Programmer documentation clearly and comprehensively communicates the business aims of the system, the system architecture, detailed design, and operation through delivery of a comprehensive, attractive, and well-presented suite of architecture and design documentation using well defined and appropriate language, and well-structured syntactically correct UML diagrams.

- Business aims (Words on how we meet these business aims, everything subheadings)
  - Consensus
  - Fault tolerance
  - Based on proven algorithm - Raft
  - Cross platform
  - Open source
  - Usability
  - Security
  - Troubleshooting
- System architecture
  - Those 6 architectural components
  - Extended Layers of Architectural Framework diagram
    - Talk about the elements, what is CAS?
  - Physical Architecture one: Nuget -> UAS/UAC/CAS -> Library
  - Code one (inside the UAS/UAC): Consensus -> Distributed log -> Networking
  - Class object diagram, how all the classes link together
- Detailed design / operations
  - Then a details class diagram and architectural diagram for each of the following systems
    - Common
      - PCQueues
    - Consensus
      - Usage of class
      - Joining cluster
      - The Raft Log
      - Wait loop system
      - Messages -> Handling, filtering, processing
      - Consensus itself -> Raft -> look at Raft
      - Threading
    - Networking
      - Usage of class
      - Threading
      - Message polymorphism
      - Node to IP conversion
      - States
      - Security
    - Logging

  - Usage of class
  - Named pipe logging system
  - Named pipe/event/file
  - Buffering
- Unit testing
  - How to
  - Inheritance
- Nuget
  - Compile a new library
  - Upload to Nuget website, hide the others

Programmer documentation supports rapid and accurate understanding of software in a way that would allow a programmer to become productive in maintaining the software immediately after reading the programmer documentation.

- Show that it would allow a programmer to maintain software immediately
  - Show they can solve various non-existent bugs or extend features?