Software Lifecycle model

We chose the agile development process for this project as it is the most flexible of all the software development choices.

While making the decision we looked at more rigid processes such as the “waterfall model” and the “v-model” but decided that the potential for the project would be much too high. We would prefer a short time between initial project time investment and either failing fast or knowing that an approach will work - So we quickly established our needs as a development team:

* Working software to be delivered frequently.
* Close, daily cooperation between developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements need to be implemented

It was clear that the agile development process was best suited to our needs. Always having a working product, starting with the very first sprint, so that it was harder for the project to go wrong. It seemed to fit as well because a typical scrum team is usually between 5 and 9 people – and as we are a team of 5 that suits us perfectly.

Instead of big specs, we’d discuss requirements together in meetings. Instead of wondering what anyone is doing in our project, we’d collaborate around a task-board (we used a free website called “Trello”) discussing progress. Instead of long project plans, we’d discuss what’s right for the project as we go and the [team is empowered](http://www.allaboutagile.com/2007/03/agile-principle-2-agile-development.html) to make decisions. In turn this helps to create highly motivated, high performance teams that are [highly cooperative](http://www.allaboutagile.com/2007/04/agile-principle-10-no-place-for-snipers.html).

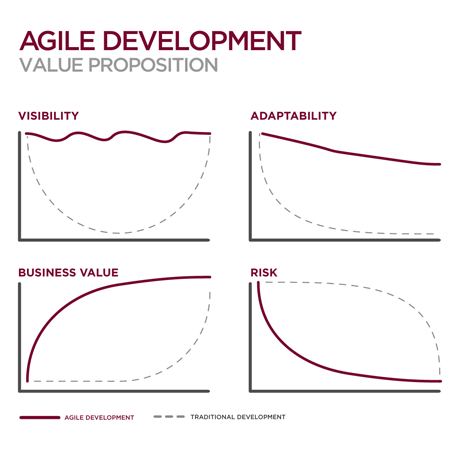
A key principle of agile development is that [testing is integrated throughout the lifecycle](http://www.allaboutagile.com/2007/04/agile-development-agile-testing-is-not.html), enabling regular inspection of the working product as it develops. This this will allow us to make adjustments if necessary and gives the product team early sight of any quality issues.

Unlike the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/) in agile model very limited [planning](http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/) is required to get started with the project. Agile assumes that the end users’ needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.

On agile projects, every member of the project team has the opportunity to know how the project is going at any given time. Daily scrum meetings and weekly sprint reviews offer concrete ways to see progress.

By delivering working, tested, deployable software on an incremental basis, agile development delivers increased value, visibility, and adaptability much earlier in the life cycle, significantly reducing project risk.

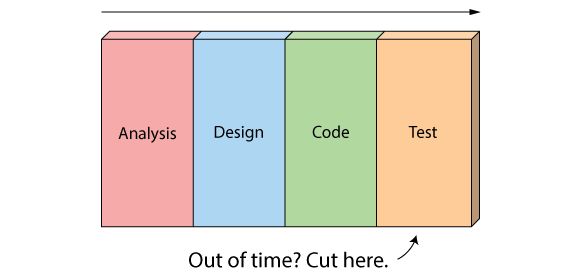
### **Agile Development Value Proposition**



# Agile vs Waterfall

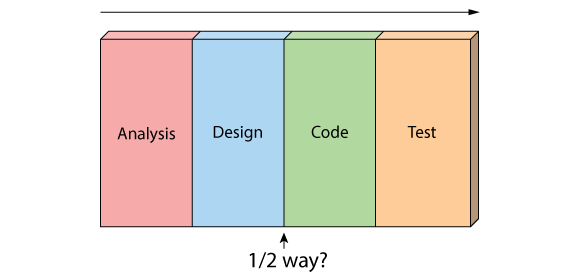
### Waterfall challenges

### Poor quality



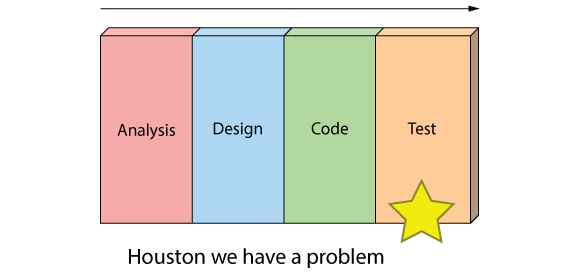
First off, when the project starts to run out of time, testing is the only phase left.

### Poor visibility



Secondly, because working software isn't produced until the end of the project, you never really know where you are on a Waterfall project. That last 20% of the project always seems to take 80% of the time.

### Too risky

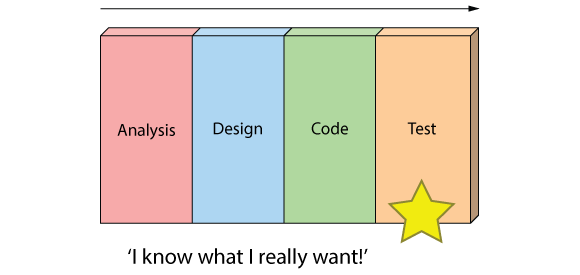


Thirdly you've got schedule risk because you never know if you are going to make it until the end.

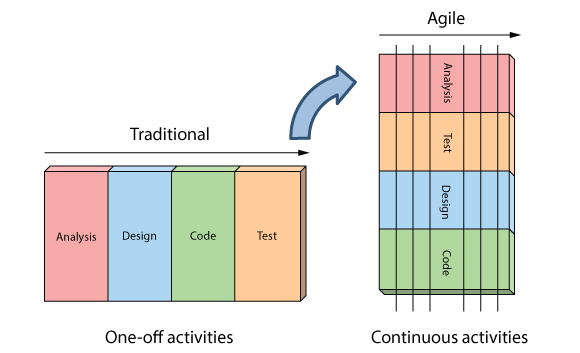
You've got technical risk because you don't actually get to test your design or architecture until late in the project.

And you've got product risk because don't even know if you are building the right until it's too late to make any changes.

And finally, most importantly, it's just not a great way for handling change.



## The Agile Approach



Resources:

<http://www.dummies.com/how-to/content/ten-benefits-of-agile-project-management.html>

<http://epmlive.com/5-benefits-of-agile/>

<https://cs.anu.edu.au/courses/comp3120/public_docs/WP_PM_AdvantagesofAgile.pdf>

<http://www.seguetech.com/blog/2013/04/12/8-benefits-of-agile-software-development>

<https://www.versionone.com/agile-101/agile-software-development-benefits/>

<http://www.allaboutagile.com/10-good-reasons-to-do-agile-development/>

<http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/>