

Week 3

1. Waterfall Model
 - Model doesn't work with a project that has lots of changes
2. Customer Chaos Model
 - Too many customers asking devs for stuff
 - Loudest voice wins
3. Assumptions of *empirical methods*
 - Can't repeat process from input to output due to variables and noise (unpredictable)
 - Management consists of constantly monitoring and adapting
4. Scrum process
 - Prioritizes certain user stories to be implemented
 - Sprint planning
 - Sprint backlog (what team wants to implement)
 - Sprint execution (1-2 weeks implementing feature)
 - Daily scrum (meeting to report progress)
 - Potentially shippable product increment (what was done in the sprint)
 - Sprint review (team + product owner + customer + stakeholders look at output of sprint)
 - Sprint Retrospective (take feedback from review and see how to improve)
5. Scrum Roles
 - Stakeholders
 - Internal stakeholders (CEOs etc.)
 - Customers/users (must organize user stories by priority)
 - Product owner (Represents customers)
 - Scrum Team
 - Scrum master (helps dev team with scrum practices and helps the product owner)
 - Dev team (3-10 members in one location, chooses their own roles [not boss])
 - Product owner (Must "OK" the products before release)
6. Examples
 - Small Start-up
 - One scrum team for one project
 - Roles:
 - * Product owner
 - * Dev team
 - * Scrum master
 - Mid-sized
 - Many scrum teams with indep. projects
 - Roles:
 - * Stakeholders
 - * Product owner
 - * Dev team
 - * Scrum masters
 - Large multi-national
 - Many scrum teams divided on one large project
 - Roles:
 - * Stakeholders
 - * Product owner
 - * Dev team
 - * Scrum master
7. Backlogs
 - Product backlogs (stakeholders + dev team can add to this)
 - User stories should be DEEP (detailed, emergent, estimated, prioritized)

- Product owners prioritize items
 - Devs estimate items
- Release backlogs (product owners add to this because they represent the customers)
 - Subset of product backlog
- Sprint backlogs (features to be completed for next sprint)
 - Subset of release backlog