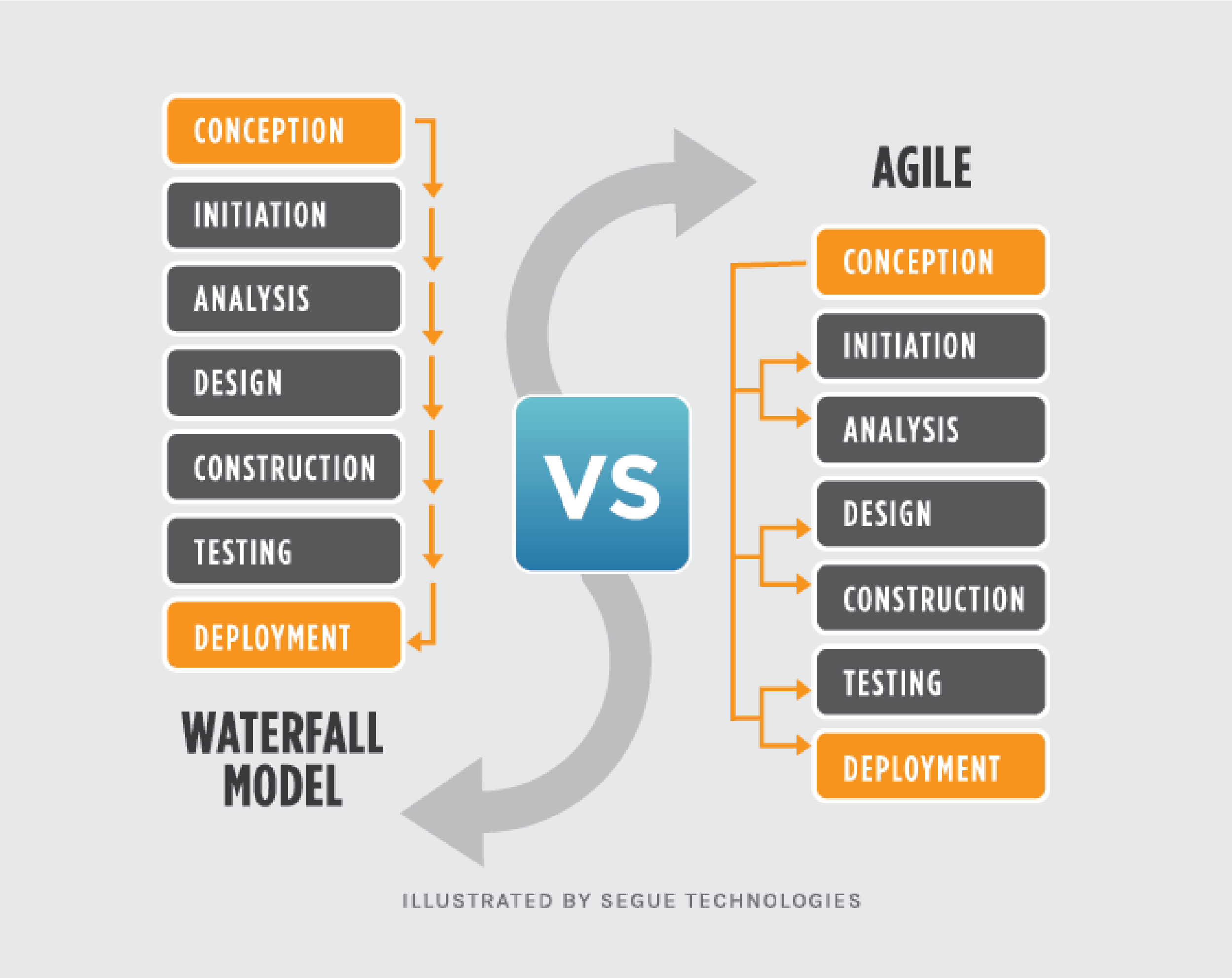
SYNTAX TECHNOLOGIES

SDLC Day 2

Syntax Technologies Inc. All Rights reserved
www.syntaxtechs.com

Waterfall & Agile Methodologies



ILLUSTRATED BY SEGUE TECHNOLOGIES

Waterfall

Waterfall

- Like Construction, waterfall methodology is a **sequential** design process.
- Once a step has been completed, developers cant go back to a previous step – unless they scratch the whole project and start from the beginning.
- No room for error, so project outcome & plan **MUST** be determined before starting it.

When to use Waterfall

- When there is a **clear** picture of what the final product should be.

- When **Quality/Definition** is a key to success not speed.

- When the client **wont change** the scope of the product once the project has begun.

Advantages of Waterfall

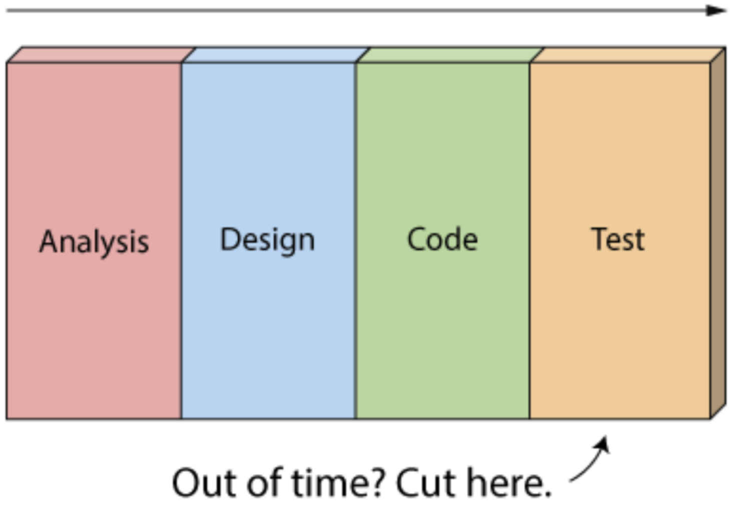
- Easy to **understand**
  - This model is simple and easy to understand and use
- Easy to **manage**
  - each phase has specific deliverables and a review process
- Suitable for **simple or smaller projects**
  - Waterfall model works well for smaller projects where requirements are very well understood.
- **Step by step process**
  - The phases are processed and completed one at a time. Phases do not overlap.

Disadvantages of Waterfall

- **Step by step process**
  - Once a step is completed, developers cannot go back to a previous step and make changes
- Clear **initial requirement**
  - Waterfall relies heavily on initial requirements. The project is doomed if requirements are not clear.
- Does **not allow** for requirement change
  - If a requirement error is found or change needs to be made, the whole project has to restart from beginning.
- No working product **until near completion**
  - All the steps/stages have to be independent completed before the final product is delivered.

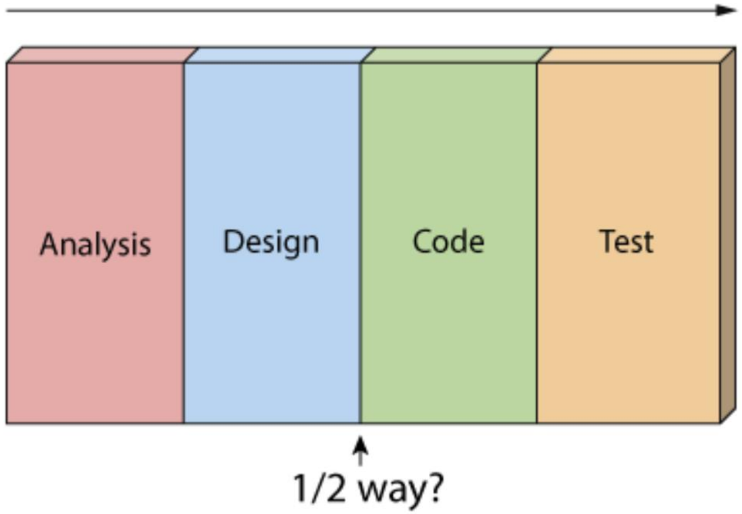
Poor Quality

First off, When the project starts to run out of time and money, testing is the only phase left. This means good projects are forced to cut testing short and quality suffers.



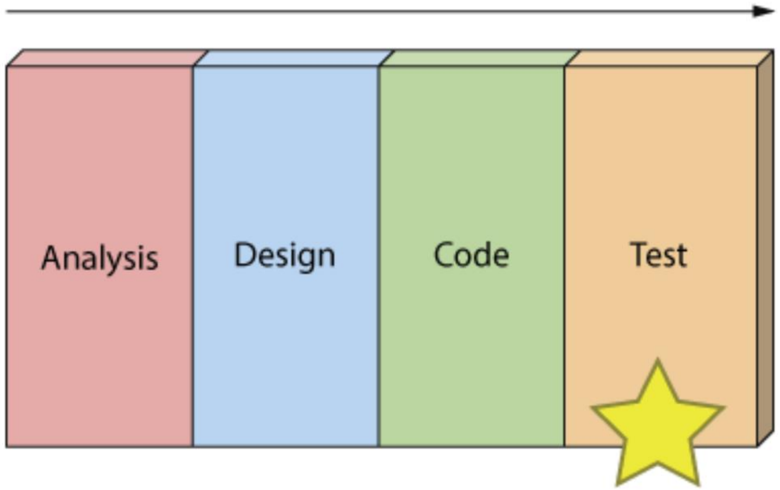
Poor Visibility

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.



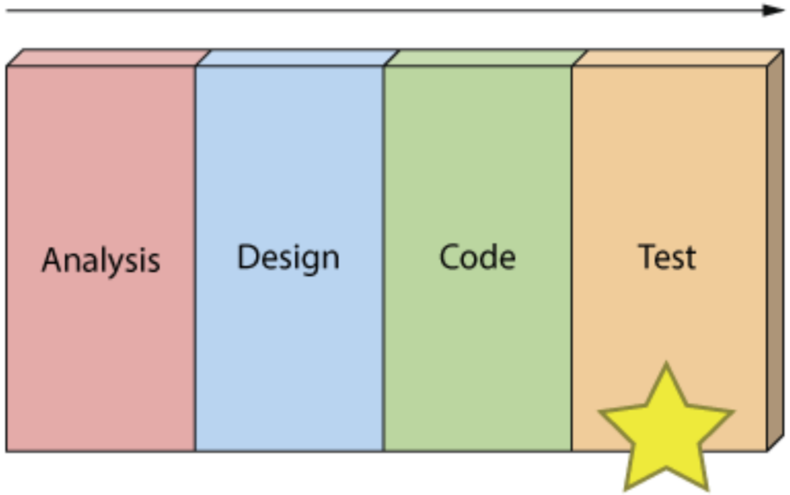
Too Risky

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 cha



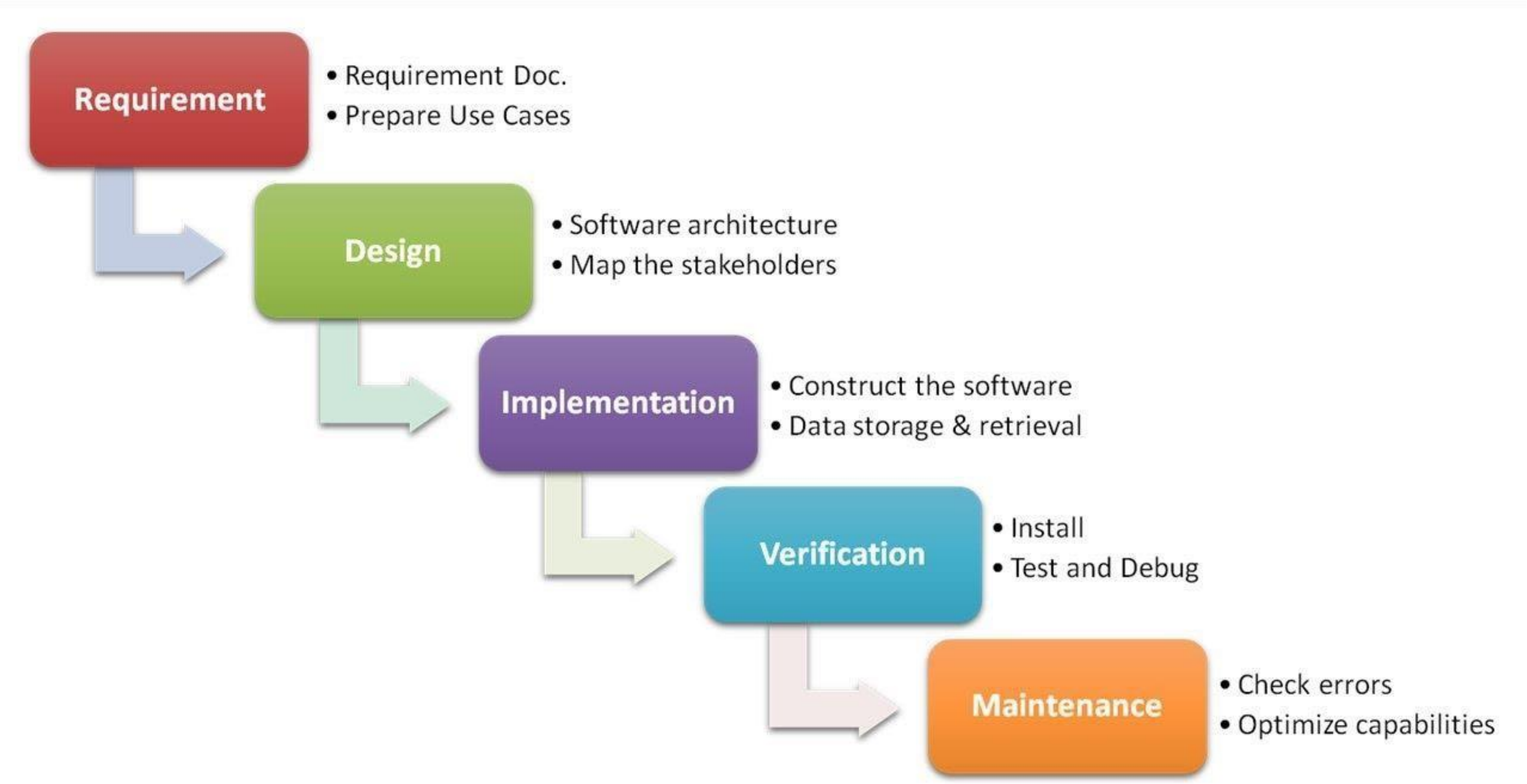
Can't handle Change

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



'I know what I really want!'

Waterfall methodology



Requirement
- Requirement Doc.
- Prepare Use Cases

Design
- Software architecture
- Map the stakeholders

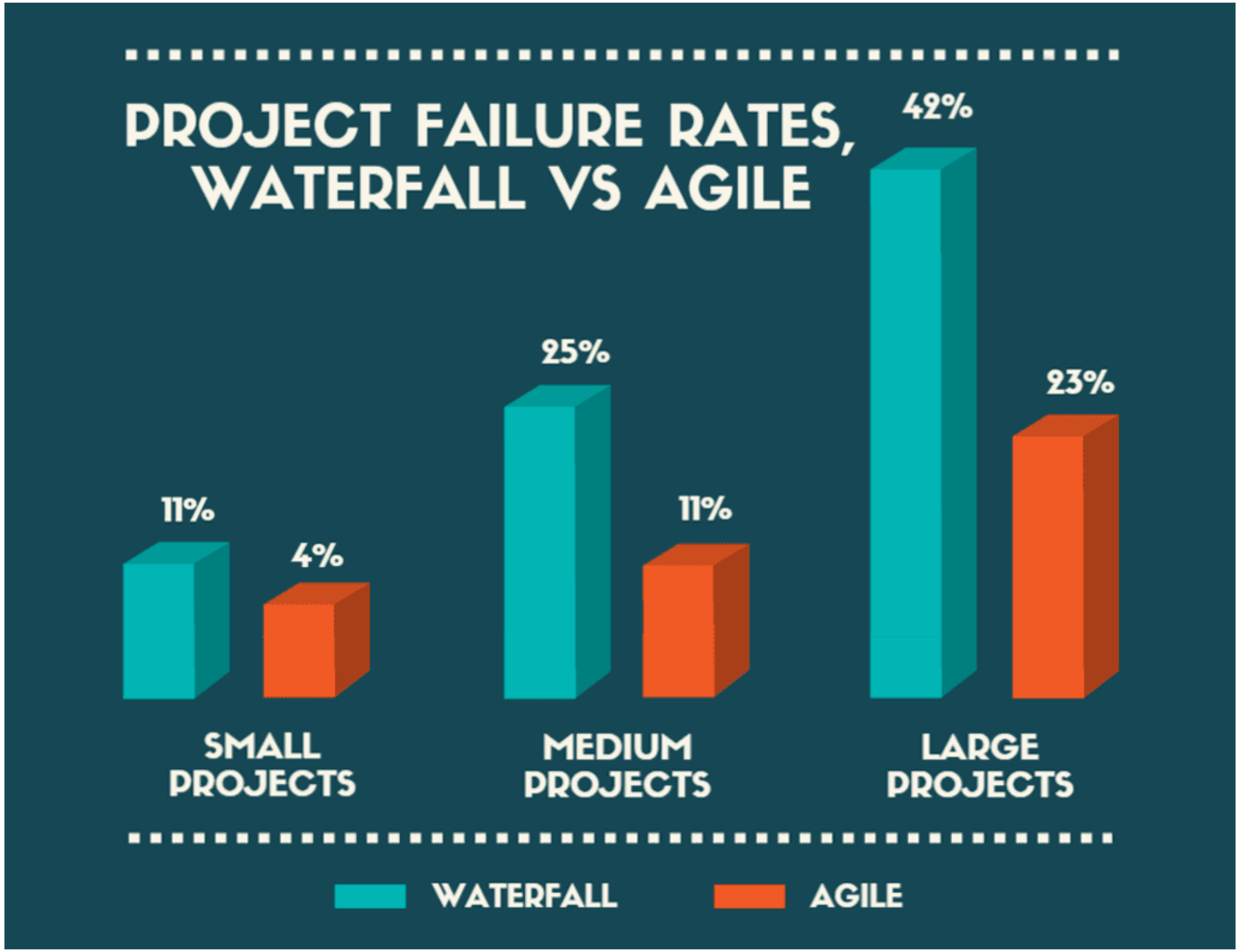
Implementation
- Construct the software
- Data storage & retrieval

Verification
- Install
- Test and Debug

Maintenance
- Check errors
- Optimize capabilities

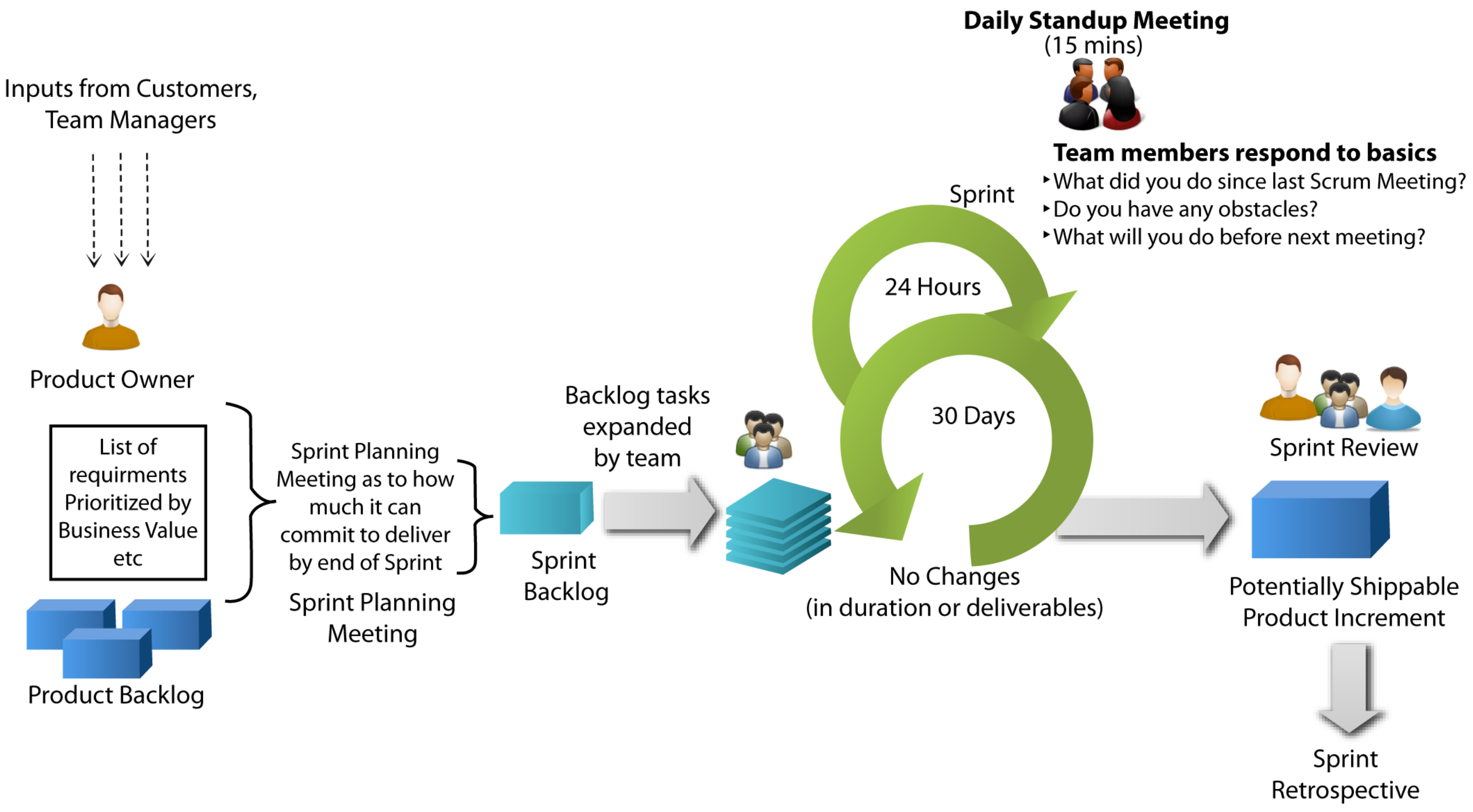
Example

Statistics says:



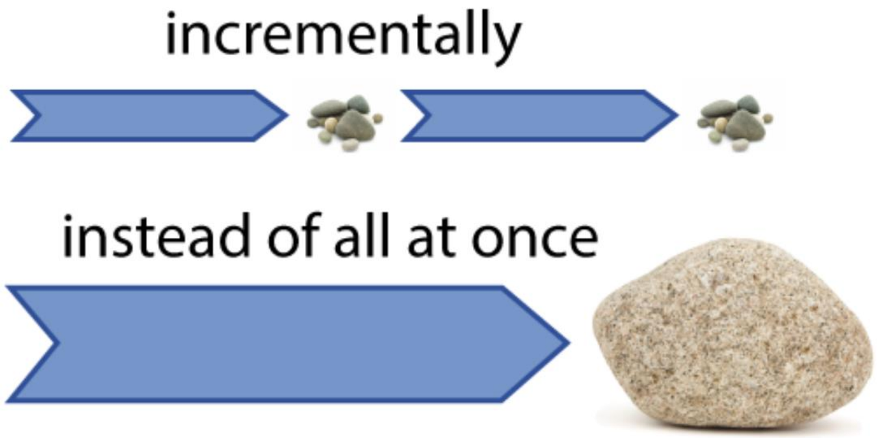
Agile

Agile Scrum Methodology


Daily Standup Meeting
(15 mins)
Team members respond to basics
What did you do since last Scrum Meeting?
Do you have any obstacles?
What will you do before next meeting?
Inputs from Customers, Team Managers
Product Owner
List of requirments Prioritized by Business Value etc
Product Backlog
Sprint Planning Meeting as to how much it can commit to deliver by end of Sprint
Sprint Planning Meeting
Sprint Backlog
Backlog tasks expanded by team
Sprint
24 Hours
30 Days
No Changes
(in duration or deliverables)
Sprint Review
Potentially Shippable Product Increment
Sprint Retrospective

Agile

- Definition:
  - Able to move **quickly** & **easily.**
- IT Definition:
  - Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.



Agile

- Agile came as a “solution” to the disadvantages of Waterfall. Instead of a **sequential** process, Agile follows an **incremental** approach.
- The Team starts off with a project design, and then begin to work on small modules. The modules is done in **weekly** or **monthly sprints**, and the end of each sprint, the project priorities are evaluated and tests and run.
- These sprints allow for bugs to be discovered & fixed, and customer feedback is incorporated before starting next sprint run.It works by breaking projects down into little bits of user functionality called **user stories**, prioritizing them, and then continuously delivering them in short cycles.

When to use Agile

- When there **isn't** a clear picture of what the final product should be.
- When **rapid production** is **more** important than the **quality** of the product.
- When clients **will** be able to **change** the scope of the product.
- When product is intended for an industry with **rapid changing** standards.

Advantages of Agile

- **Allows for changes**
  - Agile allows for changes to be made after initial planning. Changes can be made per client request.
- End of **sprint evaluation**
  - At the end of each sprint, project priorities are evaluated. The client can add their feedback for the next sprint to get their desired product.
- Testing **completed** at each sprint
  - Testing at the end of each sprint ensures that the bugs/defects are caught and taken care.
- Release product at **end of cycle**
  - Product is launched at the end of each cycle. All the testing and development is completed before going into production

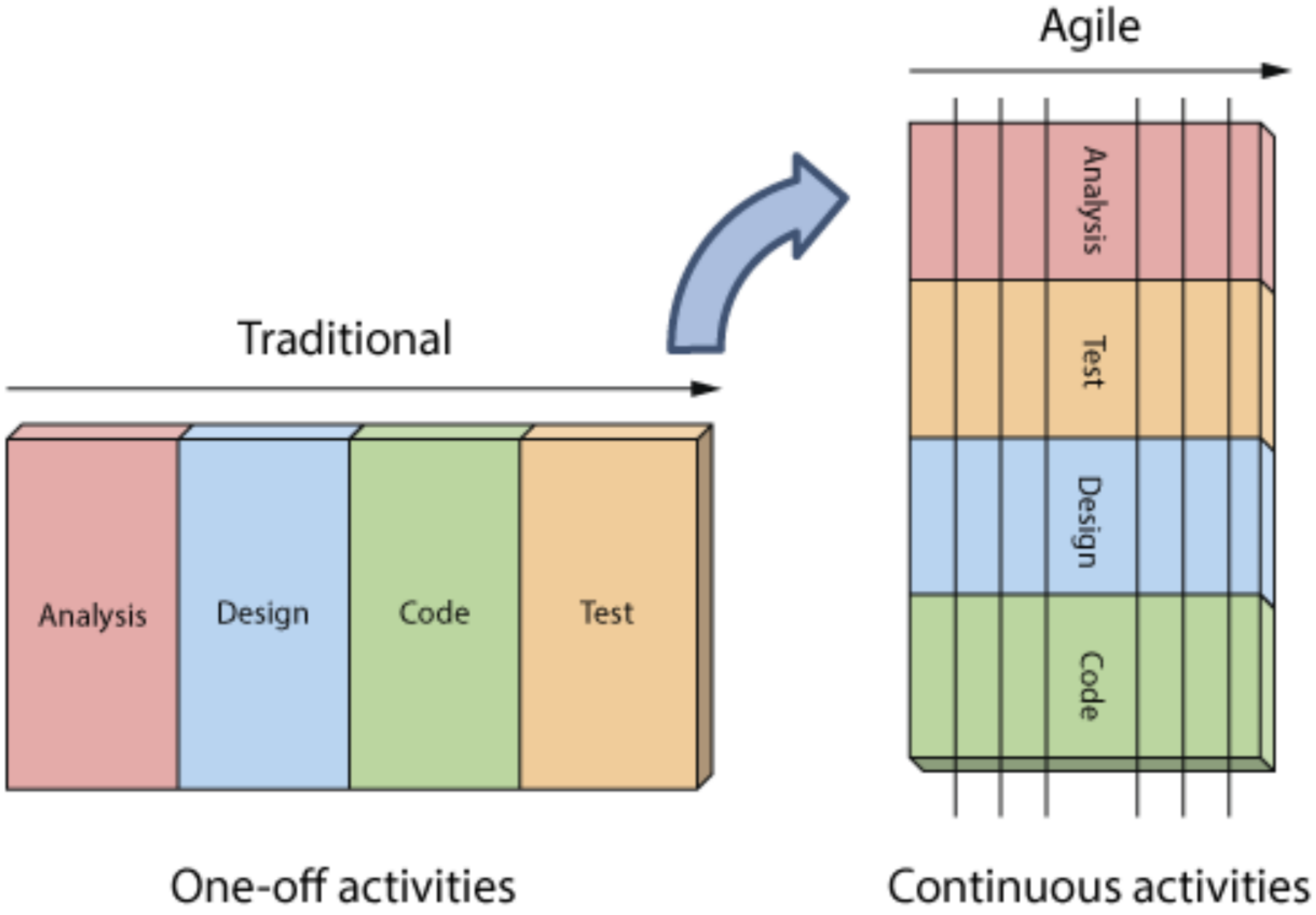
Advantages of Agile

- **People & interactions** are emphasized more than **processes & tools**.
- **Face-to-Face** conversations.
- Regular **adaptation to changing** circumstances.
- Late **changes** in requirements are **welcomed**.

Disadvantages of Agile

- Less importance given to **Design and Documentation.**
- Unclear client requirements leads to a **messing project.**
- Bigger & complex project, difficult to determine the **efforts estimation** at beginning of the project.

Agile VS Waterfall



Which is better???

Which is better???

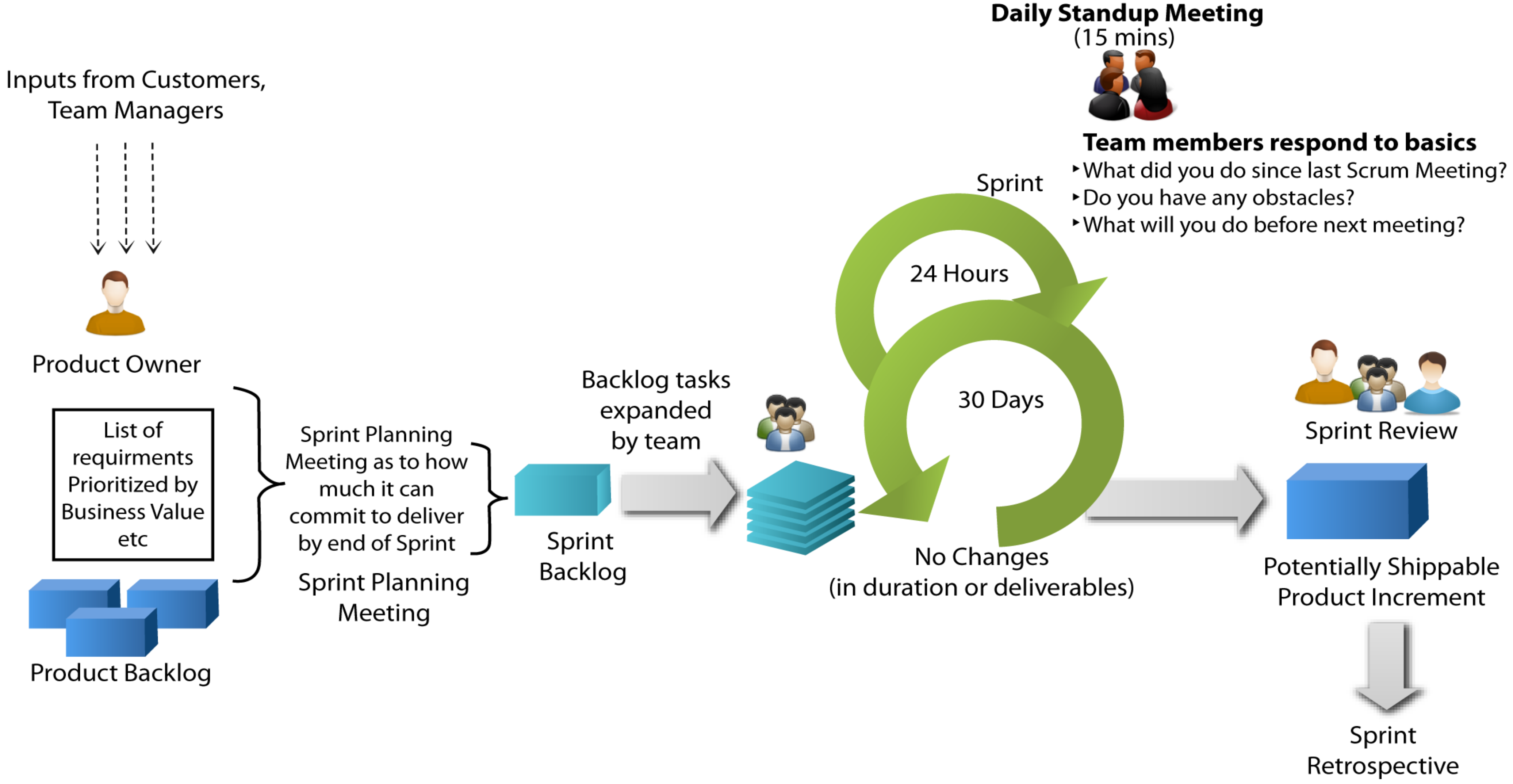
- The key to deciding which methodology is right comes down to the context of the project.
    - If it is going to **change rapidly** – Choose **Agile**.
    - If you know what **exactly you need** – Choose **Waterfall**.

Syntax Technologies Inc. All Rights reserved

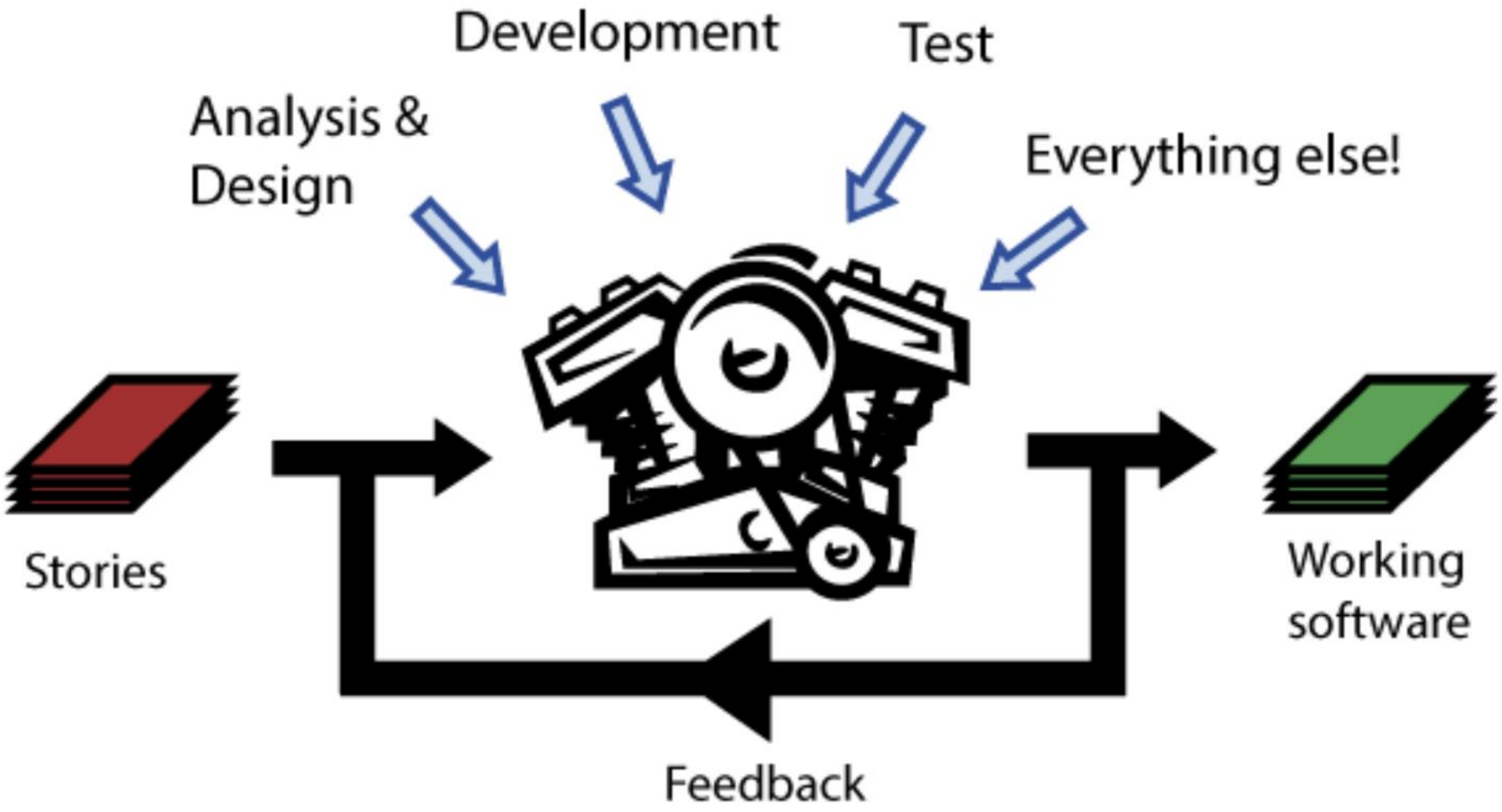
Agile Methods & Frameworks

- **Scrum (70 %)**
- Kanban (5 %)
- Scrumban (7 %)
- Custom Hybrid (8 %)
- Scrum/XP Hybrid (10 %)

Agile Scrum Methodology



Inputs from Customers, Team Managers
Product Owner
List of requirments Prioritized by Business Value etc
Product Backlog
Sprint Planning Meeting as to how much it can commit to deliver by end of Sprint
Sprint Planning Meeting
Sprint Backlog
Backlog tasks expanded by team
Daily Standup Meeting (15 mins)
Team members respond to basics
‣What did you do since last Scrum Meeting?
‣Do you have any obstacles?
‣What will you do before next meeting?
Sprint
24 Hours
30 Days
No Changes (in duration or deliverables)
Sprint Review
Potentially Shippable Product Increment
Sprint Retrospective



Syntax Technologies Inc. All Rights reserved

Scrum Framework

- Scrum is an Agile framework for completing complex projects.
- It is applied to any project or product development effort.

Roles

Product Owner
Scrum Master
Team

Artifacts

Product Backlog
Sprint Backlog
Burn-down Charts

Ceremonies

Sprint Planning
Sprint Review
Sprint Retrospective
Daily Scrum Meeting

Syntax Technologies Inc. All Rights reserved
www.syntaxtechs.com

Roles

Product Owner

- Defines the **items** in the **Product Backlog**
- Manages project features & release to optimize ROI.
- **Prioritizes** features according to user & stockholders needs.
- Can change feature and priority every sprint.
- Accepts or reject work from the Development team.

Roles

Scrum Master

- Ensures Scrum is fully functional, productive & improves quality.
- Ensures Scrum is understood & enacted.
- **Shields** the team from external interferences.
- Communicates to Product Owner & Team
- **Does not** Make decisions for the team.

Roles

Development Team

- **Cross-functional**
- Six +/- 3 members
- **Selects** the sprint goals & specific work results.
- Commits to what it feels it can accomplish.
- Self organized, manages itself & Sprint Backlog.
- **Demonstrate** Sprint results to Product Owner.