Contents

[Software development process models 2](#_Toc489285167)

[Waterfall (1790 Winston Royce) 2](#_Toc489285168)

[V-Model (1986 Paul E Brook) 3](#_Toc489285169)

[Agile 4](#_Toc489285170)

[Agile Software Development Methodologies 4](#_Toc489285171)

[Agile Diagram 5](#_Toc489285172)

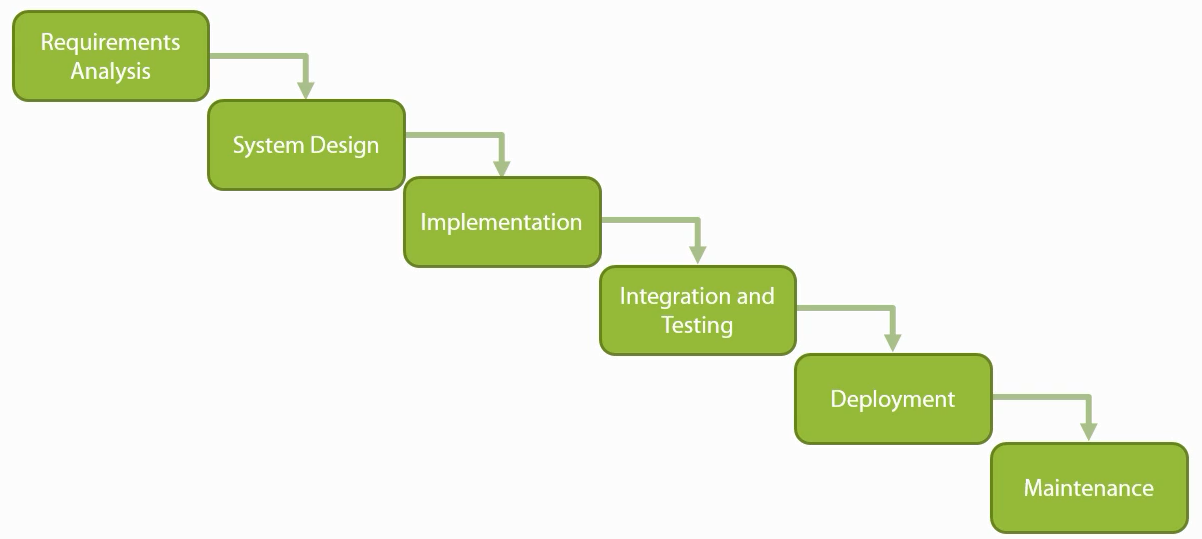
[Agile Manifesto 5](#_Toc489285173)

[Agilge Manifesto Principles 6](#_Toc489285174)

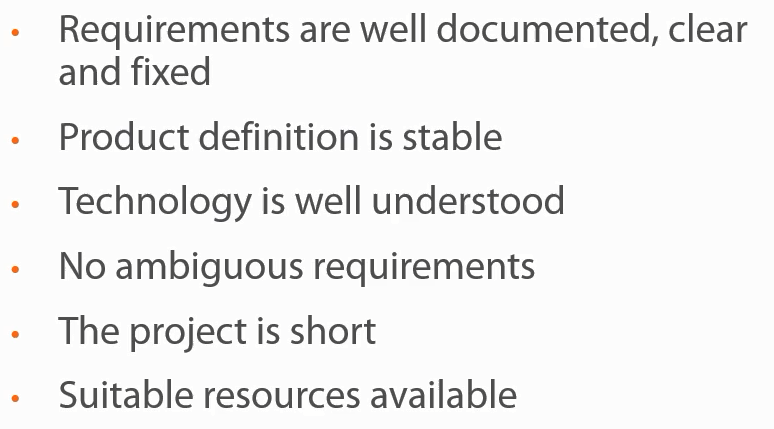
[Scrum 7](#_Toc489285175)

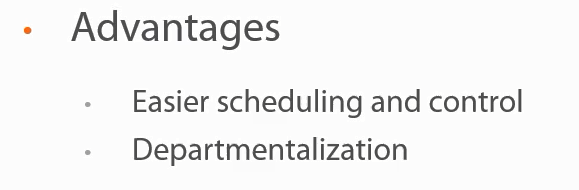
# Software development process models

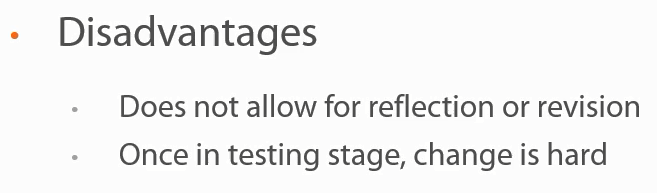
## Waterfall (1790 Winston Royce)

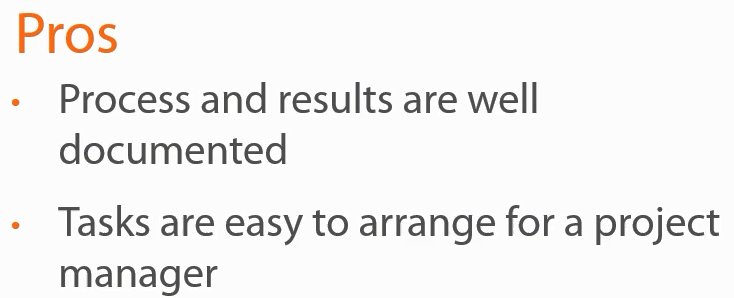


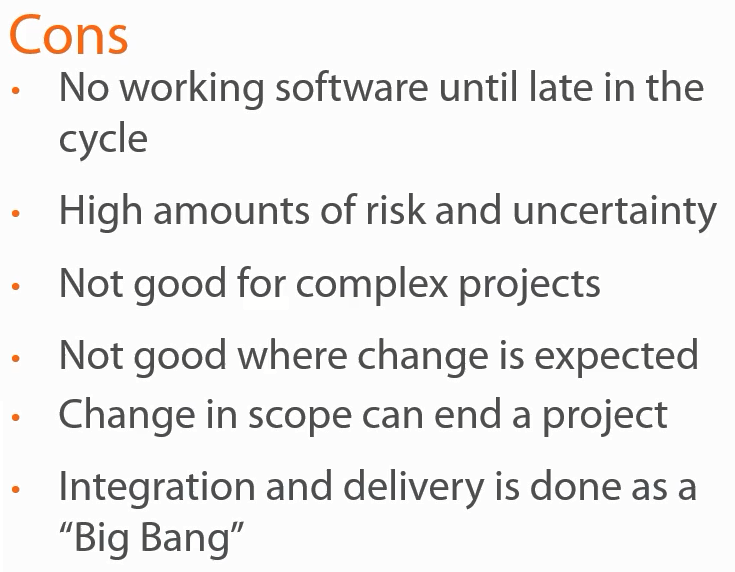
Suitable when:



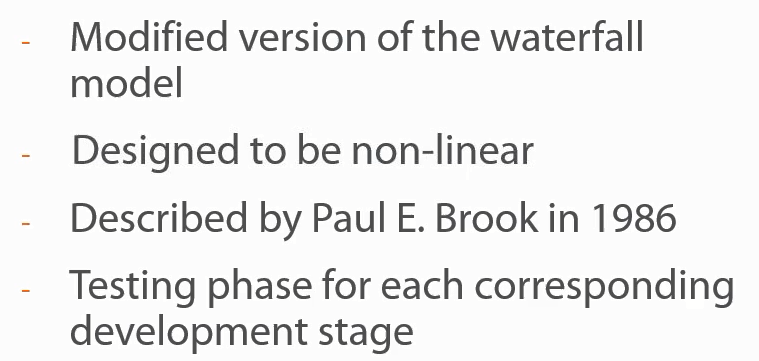


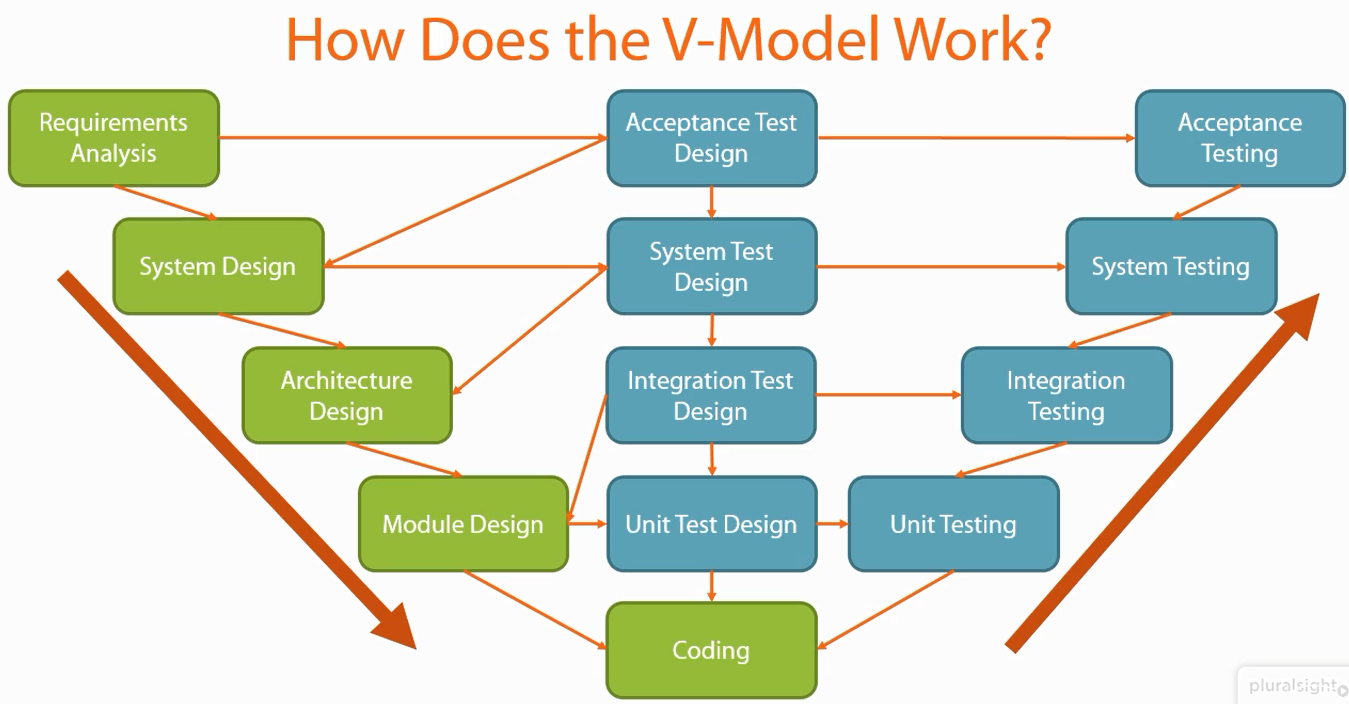




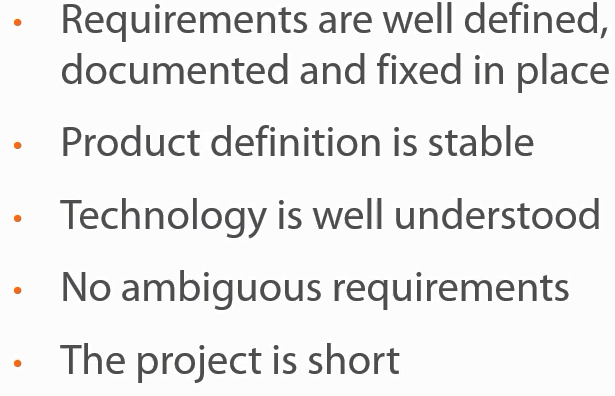


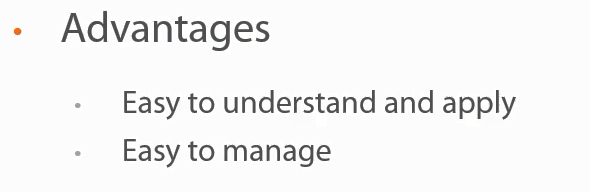
## V-Model (1986 Paul E Brook)

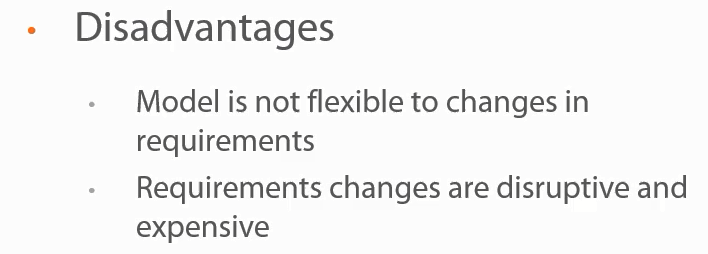


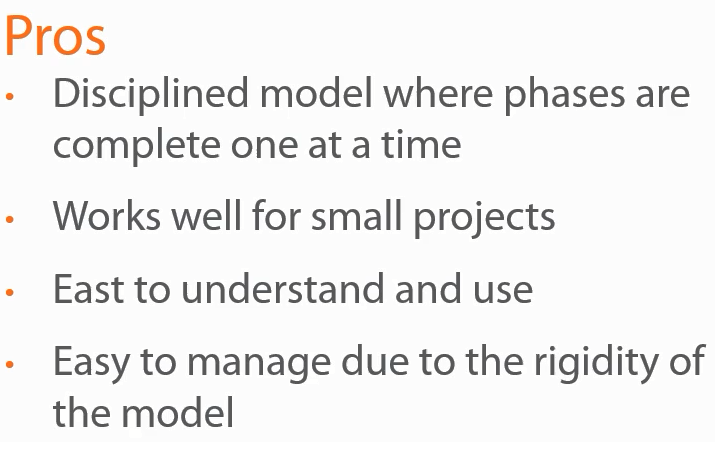


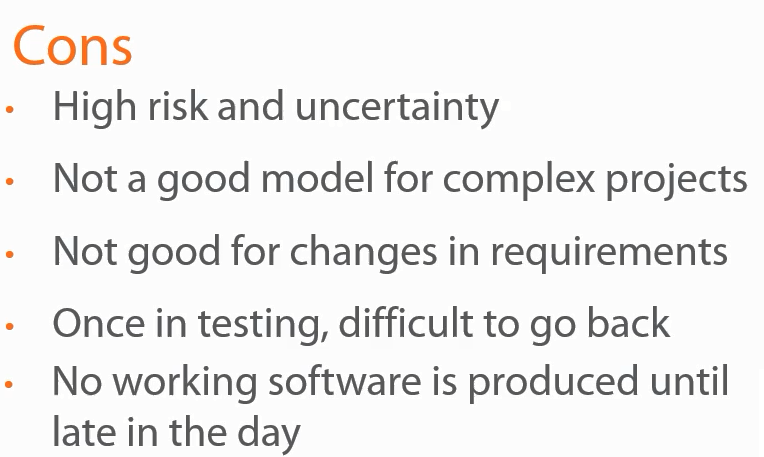
Suitable when:









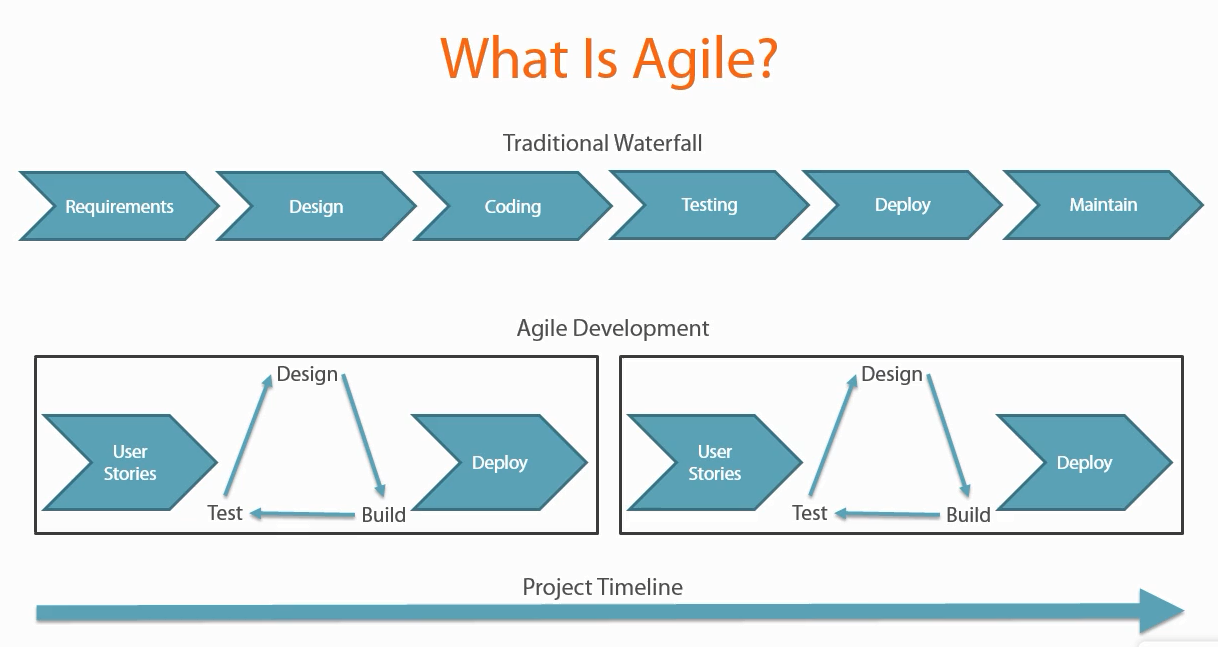


## Agile

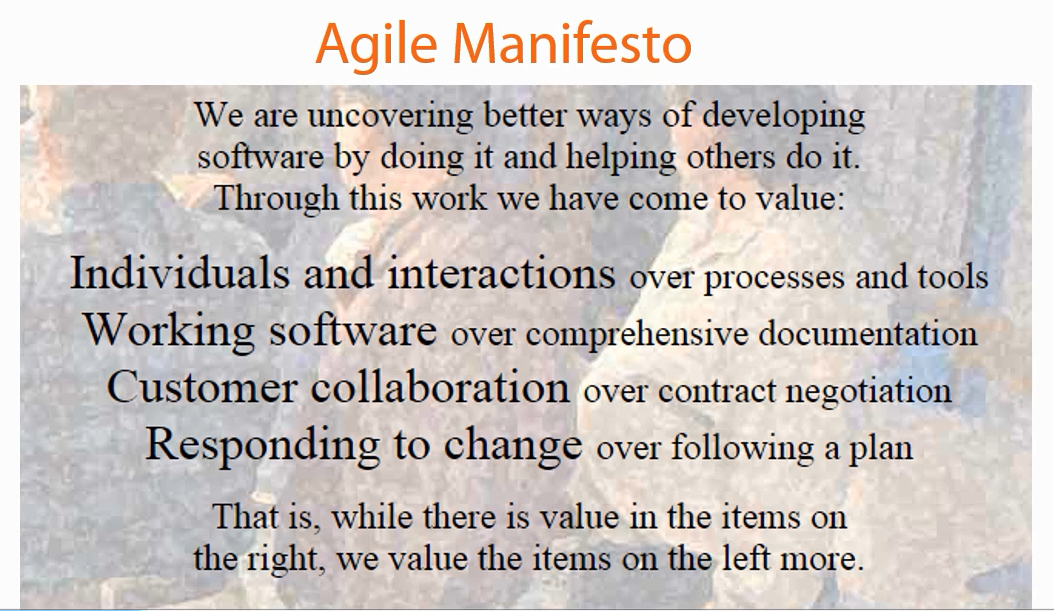
# Agile Software Development Methodologies

1. **Extreme Programming (XP)**
2. **Scrum**
3. Lean
4. Kanban
5. DSDM (Dynamic Software Development Method)
6. FDD (Feature Driven Development)
7. Crystal

## Agile Diagram



## Agile Manifesto



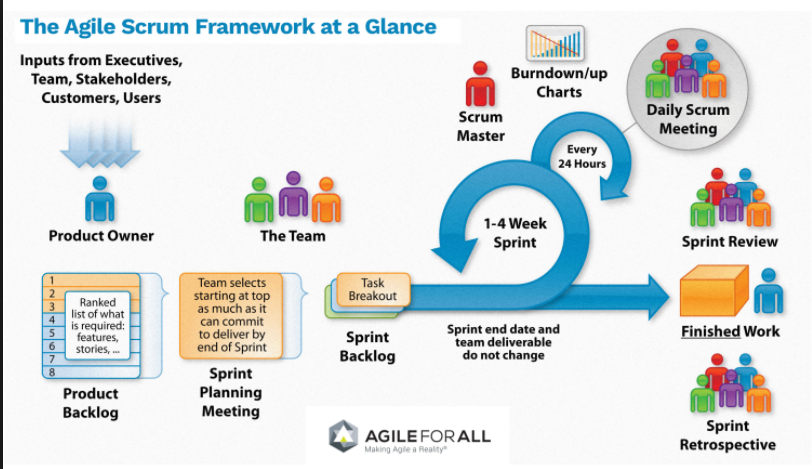
## Agilge Manifesto Principles



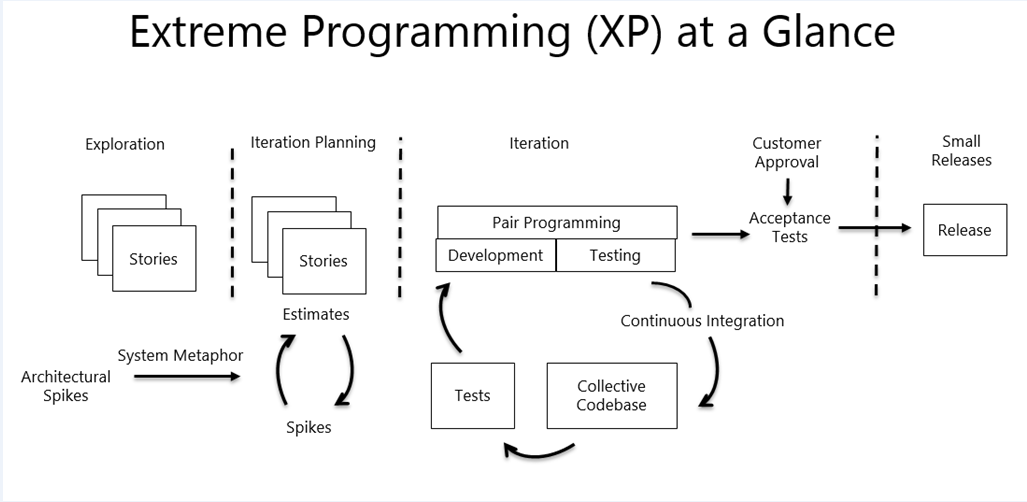
The twelve principles of agile development include:

1. **Customer satisfaction through early and continuous software delivery**– Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
2. **Accommodate changing requirements throughout the development process**– The ability to avoid delays when a requirement or feature request changes.
3. **Frequent delivery of working software**– Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
4. **Collaboration between the business stakeholders and developers throughout the project**– Better decisions are made when the business and technical team are aligned.
5. **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
6. **Enable face-to-face interactions**– Communication is more successful when development teams are co-located.
7. **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
8. **Agile processes to support a consistent development pace**–Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
9. **Attention to technical detail and design enhances agility**– The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
10. **Simplicity** – Develop just enough to get the job done for right now.
11. **Self-organizing teams encourage great architectures, requirements, and designs**– Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
12. **Regular reflections on how to become more effective**– Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

# Scrum



# Extreme Programming (XP)



* 12 Practices of Extreme Programming

1. Coding Standards
2. Collective Ownership
3. Continuous Integration
4. On-Site Customer
5. Pair Programming
6. Planning Game
7. Refactoring
8. Short Releases
9. Simple Design
10. Sustainable Pace (40 Hour Week)
11. System Metaphor
12. Test-Driven Development