

Software Development Entrepreneurship

4 September, 2019

ENTREPRENEURSHIP..

What you thought-



VICTARE

What you got-



ENTREPRENEURFAIL.COM

Learning outcomes

At the end of this course, you will be able to:

- assess the merits, risks and approach of a business idea using a lean canvas
- develop a software product in a team using agile development
- identify relevant decision makers and approach them in the most promising way

Combining these facets, the goal is that you will also learn to:

- create a successful business by doing whatever is necessary to get there

Course setup

- **Today:** make teams
- **This week:** come up with business ideas
- ... work out your idea:
 - write up user stories
 - make a lean canvas
 - start on a business plan
- **2 October:** initial presentations (present your idea)
- ... start implementing, doing user research and approaching decision makers
- **27 November:** intermediate presentations (present your progress)
- ... complete your product
- **10 January:** final presentations (present your product)

Learning outcomes

At the end of this course, you will be able to:

- assess the merits, risks and approach of a business idea using a lean canvas
- develop a software product in a team using agile development
- identify relevant decision makers and approach them in the most promising way
- (create a successful business by doing whatever is necessary)

During this course, you will:

- **have to work with people from completely different backgrounds**
- **be taken thoroughly out of your comfort zone**
- **struggle and fall on your face multiple times**
- **power through it anyway :)**

FIRST DAY AT A STARTUP

WHAT?
NO
PARACHUTE?!



VICARE

ENTREPRENEURFAIL.COM

How you might think it works:

- come up with a good idea
- implement it
- profit

What actually happens if you try to do that

- project becomes increasingly complex, is never finished
- project is finally finished, turns out expected use does not model reality
- project is finally finished, but full of bugs
- time is wasted on infeasible or unimportant features
- market or technology has changed before project is complete

How it should work

- come up with a good idea
- determine the **absolute bare minimum** version of that
- implement a wireframe and show it to people, get feedback
- show it to decision makers, see about acquiring support
- implement bit by bit, always trying to have something working that you can show people
- realise that your plans have completely changed, just roll with it
- steadily improve design and add features
- have a **minimal working product** that could go to market
- get customers
- continue improving and building your customer base
- profit

Agile development: a key principle

Do not fight change.
Embrace it!

Another key principle

Deliver working software frequently.

The agile manifesto

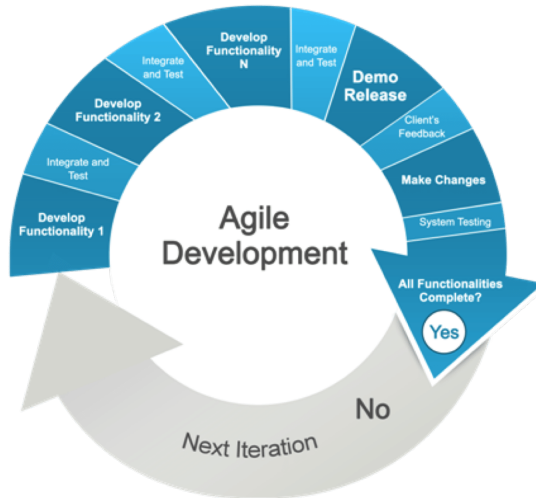
We are uncovering better ways of developing software by doing it and helping others do it.

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.

Agile development



Agile principles

Agile principles

Satisfy the customer through
early and continuous delivery
of working software.

Agile principles

Welcome changing requirements.
Harness change for the customer's
competitive advantage.

Agile principles

Business people and developers must
work together throughout the project.

Agile principles

Working software is the
primary measure of progress.

Agile principles

Simplicity is essential.
Do as little as possible.

Agile principles

Convey information to and within the team through face-to-face conversation.

Agile principles

Build projects around motivated individuals.
Give them space and support.

Agile principles

The best work emerges from
self-organising teams.

Agile principles

At regular intervals the team
reflects, then tunes and adjusts
their behaviour accordingly.

Agile principles

Development should be sustainable.
Maintain a constant pace.

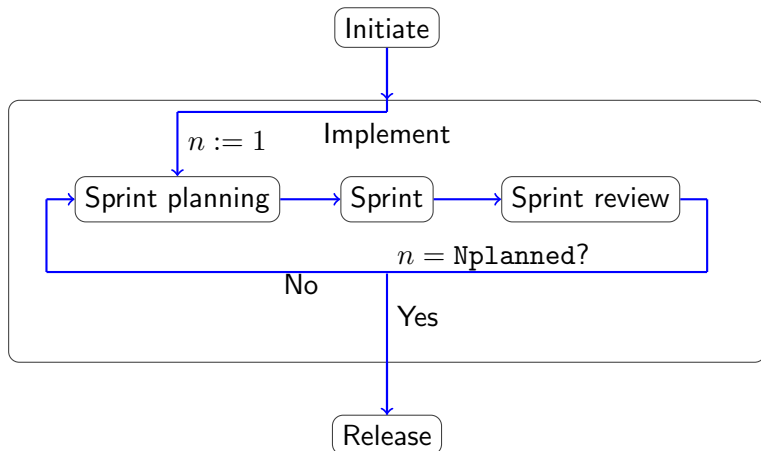
Core practices

- time-boxed iterations
 - between one and six weeks, set beforehand
 - no requirements may be added or changed during an iteration
 - requirements may be removed if it seems that the deadline will not be made
- incremental delivery
 - risk-driven and client-driven planning
 - get a working product ASAP, build from there
- communication
 - regular communication between customers and developers
 - evolutionary requirements analysis and adaptive planning using results of early work
 - constant in-team communication
- respect for programmers
 - teams have input on planning and organisation
 - maintain a sustainable pace

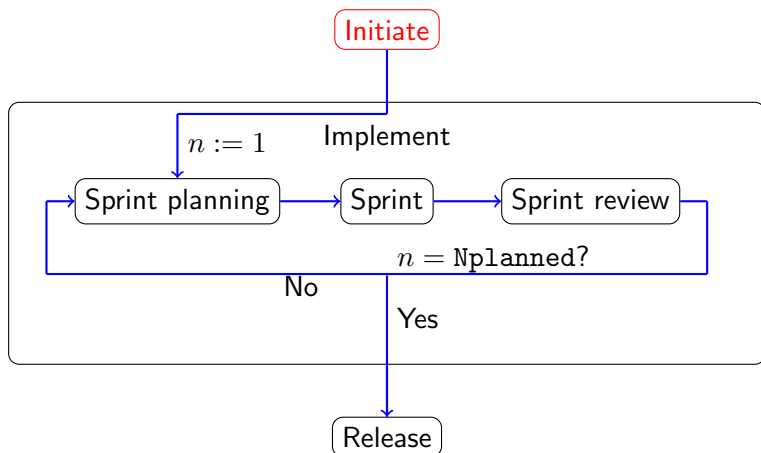
A common mistake

“Oh yes, we’re using agile methods! We’ve just finished the plan of what we’ll do in each iteration...”

The Scrum lifecycle



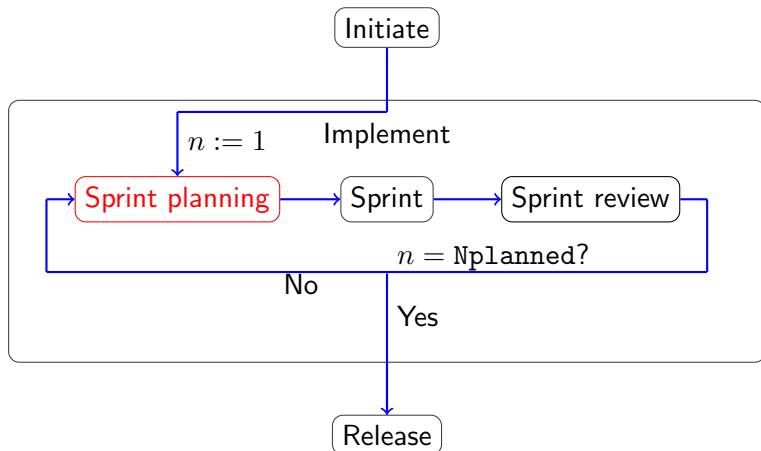
The Scrum lifecycle



Scrum Initiation

- create project vision
- create initial product backlog: prioritised requirement list
- create initial user stories
- designate Product Owner and Scrum Master
 - Product Owner: product visionary
 - Scrum Master: removing blocks, making decisions
- (assemble development team)

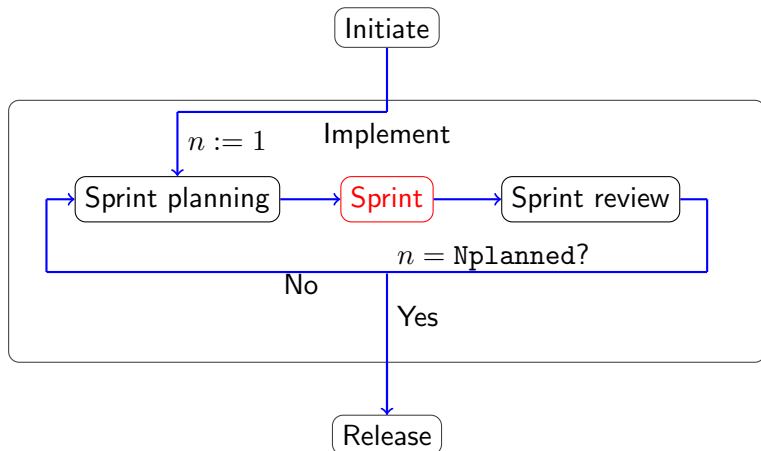
The Scrum lifecycle



Sprint Planning

- refine and reprioritise product backlog
- improve relevant user stories
- (re-)estimate times for features in product backlog
- create **Sprint backlog** of tasks (4–16 hours)
- estimated time may not exceed available time!

The Scrum lifecycle



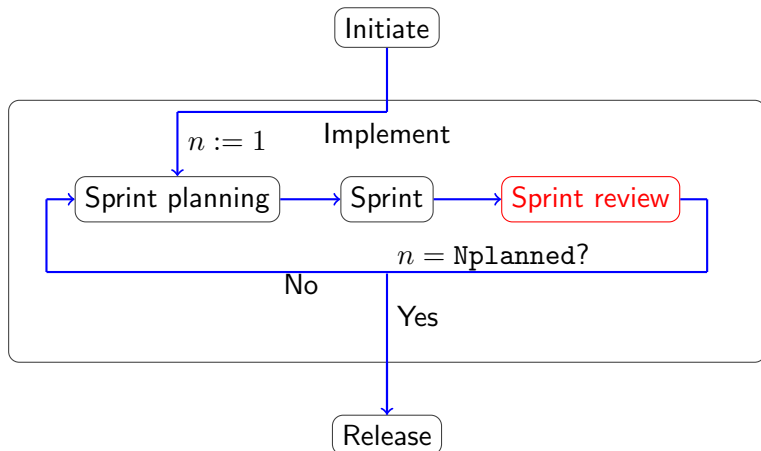
Scrum Sprint

- Usually 60 days.
- Work is done in a common project room.
- Daily Scrum meeting, attended by the full team.
- Daily build, integrating all code.
- Sprint backlog continuously updated.

Scrum Meeting

- Short meeting: about 2 minutes per person.
- Everyone answers the following questions:
 - What have you done since the last meeting?
 - What will you do before the next meeting?
 - What is getting in the way of meeting the Sprint goals?
- Reported blocks should be removed before the next meeting.
- Necessary decisions are taken in one hour.

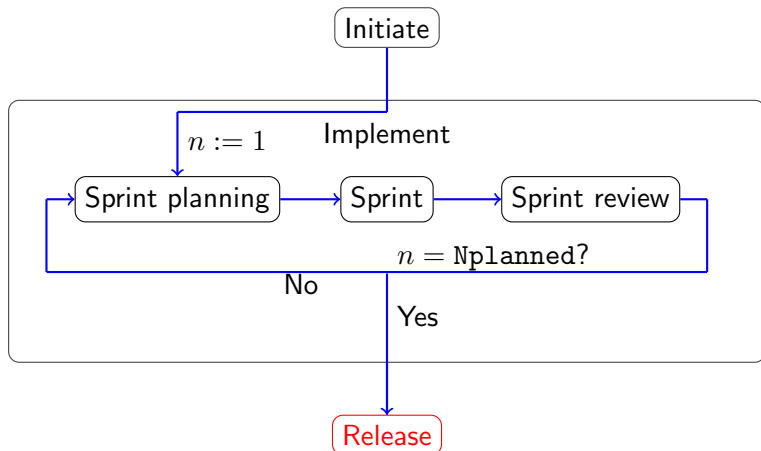
The Scrum lifecycle



Sprint review

- Meeting attended by the full team (and other stakeholders).
- Discuss what has been accomplished.
- Update customers / stakeholders, give a product demo.
- Brainstorming on directions, but no commitments.

The Scrum lifecycle



A different agile approach: eXtreme Programming

- emphasis on **oral communication**:
 - minimal requirements documentation; just story cards
 - everyone in one room, including at least one customer
 - pair programming with regularly mixed pairs
 - entire team responsible for all code
- evolutionary delivery through small, frequent releases
- test-driven development:
 - automatic acceptance tests for all features
 - unit tests for most code
 - first write the test, then write the code that makes it succeed
 - continuous integration on a dedicated machine that runs all tests
- frequent refactoring

An example

Goal: a website for publishing and reading stories

Vision: Provide an easy way for authors to make their stories available and earn money with them. Provide an easy way for readers to find and read good stories.

Initial product backlog

- uploading stories or chapters
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

- uploading stories or chapters
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

- uploading stories or chapters
 - As a **⟨role⟩** I can **⟨what⟩** so that **⟨why⟩**
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

- uploading stories or chapters
 - As an author I can upload a story from my computer so that it is in my account, to edit or publish.
 - As an author I can upload a single chapter from my computer so that it is in my account.
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

- creating accounts
 - As an author I can create an account so that my stories are saved under my name.
- uploading stories or chapters
 - As an author I can upload a story from my computer so that it is in my account, to edit or publish.
 - As an author I can upload a single chapter from my computer so that it is in my account.
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

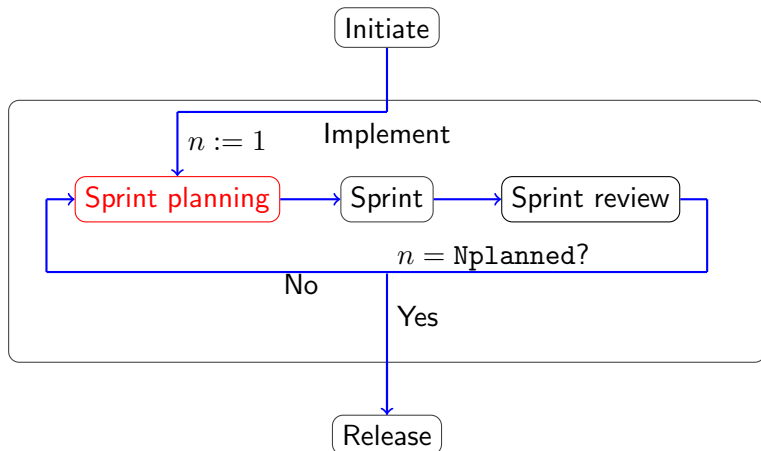
- creating accounts
 - As an author I can create an account so that my stories are saved under my name.
- uploading stories or chapters
 - As an author I can upload a word/latex/html document from my computer so that it is in my account.
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics

- creating accounts
 - As an author I can create an account so that my stories are saved under my name.
- uploading stories or chapters
 - As an author I can upload a word/latex/html document from my computer so that it is in my account.
- reading stories online
- downloading e-books
- searching for stories
- discussion forums per story
- a rating system on several characteristics
- user accounts
 - As an author I can edit the text of my story directly online, so that I can easily correct small mistakes.
 - As an author I can split and merge chapters so that I have full control over what to publish when.
 - As an author I can publish one or more chapters of a story, so that I can provide a web serial to my readers.

Prioritised product backlog

- minimal account creation 1 day
- minimal writing and editing chapters in the browser 1 week
- figuring out the legalities of copyright and donations 2 days
- publishing chapters and full stories 3 days
- searching for stories by keywords and genre 1 day
- rating a story on language, plot, humour, etc. 1 week
- filtering story search based on ratings, length, newness 1 day
- extended account creation 2 weeks
- donation button hooked up to paypal 1 week
- ...

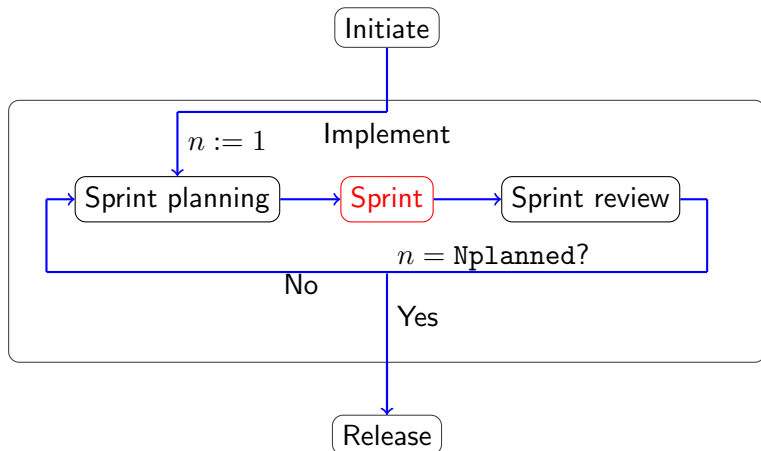
The Scrum lifecycle



Initial Sprint backlog

- designing and setting up the database 6 hours
- setting up the server 8 hours
- “create account” webpage with one textfield 1 hours
- googling for copyright laws 5 hours
- sending mails to my old friend who studied law 1 hours
- sending mails to whoever she recommends 5 hours
- story creation (overall data, without chapters) 4 hours
- chapter creation without any lay-out features 2 hours
- marking chapters in a story as published 5 hours
- presenting published chapter as HTML for publication 8 hours
- present story as a whole (with buttons, index) 3 hours
- ...

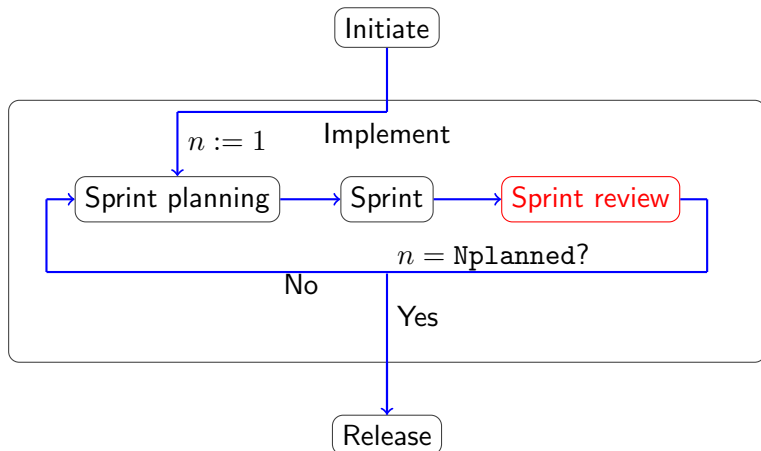
The Scrum lifecycle



In the meetings:

- **SM:** What have you done since last meeting?
- **P:** I have designed the database
- **SM:** What will you do before the next meeting?
- **P:** Implementing the database, and putting some default users in
- **SM:** What is getting in the way of meeting the Sprint goals?
- **P:** The database is different from what I learned. I may need some help in figuring out the right SQL queries.
- **SM:** Okay, let's update the planned time. How long do you think this will take you?
- ...
- **Overall:** discussion possible, but try to stay below 2–3 minutes each
- Scrum Master is responsible for keeping the meeting on track!

The Scrum lifecycle



For next week

- Discuss two or three ideas for a software product to base your start-up around.
 - Who are your likely customers?
 - Why would they use it?
 - Is there a similar product on the market?
 - Do you have a unique advantage that your competitors do not have?
 - How will your idea make money? (Short-term? Long-term?)
 - Can you implement this in the given time? (Preferably: well before!)
- Look into the resources that you will need.
 - What language will you use?
 - Are there services (e.g., Google Maps API) that you can use?
 - Are you familiar enough with them to succeed in time?