**Lecture Part 1 🡪 Agile Production**

Agile vs. Waterfall

* Waterfall
  + Older form of production
  + Big plan upfront, long execution
  + Good for less experienced teams
  + Less responsive to changing requirements
* Agile
  + Subject of this lecture

The Agile Manifesto

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* **Responding to change over following a plan**

Kinds of agile development

* Extreme programming
* Kanban
* SCRUM

Heart of Scrum: Rapid Iteration

* Weekly sprint
* Cut scope instead of slipping the deadline
* Sprints should align with desk critique
* Never have a broken build by Thursday
* Post Sprint PHYSICAL meeting 🡪 part of the process log 🡪 retrospective
* Evaluate how things went
* Do it again
* **Take detailed notes from discussions (Ross Brown should be able to ask to see these somewhere)**

Scrum in game development

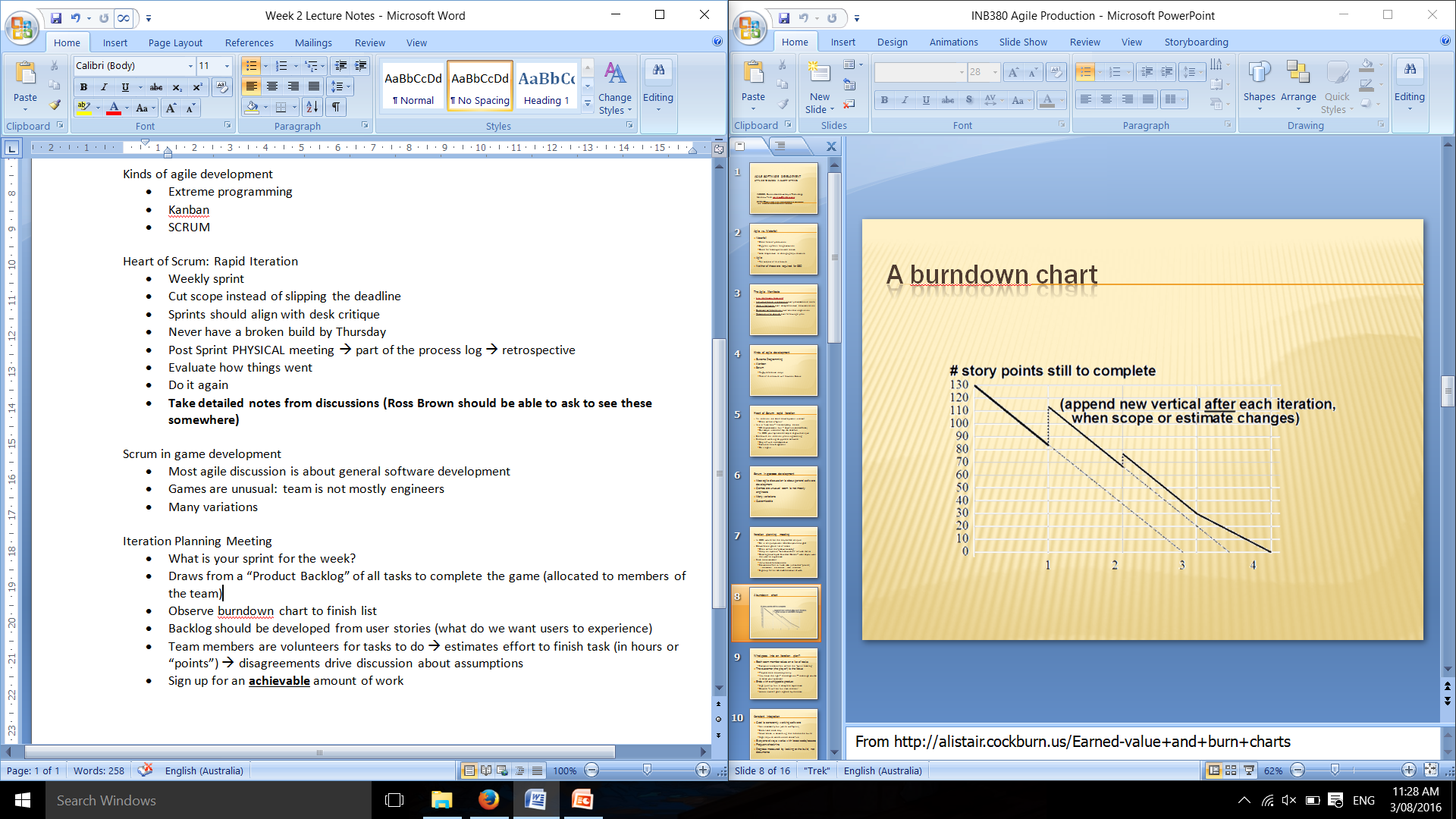
* Most agile discussion is about general software development
* Games are unusual: team is not mostly engineers
* Many variations

Iteration Planning Meeting

* What is your sprint for the week?
* Draws from a “Product Backlog” of all tasks to complete the game (allocated to members of the team)
* Observe burndown chart to finish list
* Backlog should be developed from user stories (what do we want users to experience)
* Team members are volunteers for tasks to do 🡪 estimates effort to finish task (in hours or “points”) 🡪 disagreements drive discussion about assumptions
* Sign up for an **achievable** amount of work

Make a burndown chart (producer)

* Make in excel?
* Consists of Product Backlog tasks (weekly backlog until testing, then try a whole project burndown chart)
* Tasks over time (weeks)



What goes into an iteration plan? (Backlog)

* Each team member takes on a list of tasks
* Customer is the focus
  + Playtest data decides priority
  + Change course to serve your customer
* Ends with a shippable product
  + High quality, fun, complete experience
  + Beware “It will be fun *next* iteration”
  + Has an overall goal agreed by the team (constantly working software)

Constant Integration

* Goal is constantly working software
  + Doesn’t need to be fun (while mid-sprint)
  + Stabilised each day
  + Never check in something that breaks the build
  + Might require coordinated check-ins
* Everyone always works with latest code/assets
* Frequent check-ins
* Progress measured by looking at the build, not documents

Stand Up Meeting

* Morning of each development day, fixed time and place
* Whole team attends
* Very very short and focused only on
  + **What I did yesterday**
  + **What got in the way of doing work (or will get in the way)**
  + **What will I do today**
* Scrum master (producer)
  + Steers process very firmly
  + Records and removes all teams impediments (what’s stopping something being done?)
* After this, everyone works without further interruptions

Customer Focus

* Player = customer
* Playtest data is **crucial**
  + Comprehensive
  + Clear
  + Timely
* Game should become MORE enjoyable, useable and engaging

Change is good

* Reduce scope to meet the goal
* as end of sprint approaches, team reduces the scope to meet the iteration goal
  + player data dictates the priority
  + reduce scope, not quality
* optimise
  + team members make deal in stand-up meetings
  + what you cut is just as important as what you make

Retrospective

Do day before critique

* what went well?
* What are our strengths?
* What need improvement?
* What will you do differently?
* Report retrospective in process log with a few sentences (dot points are fine, good even but have a sentence or two beforehand to “make if feel more human”)

Scrum at Halfbrick (Nasa procedures are similar to Scrum)

* One week sprint
* Planning 1 hour
* Retrospective and review 2 hours
* Use big whiteboard
* Use trello with scrum plugin
* Cheat by subtracting points from part-done tasks
* No interruptions policy

Tools for Agile Production

* Trello
  + Used by halfbrick
  + “Scrum for Trello” plugin for chrome 🡪 <http://www.civicactions.com/blog/2012/oct/10/five_tips_for_using_trello_for_scrum>

**Lecture Part 2 🡪 SVN Planning**

Brain as Planner

* Implementation intention – writing verb noun pair on average doubles productivity by priming and providing actions against distractions “write down what you need to do” 🡪 what am I going to do with what at what time where?
* Forming a habit of planning becomes unconscious and more common
* Planning primes the brain to relax subconsciously
* Some planning is better than none

Think about what works 🡪 track outputs to see what works under analysis

Weekly Plans

* Insight into meeting milestones
* **FOUR WEEKS**, including this one, to build your game
* Should know characters, mechanics and other assets going into the game
* Use whatever tools you like but we must see a plan
* **Spreadsheet of content ready by week 3**
* Support the plan of the team