Agile

What is Agile

- A popular methodology used in software development
- Developed in response to some of the problems that arise from other project management processes
- Scrum
- An implementation of Agile
- Let’s look at some of the issues that might arise from managing projects in the ‘traditional’ way

The Waterfall Methodology

1. Gather and document requirements
2. Design
3. Build with unit and integration testing
4. Perform system testing
5. Perform user acceptance testing (UAT)
6. Fix any issues
7. Deliver the finished product

Drawbacks

- Gathering & documenting requirements, often very difficult
- Doesn’t allow for changing requirements or deeper understandings
- Can lead to large losses in time
- Making changes can be difficult once the product is built
- Has been blamed for many a failed, expensive government project (Capita)
- Agile tries to solve some of these problems by catching inaccuracies of requirements and any changes earlier in the development process

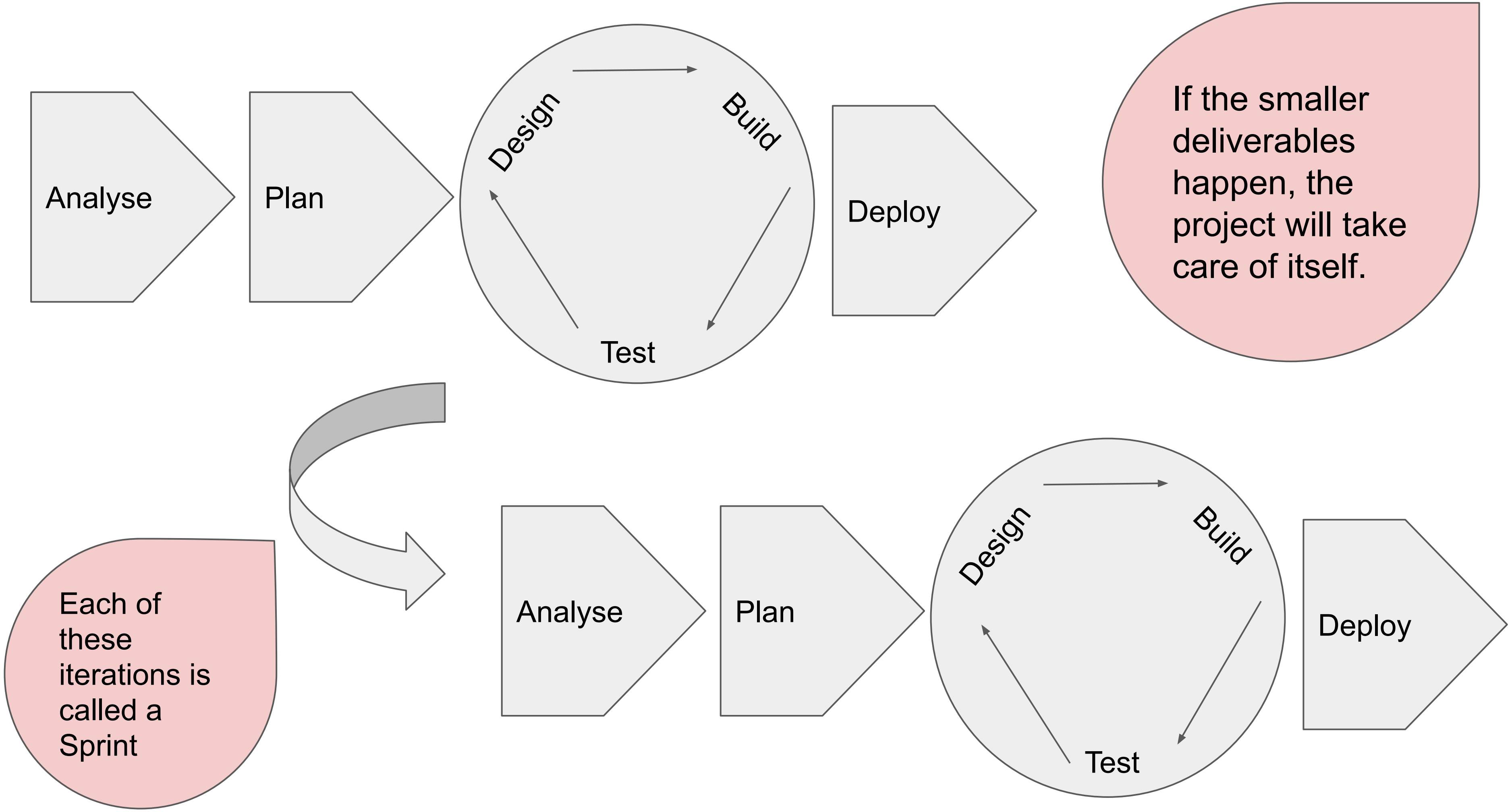
Agile

Manifesto for Agile Software Development

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

Agile recognises that some work, like software development, needed a different approach.

Software development is a bit like an experiment (try this, then try something else) and it’s **hard to plan a process of discovery**.



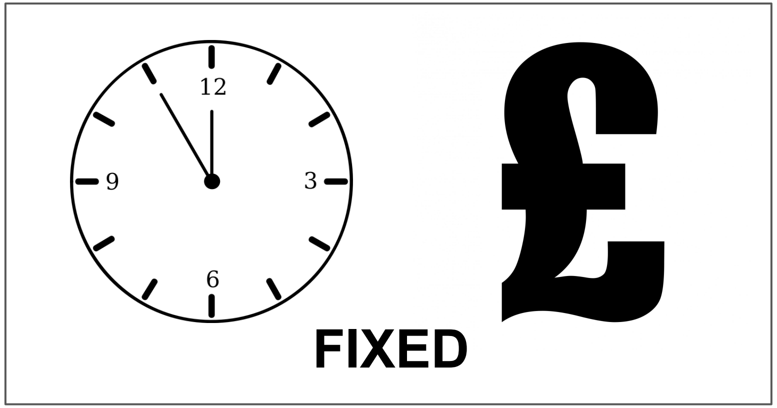
If the smaller deliverables happen, the project will take care of itself.

Each of these iterations is called a Sprint

Time, Cost and Scope

- Projects are constrained by 3 things: **Time**, **Cost** and **Scope**
- Usually fixed
- **Agile makes scope flexible**

PROJECT 1



Within these constraints, deliver what is most valuable

Scrum

- Leading Agile method
- ‘Scrum’ is taken from rugby, analogy used in an important paper in HBR (Takeuchi and Nonaka)
- In rugby, scrums are (meant to be) high-performing & cross-functional formations, that restart at certain points of the game

Scrum Team

Minimum of three specific roles:

What needs to be done

- The Product Owner
  - Champions for their product
  - Focussed on understanding business and market requirements
  - Prioritising the backlog
- Scrum Master
  - Champions for scrum within their team
  - Coach
  - Looks for improvement in practice
- Dev Team
  - Drives the plan for each sprint

How the project is done

Scrum Framework

- The team must meet every day: the **Stand Up**
  - What are we doing today? Does anyone need to be caught up? What are our blockers?
- 2 - 4 week **Sprints** are considered best
- Dedicated team (no multi-tasking teams)
- 5 - 9 members
- Have a product vision from which you can extract an **MVP** (Minimum Viable Product)
- **Team Norms** (agreements on how to work together)
- Sprints finish with a **Retrospective**

What does this mean for group projects?

- You should start every working day with a stand up
- Agree some team norms: e.g. how will you resolve conflict?
  - Agree to disagree, but move forward
  - Our priority is learning
- Plan your days like a sprint
  - 6 hours, £0, what is most valuable to do today?
  - Finish each day with a short retrospective
- Create an MVP and Extensions
- Prioritise your backlog
- You must have a working MVP before going to Extensions

Tools of the Trade

- Kanban board: Trello - trello.com
- Retrospectives (end of projects): Metro Retro - metroretro.io