

Project Management

A case study of Agile Development

Søren Lundgaard
soren.lundgaard@gmail.com

My Background

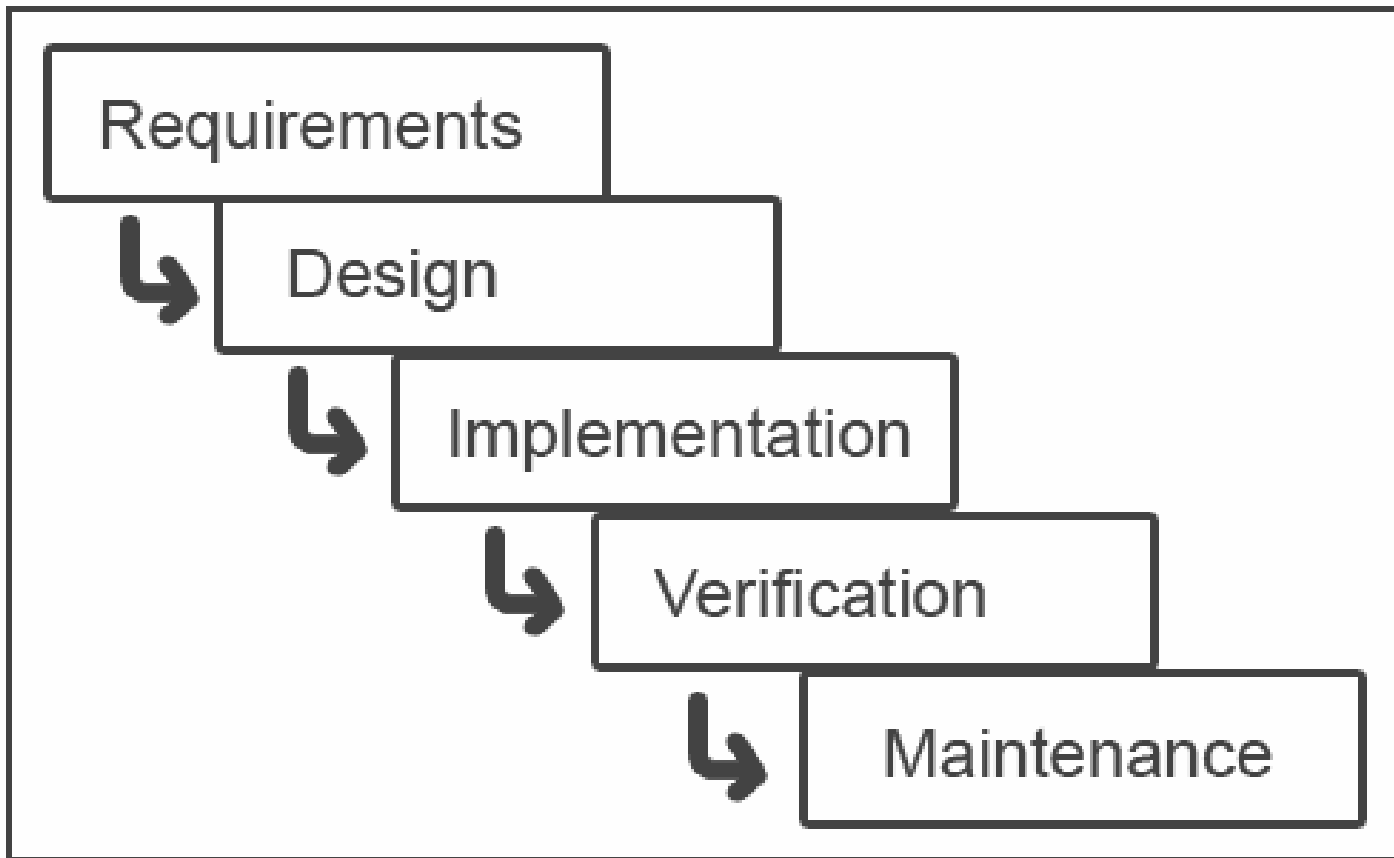
- Worked for Deadline Games 8+ years
- Lead Programmer
- Game Director
- Now half time at DADIU
- And freelance

The Case Study

- Case: “Faith and a .45” demo, 2007
- Deadline Games
 - stuck in traditional game development method
 - preparing to grow
 - Ideas didn’t flourish
 - Commercial goal: survive on own ideas

Traditional Game Development

- Waterfall Method



Waterfall

- Strictly staged development
- Define milestones
 - when a milestone is met you move to next stage
- Workers are organized in a hierarchy
 - Craft groups (code, graphics, sound, animation, etc) have Leads and they delegate tasks

Waterfall, Positive

- If you have a good definition of you end-product then the Waterfall method will deliver (and deliver on time)
- It's use of Milestones makes it a good tool for upper-management and external people
- Workers on lower levels need only to know their own craft

Waterfall, Negative

- When a Change is introduced the whole Milestone schedule has to be reworked
 - Change-Management is way to handle this, but the problem is that it assumes that you can guesstimate the impact of the Change
- Developing new ideas and tech is hard since craft groups are in principle isolated
 - Ideas comes from above and down

Back to Deadline's situation

- We were working on the our first big AAA title for the current consoles (360, PS3)
- It was planned to take 20 months to develop
- We realized that the Waterfall method restrained us from having something playable in any foreseeable future
 - Looking at the plan it told us we had to wait 8 months before we knew if the design was fun or not!
 - That was simply not acceptable!

GDC to the rescue

- Luck would have it that we at this point were sent to GDC in San Francisco
- GDC (Game Developers Conference) is held several times a year around the globe
- The conference in San Francisco is the largest with 10.000+ attendees

What we learned at GDC

- The last 2-3 years talks about Agile Development (and specifically SCRUM) where gaining interest
- Each company had their own approach, but general consensus was that the industry needed to change development methods
- And Agile seemed to solve many of our problems

... some examples

High Moon Studios

- A relatively new developer who tries to sell on itself on the use of SCRUM
- Developed the game Darkwatch using SCRUM
- The talk was about how they worked and how they had modified SCRUM to work in a game developing environment

Valve Corporation

- Valve created Half-life 2 (and their current titles) using what they called a Cabal
- In general they set up a prototype-test-driven environment
- The Cabal consist of very skilled people
- The development has no final deadline
 - “When its done”
- (Epic used a similar method for Gears of War)

EA Montreal

- Used a prototype-driven environment (for Army of Two)
- Small groups of people where tasked to create prototypes within a small timeframe using whatever technology
- Prototypes had to be something workable on the screen
- When the game development started it was more traditional (Waterfall-like), but the prototyping helped to focus on the fun stuff

Back at Deadline

- Returning from yet another GDC which showcased an Agile approach as something all the big players looked at, we decided to give it a go
- But the upper-management where not convinced of the need and the consequence
- So we could only introduce some Agile and decided to try out what we called Task Forces

Task Force

- Basically like SCRUM, but less formalized
- Small number of people (2-6) across different crafts
- Runs in Sprints (2-4 weeks)
- Goal for Task Force is defined by Product Owner
 - Overall Product Owner was the Game Director
 - Some teams had a Product Owner Proxy on the team
 - Task Force subdivides goal into smaller tasks and starts working
- Each week the Task Force gives an external status in production group meeting by showing on screen where they are

Some Task Force Examples

- “Character AI going into Cover”
- “Picture in Picture”
- “Engine Stress Tests”
- “Long coats on enemies”
- “Sniper rifle”
- “Carry weapons on back of characters”

Task Force Iterations

- We synced all Task Forces to each other
 - They had the same number of weeks (2)
 - This was necessary in order to reassign people between sprints
- But since not all Task Force could reach their goals in 2 weeks we ended up working with iterations
 - So after each sprint it was decided whether or not give a Task Force a new iteration
- Most of the time this was anticipated and we in general planned out 2-3 sprints in advance

Result

- A demo was developed in 11 weeks
 - 4 x 2 weeks of regular Task Forces
 - 3 x 1 week of finish-up stuff (mostly bug-fix and optimizing)
- We were 25 people on the project + 10 people in engine department
- Each sprint had approx. 10 simultaneous task forces
- We delivered on-time with little need for crunch and the result surpassed the expectations

We did we learn?

- Animators and Artist were in general less prepared for the new Agile approach
 - They preferred the Waterfall (assembly line) method
- Lead roles were broken down and most Leads felt powerless in regard to controlling their craft teams
 - We didn't anticipate that they still felt responsible

Post Mortem, Sprints, Positive

- They generate good results and tools with little overtime or stress
- They generate fast feedback
- Good team buy in
- Fast turnaround times
- Gives focus
- Feeling of progress
- Good for tracking progress
- Small size works best
- The better cross-group communication generates enthusiasm, respect and understanding.
- Sprints avoid old time bureaucracy

Post Mortem, Sprints, Negative

- There hasn't been enough time to make mistakes
- Not enough opportunity for open brainstorming sessions between sprints / phases.
- Working in taskforces generated more bugs
- Not enough use of QA
- Too many taskforces and too many people in each taskforce; over-allocation of people
- Not enough cross competence in the taskforce composition
- Responsibility and definition of roles are unclear.
- Too many limbo periods; we need a bigger backlog
- No option of long term planning due to short sprints
- Product owner shouldn't be part of the taskforce.

Consequence

- After production everybody knew what Agile meant
- The demo proved that Agile was a viable method
- It was also evident that we had to use it with care and that we would have to change the role and responsibilities for the Leads
- Today Deadline Games is still in a transition towards a more Agile approach

Questions?

Søren Lundgaard
soren.lundgaard@gmail.com