**Agile Development by Nathan Benjamin**

**Task: 6 weeks to redesign Ubiqum web design**

* tech project: the waterfall approach has a fail rate of 93%
* planning, design, execution, testing and release – 5 phases 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 consists of sprints and are 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 on a website
* 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 usually takes 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 number 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 based on 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 by way of decomposing tasks
  + similar to the Fibonacci sequence, i.e. 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 calculated from the number of sprint’s 1, 2, 3, 4, 5, 6
  + putting measures in place to improve things, i.e. 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, age ect.**

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

* **internal users:** product owner, the CEO, the different ways to track how things are used, marketing person (conversion is key)