Write short notes on following

Scrum

· Lean Development

· Extreme programming (XP)

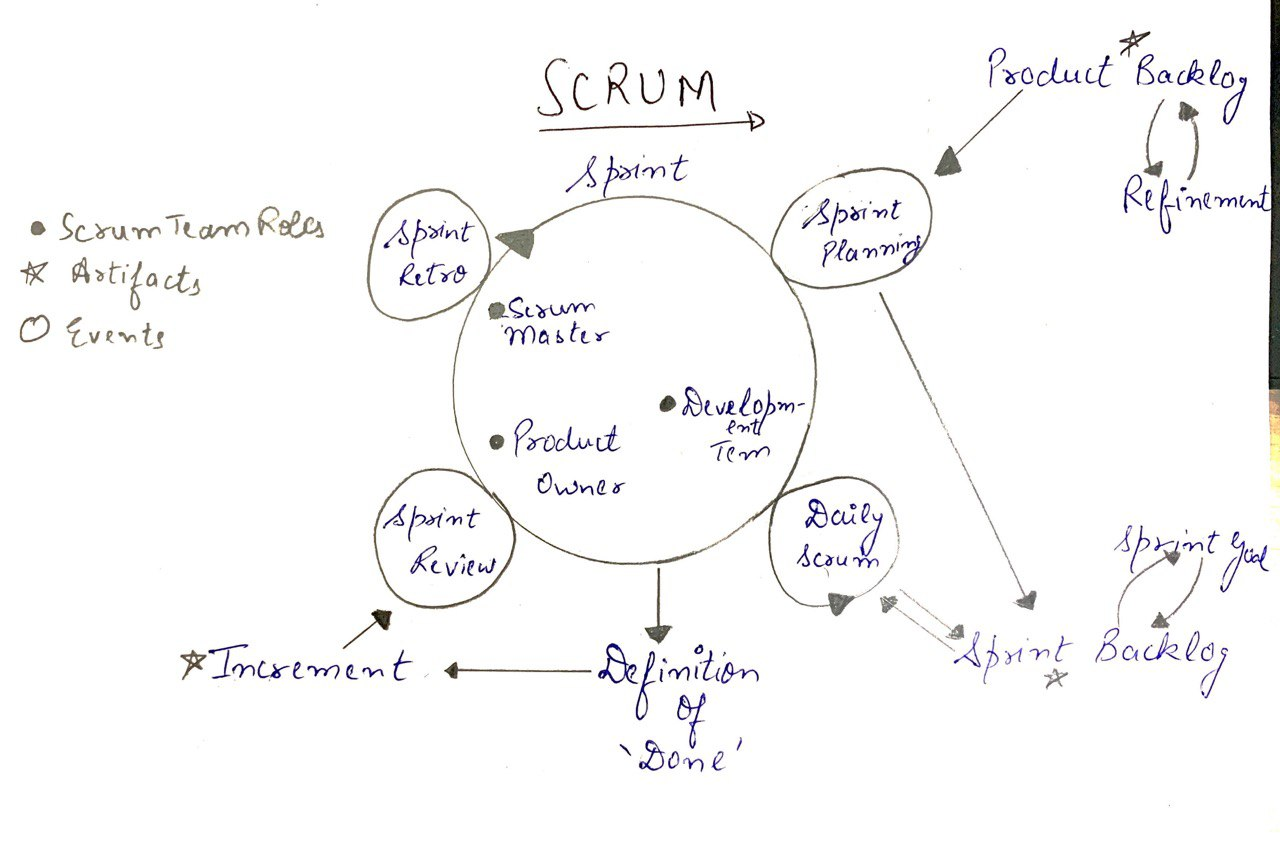
· Adaptive Software Development (ASD)

· Feature Driven Development

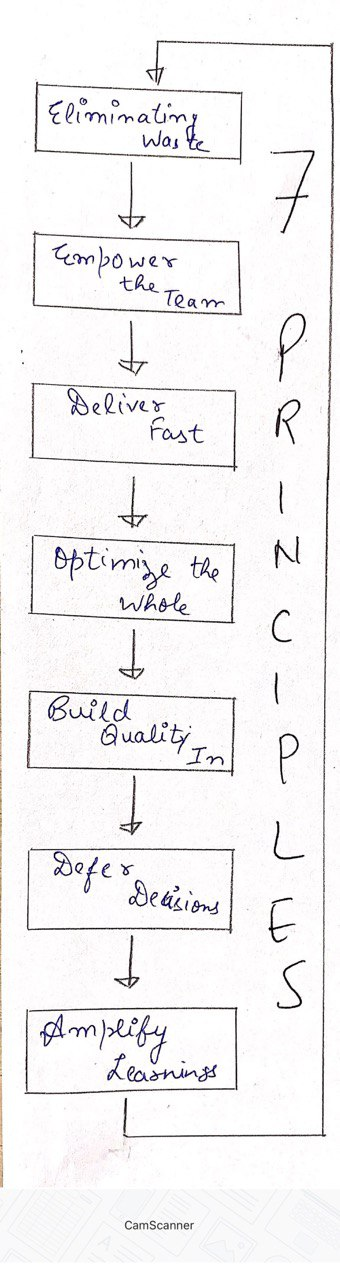
**Please scroll down for the answers.**

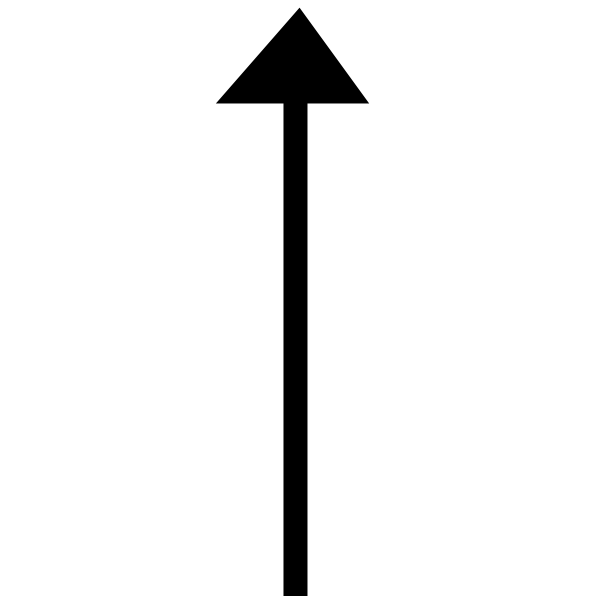
**Scrum**

* It is a framework to develop a complex product by keeping in view of majorly delivering and sustaining by emphasizing on teamwork, accountability & iterative progress.
* It helps to make the products in some chinks and in iterations, to follow according to the not stable market needs as well as the stringent goals of the organization.
* Scrum team consists of mainly 3 roles: a Product Owner, a Scrum Master and the Development team.
  + Product Owner :
    - Share the organizational view.
    - Handle the finance section of the business/product.
  + Development team :
    - Self Managed and mutual decision making.
    - Independent of other departments.
    - Usually small size teams.
  + Scrum Master :
    - Leader coaches the whole team.
    - Help in eliminating the obstacles.
    - Identifies ways to collaborate and keep security.
* Artifacts of a Scrum : Backlog of the product & sprint, Product increment and to consider ‘what is done?’.
* Activities in Scrum Basics : Product backlog refinement, sprint planning, daily scrum, sprint review and sprint retrospective.

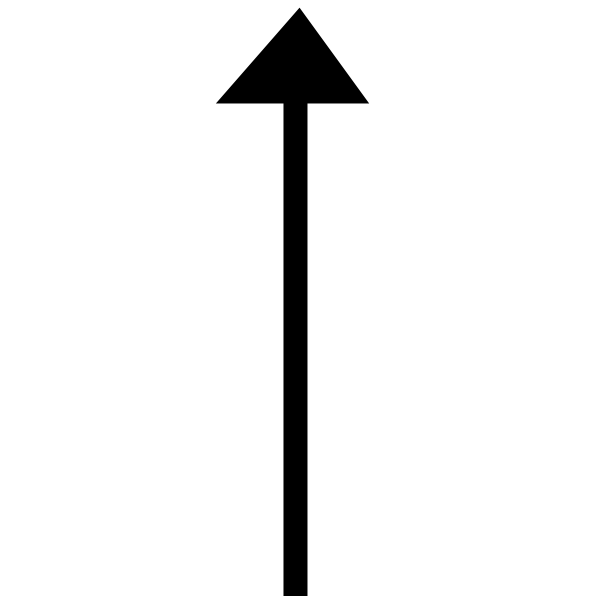


**Lean Development**

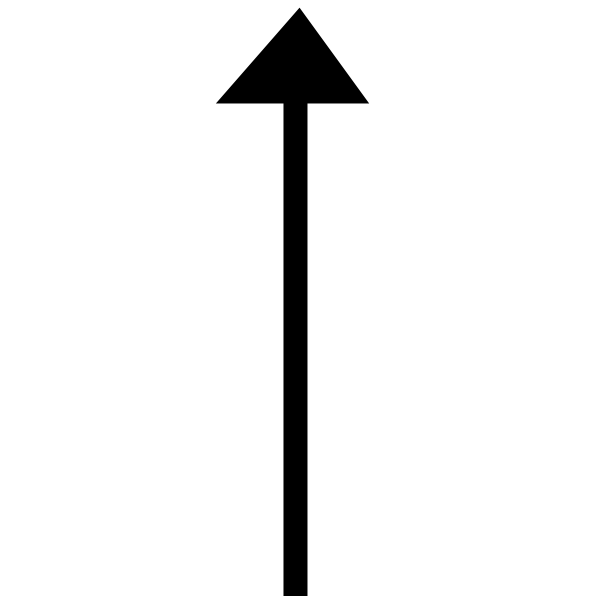
* A methodology to get better results by spending least time on development.
* Works on gathering important in the development model by doing multiple iterations.
* There are majorly 7 principles of Lean Development and it can be implemented in any idea and environment and can drastically improve the existing programming workflows.
* The 7 principles are :



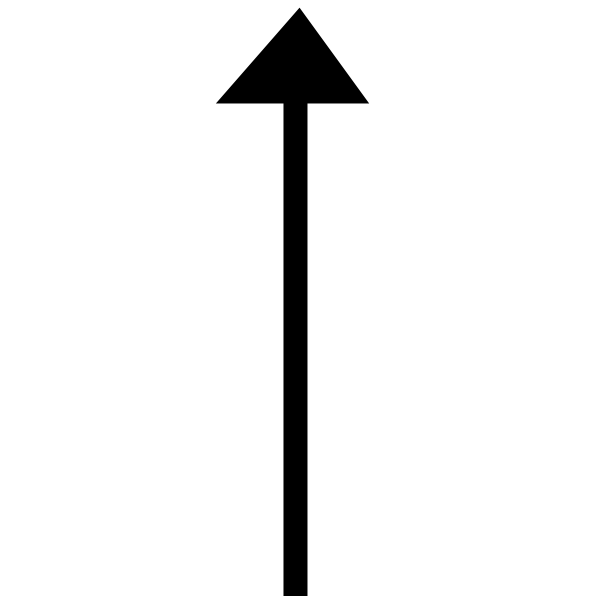
Removing what isn’t of value to the customer.



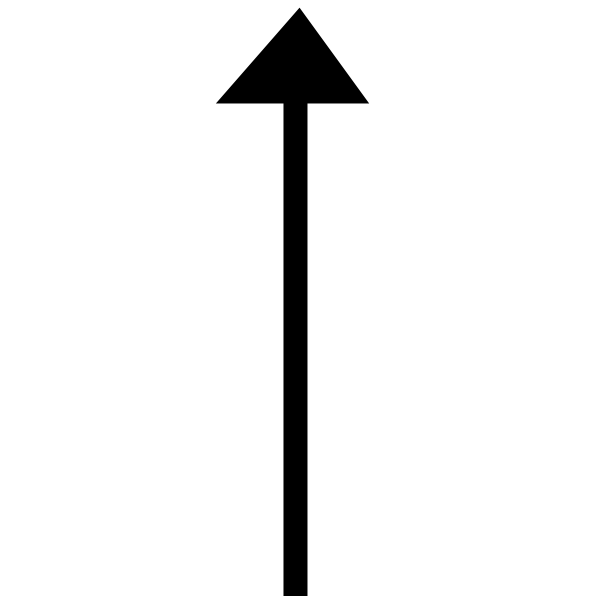
Allow the front-line workers (i.e. development) to make the major decisions.



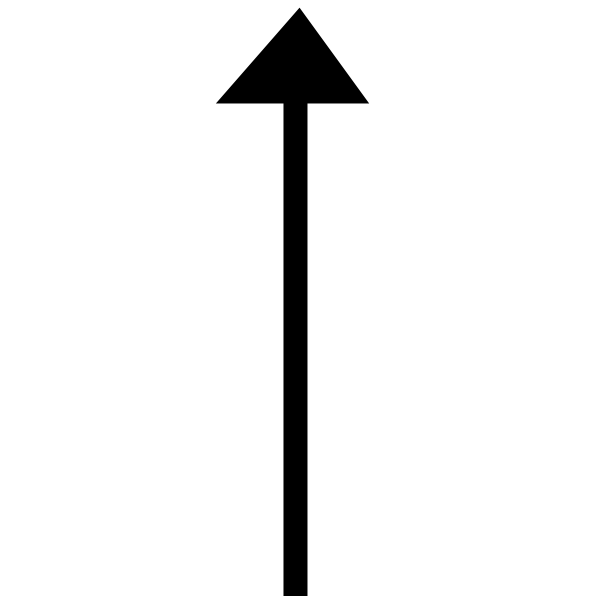
Quicker delivery of results with quicker feedback



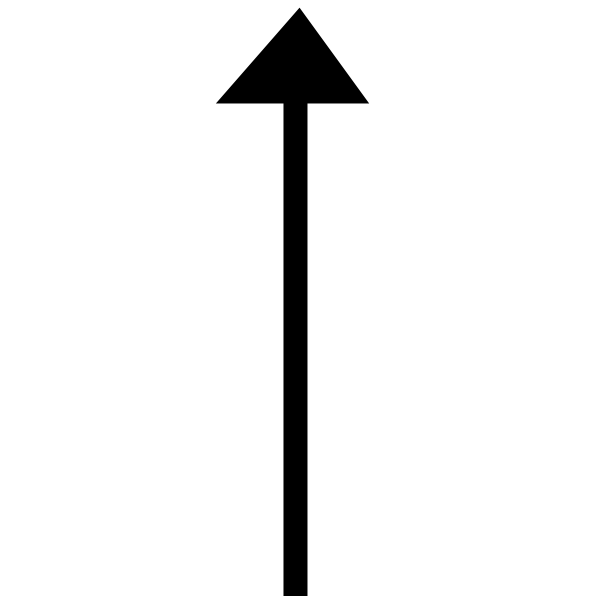
View the software as a whole and not as a sum of its parts (System Thinking)



Build for perceived (customer view) conceptual (system view) integrity, flexibility and efficiency.

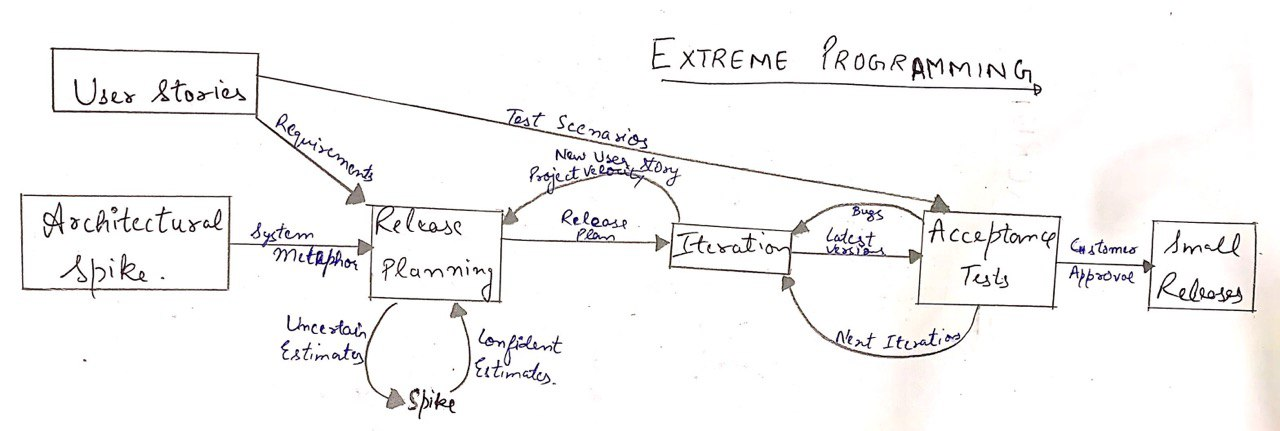


Delay and defer decisions until assumptions become fact.

Improve software development by improving the process through learning.

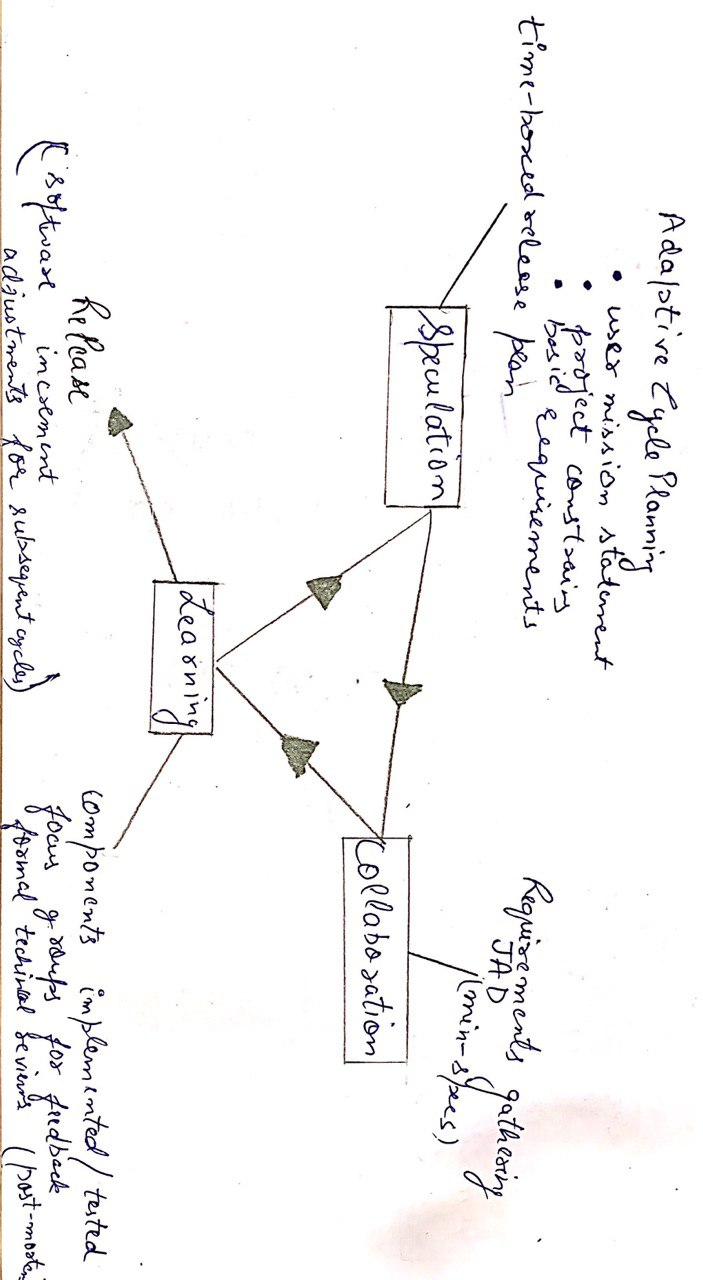
**Extreme Programming (XP)**

* An agile development methodology usually for small to medium sized teams developing software in the face of vague or rapidly changing requirements.
* It was created by Kent Beck in the mid 1990’s.
* In this basically, a set of 12 key practices are to be taken to their “extremes”.
* It basically works by uniting the whole team in the simple practices so that there would be enough feedback for the improvement in their specific domain with their unique situation.
* The four core values of XP are communication, simplicity, feedback and courage.
* It is called a light weight process because of the following reasons :
  + Since customers want up the front not the lot of information nailing down so XP works on plenty of feedback.
  + It keeps the customer involved by embracing change i.e. iterate often, design and redesign, code and test frequently.
  + In only a short period of time (say 2 weeks) by doing short iterations, the product is delivered to the customer for feedback.
  + One of the major features is the reduced cost since it eliminates the defects early.

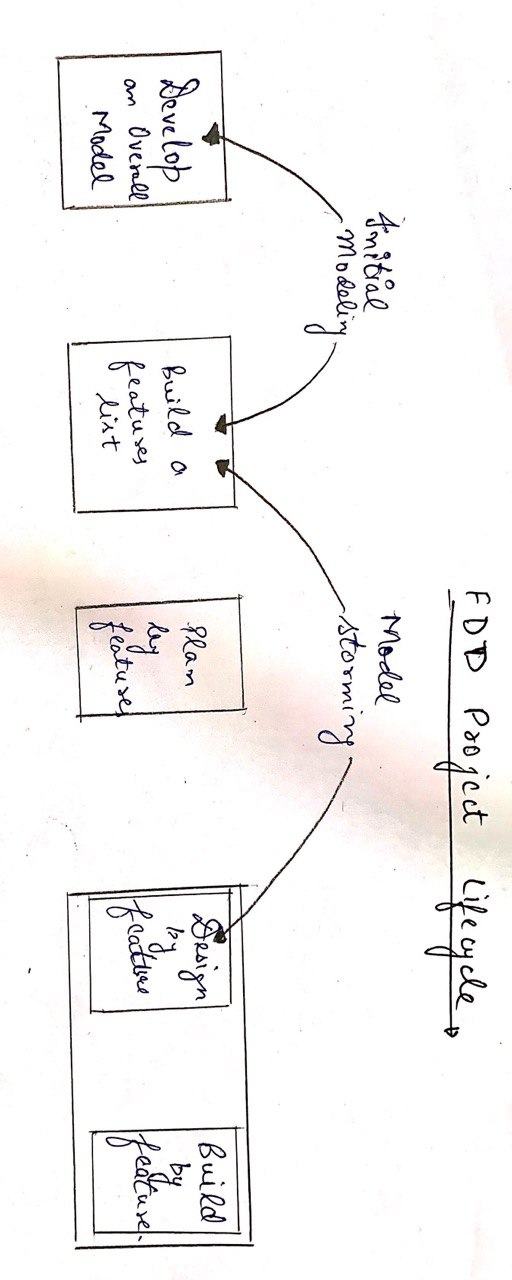


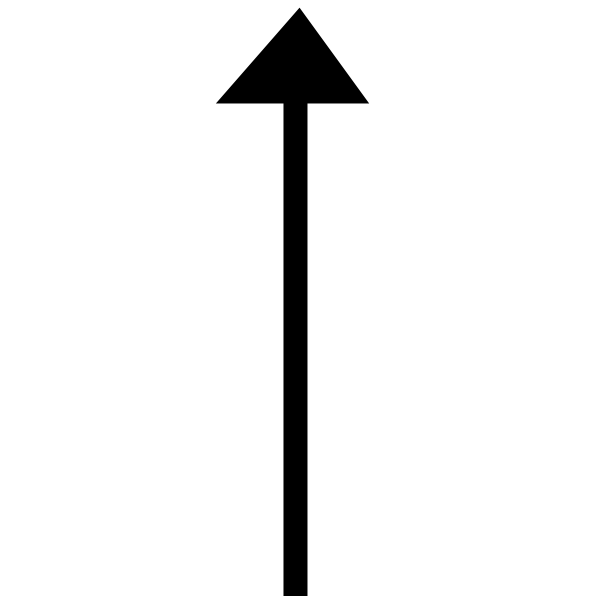
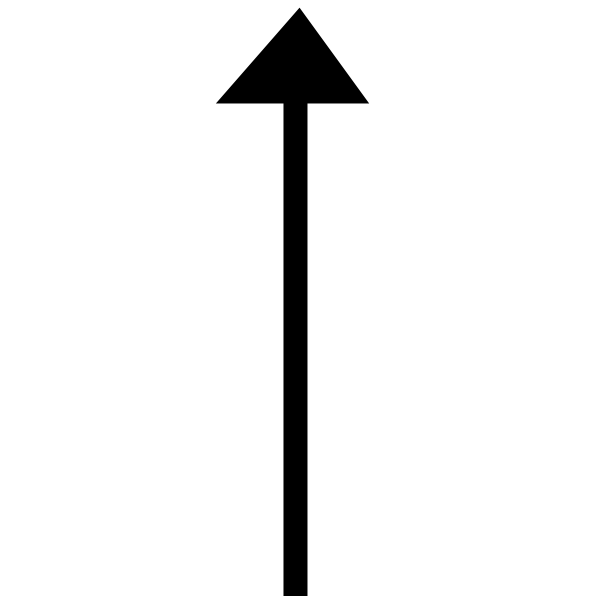
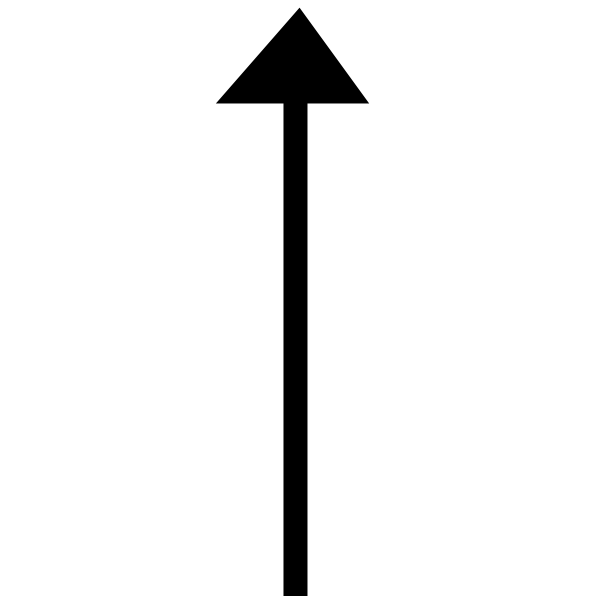
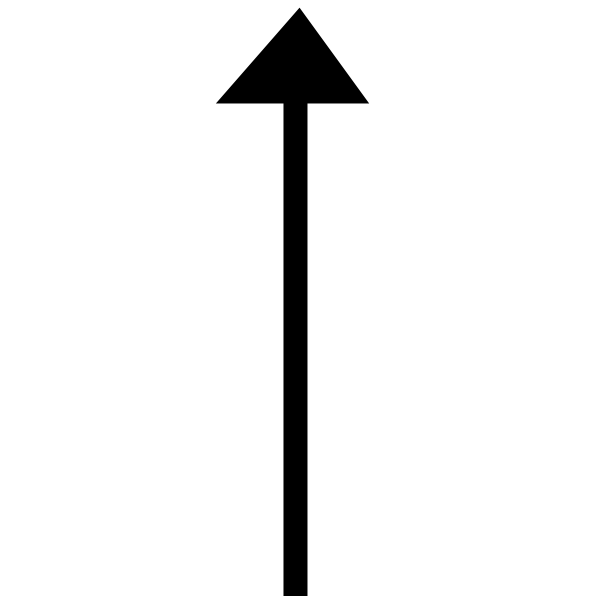
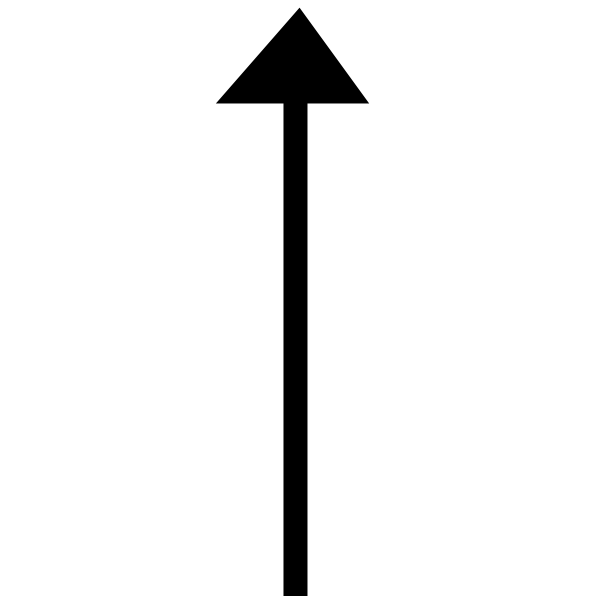
**Adaptive Software Development**

* It was originally proposed by Jim Highsmith and Sam Bayer.
* It is a collaborative approach to managing complex systems.
* It embodies the principles that continuous adaptation of the process to the work at hand is the normal state of affairs.
* It’s distinguishing and important features are :
  + Mission-driven planning.
  + Component based focus.
  + Uses “Time-Boxing”
  + Explicit consideration of risks.
* It is like the Evolutionary model because of its cyclical structure, with the phases names reflecting the unpredictability in the complex systems which are :
  + Spectuate
  + Collaborate
  + Learn

****

**Feature Driven Development**

* It blends a number of industry-recognized best practices into a cohesive whole, including domain object modeling, feature teams, and version control.
* The best practices that make up FDD are :
  + Domain Object Modeling
  + Developing by feature
  + Individual Class (Code) Ownership
  + Feature Teams
  + Inspections
  + Regular Builds
  + Configuration Management
  + Reporting/Visibility of Results
* In short it is a framework following agile methods, organizing software development around making progress on features.



(more shape than A list of features A development A design Completed

content) grouped into sets plan Class owners package client

and subject areas Feature set owners valued

function