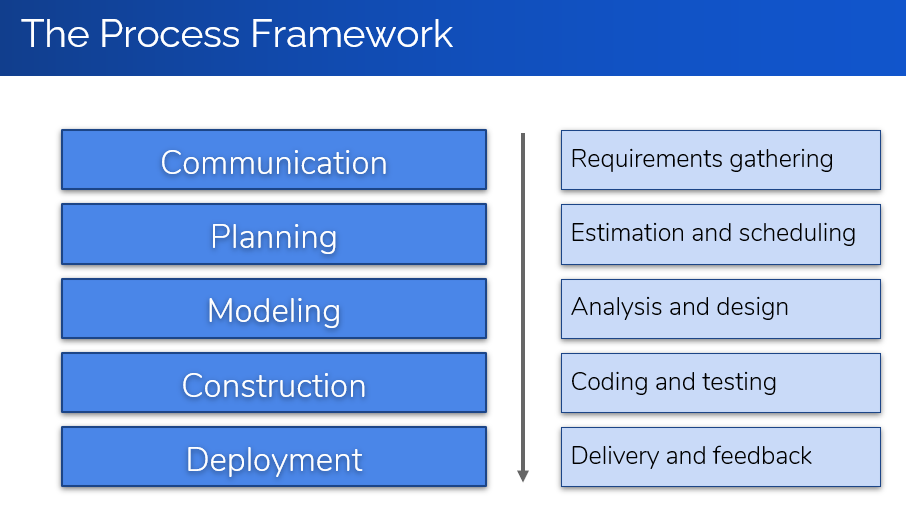
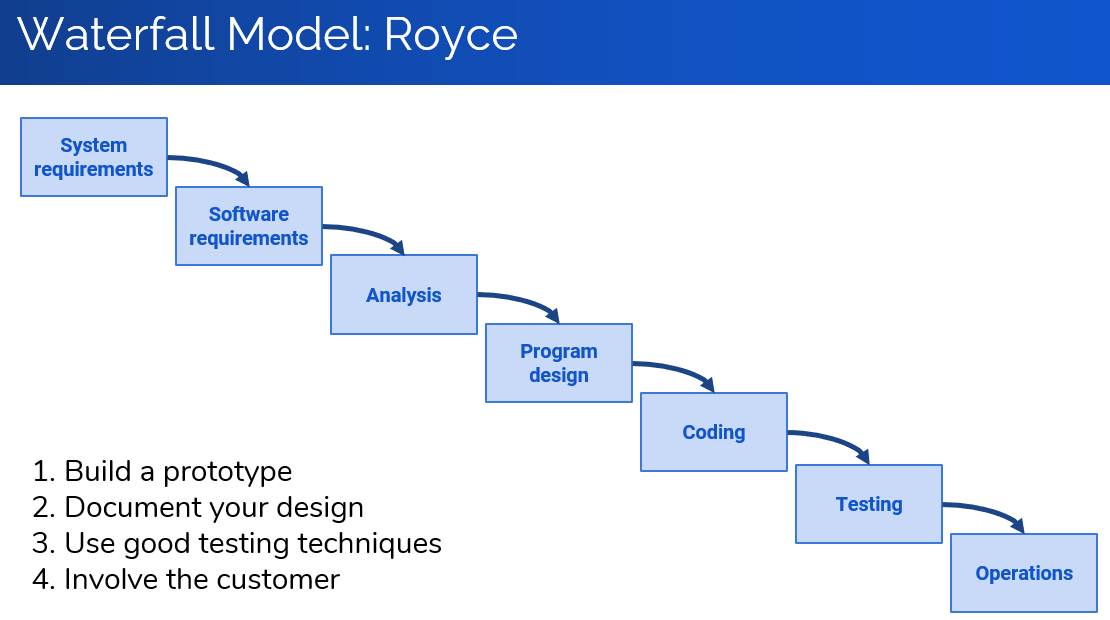
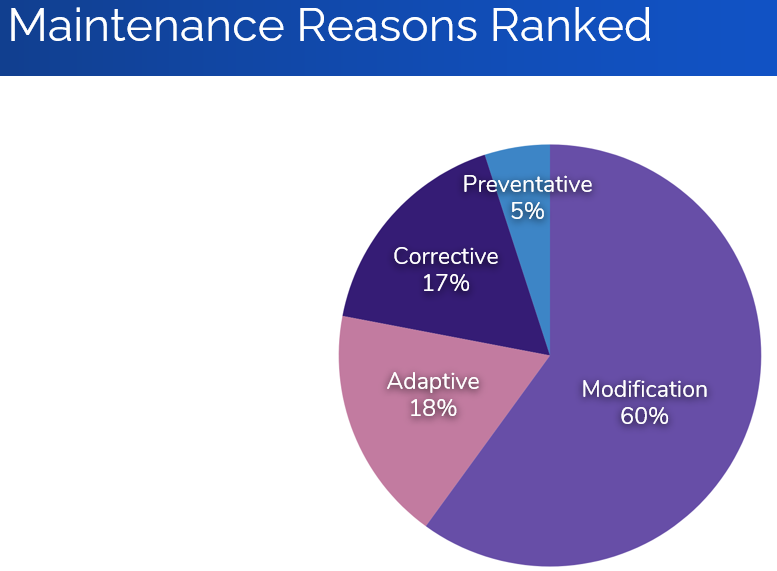
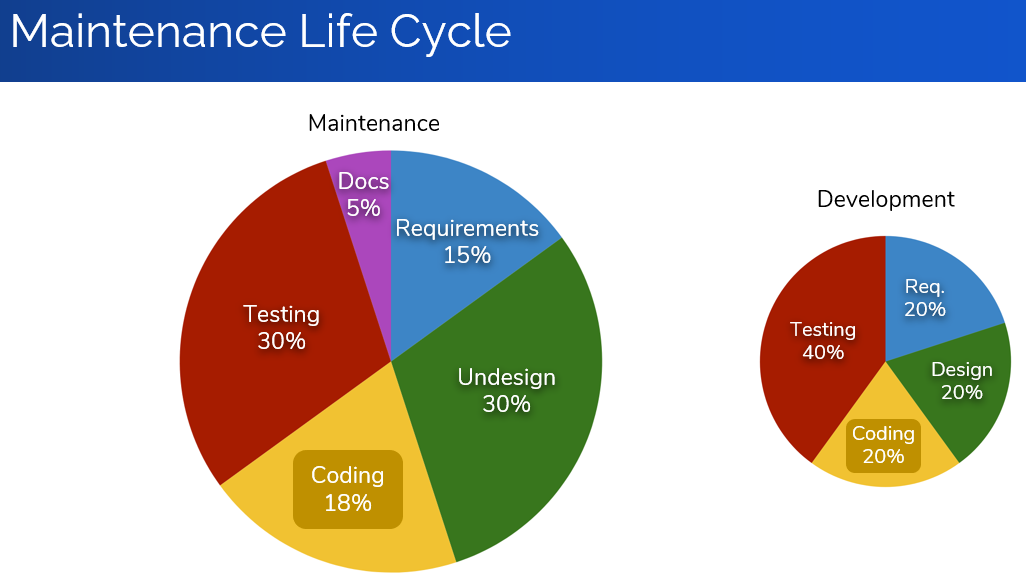
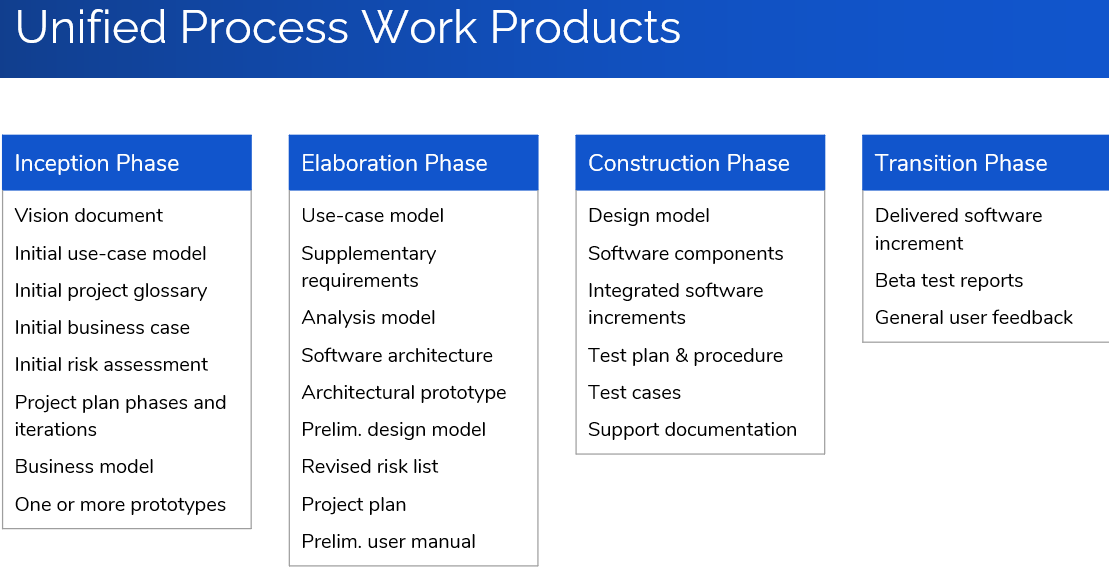
**Module 1**

* \*quality software\* is the foundation of software engineering.
* Software engineers want a disciplined approach to software development because they want to be able to build a quality product. But they also want an agile approach that allows them to respond effectively to change.
* 
* General Principles:
  + Provide value to users
  + Keep it simple
  + Maintain the vision
  + What you produce,  
    others consume
  + Be open to the future
  + Plan ahead for reuse
  + Think before you act
* The Essence of Practice:
  + Understand the problem  
    (communication and analysis)
  + Plan a solution  
    (modeling and design)
  + Carry out the plan  
    (code generation)
  + Examine result for accuracy  
    (testing and quality assurance)
* Layered Approach to Software Engineering:
  + Tools
  + Methods
  + Process
  + Quality Focus
* Questions Asked by Software Engineers:
  + Why does it take so long?
  + Why are costs so high?
  + Why can’t we find errors earlier?
  + Why is maintenance so hard?
  + Why is measuring progress so hard?
* Brooks Law: Adding more programmers to a late project will make it even later

**Module 2**

* Waterfall model disadvantages:
  + Real projects rarely follow a sequential workflow.
  + It’s hard of a customer to explicitly state all requirements at the beginning of the project.
  + The customer may need to wait a long time before they get a working program.
  + Major mistakes may not be detected until the working program is reviewed.
* 
* The most time-consuming phase of the software development life cycle is **Maintenance**
* Second time-consuming is **Testing**
* Reasons for Maintenance:
  + Adaptive maintenance
  + Corrective maintenance
  + **Modifications** – the highest ranking
  + Preventive maintenance
* 
* 60 percent of software’s dollar is spent on maintenance, and 60 percent of maintenance is enhancement.
* 
* Prototyping Potential Problems:
  + Customers see what appears to be working software.
  + Design compromises made for the prototype become part of the system.
* **Rapid prototype** (throwaway) – Simple model of system to visually show users what their requirements may look like
* **Evolutionary prototype** – Robust, structured prototype that can continually be refined and rebuilt
* Spiral Model Misconceptions:
  + The spiral is simply a sequence of waterfall increments.
  + All project activities follow a single spiral sequence.
  + Every activity in the diagram must be performed, and in the order shown.
* Unified Process Model Characteristics:
  + Use-case driven
  + Architecture centric
  + Iterative
  + Aligned with UML
* 
* Spiral model has an evolutionary flow

**Module 3**