System Design Checklist

|  |  |
| --- | --- |
| **Reusability** | |
| Q1. Do the requirements contain software reuse details? |  |
| Q2. What are the maximum memory requirements for the reused component, and can those requirements? |  |
| Q3. What are the requirements/design specifications the reused software should meet? |  |
| Q4. Do the requirements contain remote access to reusable legacy systems? |  |
| Q5. Do the requirements contain the list of reusable objects/components which exist already? |  |
| Q6. What kind of strategy or policy does the project adopt to reuse software? |  |
| Q7. What percentage of software can be reusable in the project? |  |
| Q8. What kind of software reuse methodology does the project adopt? |  |

|  |  |
| --- | --- |
| **Usability** | |
| Q1. Based on existing documentation/information, do you understand the system in the context of each of the views in the system engineering hierarchy? |  |
| Q2. Is system output and input adequately defined? |  |
| Q3. Have expert and novice modes of interaction been defined? |  |
| Q4. Have important interfaces to all system elements been described? |  |
| Q5. Is the behaviour of the software consistent with the information it must process and the functions it must perform? |  |
| Q6. Has the UI been designed effectively with the use cases in mind? |  |
| Q7. Have business requirements been met in the use cases? |  |
| Q8. Have all users been identified? |  |

|  |  |
| --- | --- |
| **Integrity** | |
| Q1. Have all data objects been described? |  |
| Q2. Have all attributes been identified? |  |
| Q3. Do major functions remain within scope and has each been adequately described? |  |
| Q4. Does the system have a consistently designed user interface? |  |
| Q5. Will end users be able to find the functions they are familiar with? |  |
| Q6. Will proper naming conventions being followed? |  |
| Q7. Do the developers’ goals match with the customers goals? |  |
| Q8. Can we prevent the system from corrupting? |  |