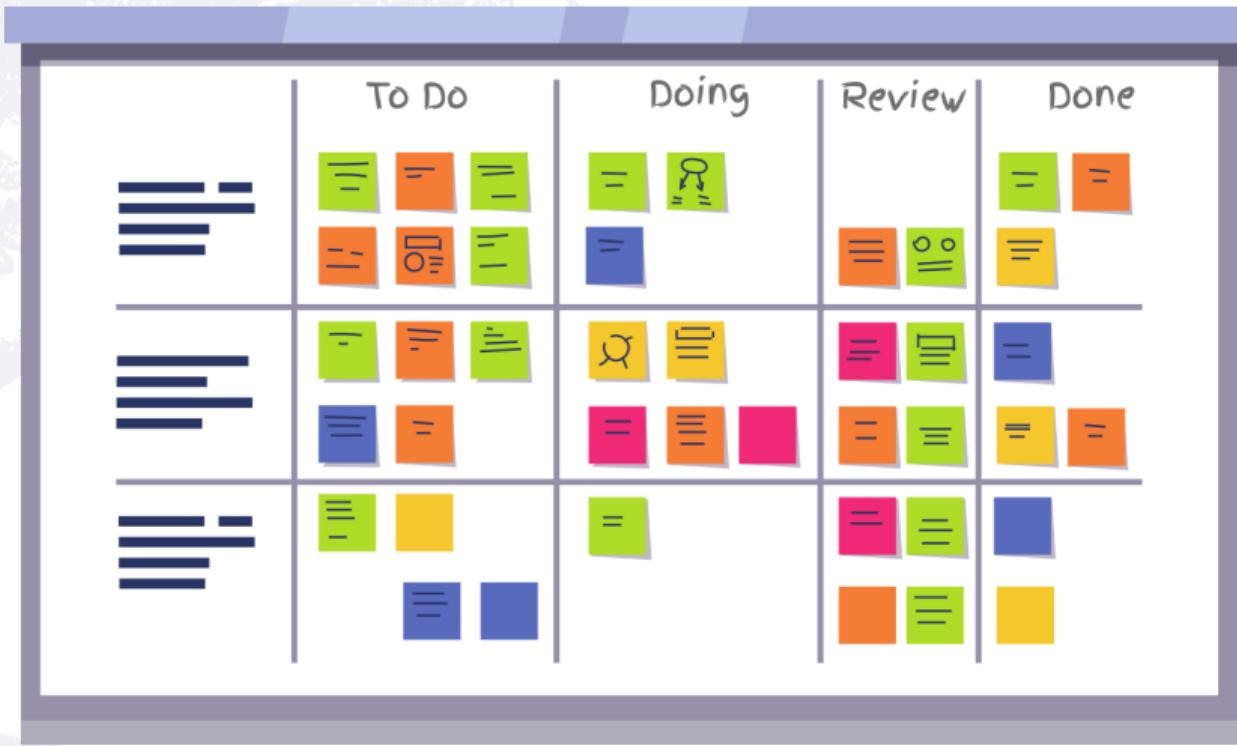
Product backlog

- contains **product backlog items** (PBIs)
- **everything** that still must be done
- **prioritised** and **maintained**
- **some estimations** may be missing

Sprint backlog

- items that will be handled this **iteration**
- **should be fixed** for the sprint
- only **high-priority items may modify** current sprint
- **all estimations** must be set

Agile board



Snow ploughing





Story breakdown

Story

User's point of view regarding a system's functionality

- one functionality and one goal
- written in natural language by the user
- one set of acceptance criteria
- usually follows: As a *<role>*, I *<action>* so that *<value>*

The image shows two sticky notes on a light blue background. The left sticky note has a red vertical border on its left side. It contains the following text:

Title: Order by pub date ID: 14

As a librarian
I want to see all books ordered by descending publication date
so that I can propose recent books to my customers

Estimate: 5 Priority: 3 (Medium)

The right sticky note has a red vertical border on its right side. It contains the following text:

Acceptance criteria

- Librarian may view all books.
- Librarian may view the details of the books, ie title, author, publication date, isbn, editor.
- Librarian may sort the books on their publication date.

Bill Wake's INVEST

independent stories should be implementable in any order

negotiable invitation to a conversation

valuable every story serves a purpose

estimable development time

small the holly KISS (keep it simple, stupid)

testable think early about acceptance criteria

The right level of details

Does it address **multiple goals**?

Does it comply with the **INVEST** rules?

Can we break it down further and **still get value** out of it?

Promoting **stories into epics** if there are

- **vague** or contain undefined terms
- conjunctions or **multiple use cases**
- **hidden** or unexplored **business rules**
- **multiple display** possibilities (devices or APIs)
- possible **exception flows**
- unidentified **data types** or **operations**

Time for switching your brain on

Team up with your neighbours (by 2 or 3)

- identify what the problem is
- propose a rephrasing
- you have 5 minutes

Row $\text{row mod } 3 = 0$

- As a customer, I want to pay for my cart with my VISA card or Paypal.

Row $\text{row mod } 3 = 1$

- As a client, I want to book the cheapest flight ticket possible.

Row $\text{row mod } 3 = 2$

- As a member, I want to receive offers and newsletters to keep me updated.



More examples of unclear stories

Vague or undefined terms

"As a customer, I can add multiple books to my shopping cart which is shown."

Problematic terms

multiple how many books?

shown where? what detail?

Derived questions

- how many items may be retained in the shopping cart?
- does the content of the shopping cart must be always shown?
- if so, how? With how much details?

Multiple use cases

"As a customer, I can search for a book and add it to my cart. "

Problematic terms

and at least two use cases

search for based on which criteria?

Derived questions

- what are the search criteria?
- how can I search for a book, *ie*, is there a search bar accessible everywhere?
- if so, are there advanced search facilities or filters?

Hidden business rules

"As a customer, I can add a book to my cart. "

Problematic terms

a customer any type of customer?

my cart retention policy?

Derived questions

- do I have to be logged in to add books to my cart or not?
- how long my cart's content is retained?
- what if the book is sold out?

Display possibilities

"As a user, I can view the content of my shopping cart. "

Problematic term

- view what type of devices are considered?
- view what type of format is considered?

Derived questions

- do we have to deal with *responsiveness*?
- do we have to build a dedicated mobile app?
- do we have to deal with exportation to separate documents?

Exception flows

"As a user I can log into my account so that I can view my cart. "

Problematic term

log into what are the needed details to log in?

Derived questions

- what happens after multiple failed attempts? How many?
- is there a "*lost password*" procedure?
- may users be blocked by administrators?

Data types

"As a customer, I can search for books so that I can find the publication I look for. "

Problematic terms

search what criteria?

find how are the results displayed?

Derived questions

- how are books defined in the system?
- how are results shown?
- does it exist additional filter possibilities?

Hidden operations

"As a customer I can review the items in my shopping cart and pay for them. "

Problematic terms

- review** what does "review" means exactly? May I suppress or update books?
- pay** what kind of payment procedure?

Derived questions

- what is the full workflow (*i.e* steps) between the review and payment?
- what are the involved third parties for the payment, if any?
- am I allowed to cancel the process at any time?

A soft-focus photograph of a young boy with brown hair, wearing a plaid pajama top. He is sleeping peacefully with his head resting on his hand, which is propped under his chin. His eyes are closed, and he has a gentle smile. The background is a bright, out-of-focus room.

Additional Reading

Weekly Reading (first of its name)

Why being a dark-room-geek is not the norm any more:

<https://dev.to/aspittel/the-most-important-non-programming-skills-for-programmers-iii>

In your 500-word report, I expect you to

- **summarise** one reading in 2/3 sentences, *i.e.* its context and addressed problem
- **reflect** on this reading
 - why did you select it, why is it relevant to you
 - what are your main **takeaways**
 - how will this **influence** you as a software engineer
- put into **relationship to this course** or your experience



**TO BE
CONTINUED... ➤**

Get dressed, we'll talk about ceremonies

How are story points and task estimated?

Let's play some (planning) poker?

How should team members be prepared for sprint review meetings?

Are sprint retrospectives kind of fights?

... come next lecture

That's all folks!