**Agile development:**

Structure of the pages and complexity

* task: 6 weeks to redesign Ubiqum web design
  + tech project: the waterfall approach, 93% fail rate
  + planning, design, execution, testing and release
  + to deliver this website
  + map the amount time in order to deliver the website in the next 6 weeks?
  + planning: roles assigned, ideas, mock design, data frame, language, max 3 days
  + designs: mock design’s,  1.5 weeks max
  + execution: frontend, 1 week
  + testing -> 2.5 weeks
  + release -> 1 week

problems with the waterfall approach:

* time delays filters throughout the project
* project manager designing the allocated tasks
* look at the final date and reverse engineer it

Agile:

-sprints and iterative

* 6 sprints (loops) have some integration of these phases
* identify specific goals
* Build the Home page, get feedback- Sprint 1
* Bounce rate? How to reduce bounce rate, get people to stay
* Second option to do first: solve the most complicated process first
* How do we hit deadlines?
* who defines what gets done? Goals from the stakeholders, product owner, collaborative activity is sprint planning
* 6 week sprints: how long the tasks will take? Scrum master manages it
* How to be more secure about providing estimates? Reflection, cross-check (points poker)
* **break down of tasks: 1-2 hours**
* **contingency plans** in cases where breaking down tasks takes longer

Sprint planning- the tool used to track a sprint? Trello

* Stand up meetings: before and currently, accountability
* iterative process: constant improvement/continuous improvement via retrospectives
* retrospectives: after every sprint, focussed, find consensus
* Projection in small smart ways: what can I rely to project?
  + **Velocity: points for each sprint (75%)**
  + the no of tasks and period based on velocity, a rough sense based on teams velocity
  + figuring out the priority of the tasks and obtain  more resources: time, work and resources
  + time and work are adjusted in order to meet the fixed deadline

**Hours vs Points:** shared understanding that’s not subjective

* points poker: a bunch of cards, fibonacci sequence
  + decomposing tasks
  + fibonacci sequence: 3 and 21 (points poker)
  + the idea of points is the shared understanding, younger developer vs a senior developer
  + variability comes with points (26 points and 40 points)
  + burn down rate: 90 point team, sprint 1, 2, 3, 4, 5, 6
  + putting measures in place to improve things
  + trello is simple to use
  + we welcome last minute changes
  + **user story: I want blank,** to do something/in order to blank, user centric, users goals and interests
  + **persona:  user demographic**

**User stories:** student (potential, current and former), company/employer, competitor, employee (vacancies) == E**xternal users**

* **internal:** product owner, the CEO, the different ways to track how things are used, marketing person (conversion is key)
* **linga franca:** trade language- common used language