System Construction and Implementation

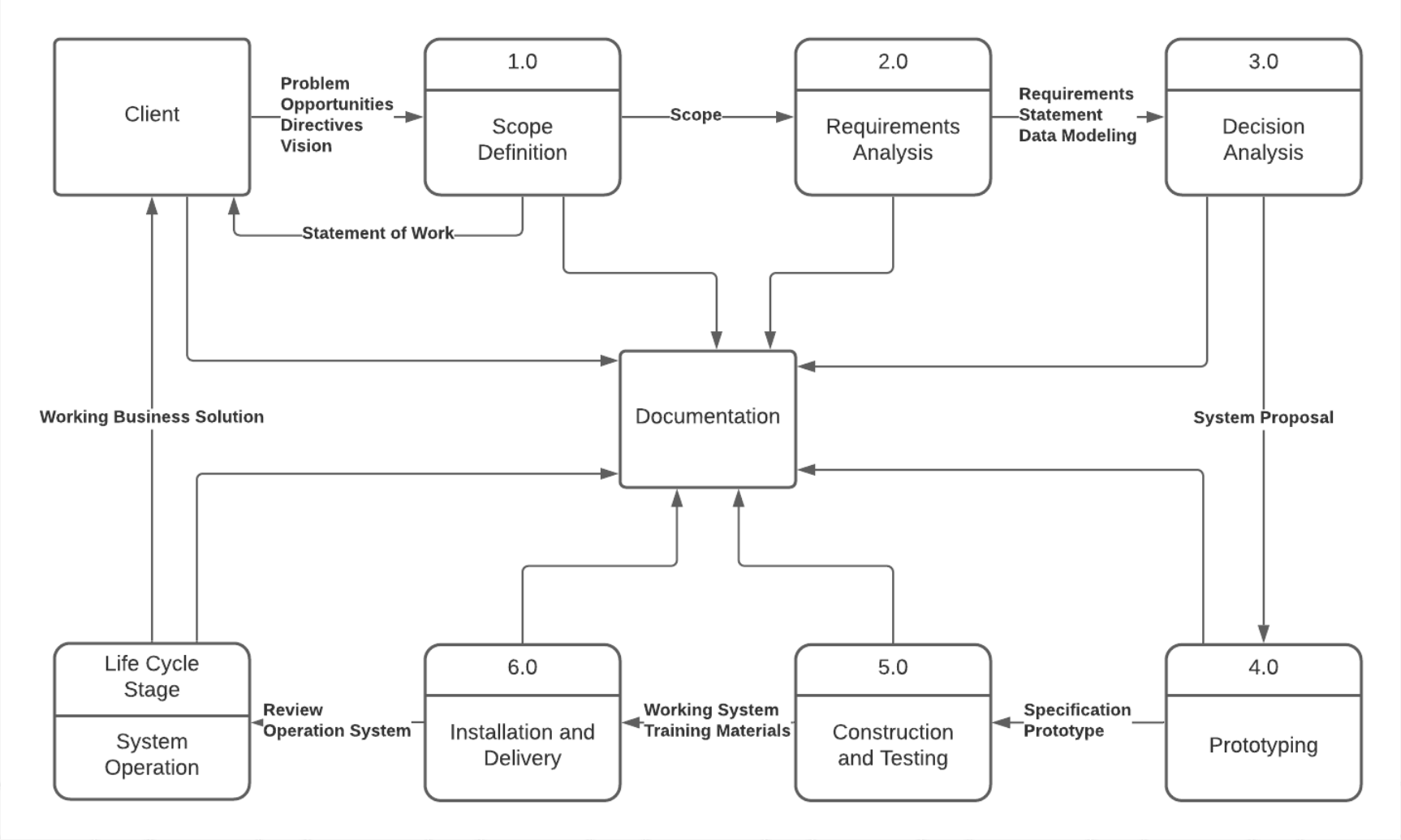
Objectives

- Understand the purpose of the construction and implementation phases of the systems life cycle
- Describe the systems construction and implementation phases in terms of your information building blocks
- Describe the systems construction implementation phases in terms of major tasks, roles, inputs and outputs
- Explain several application program and system tests
- Identify several system conversion strategies

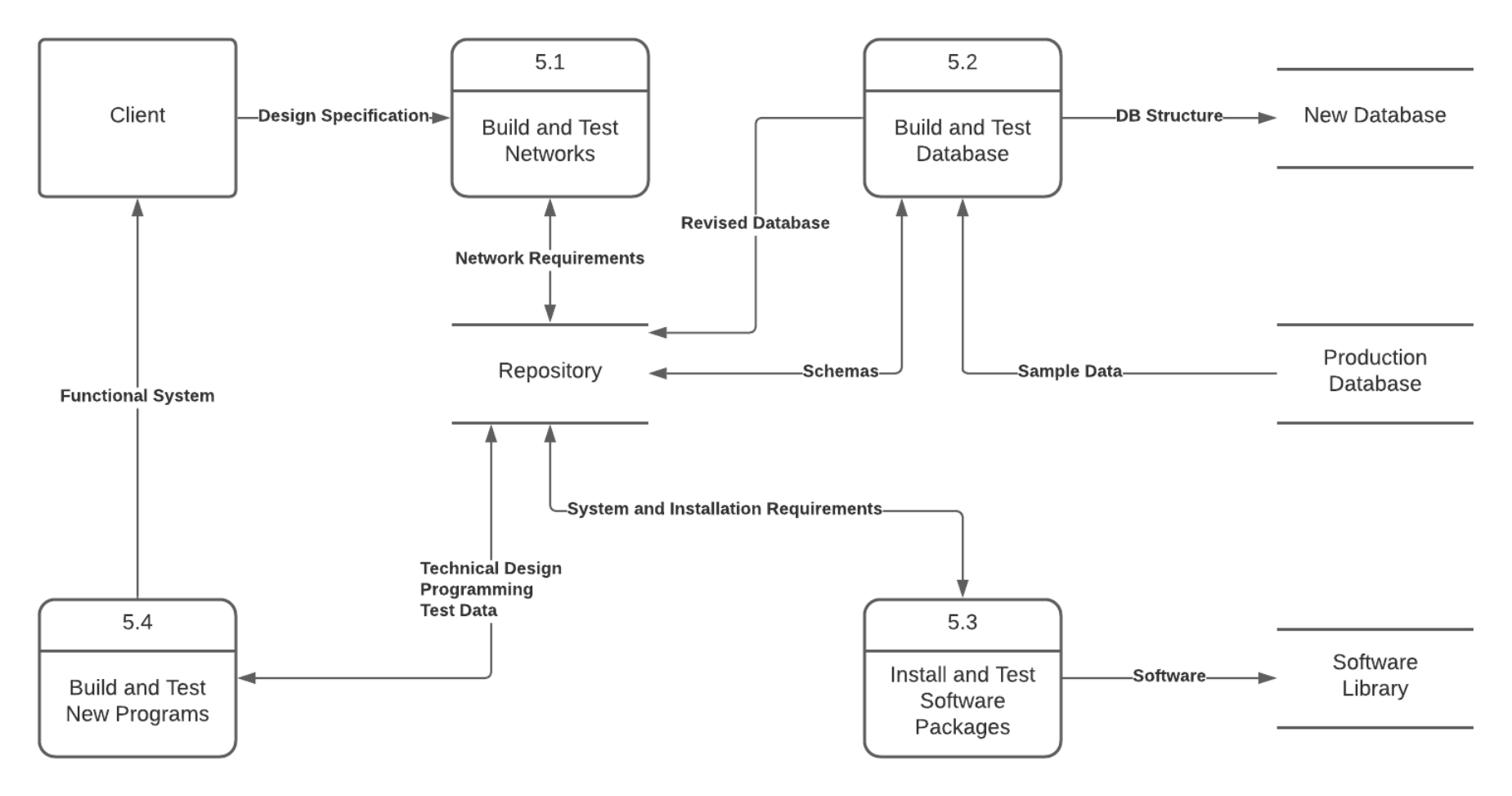
What is System Construction and Implementation?

- Systems construction – the development, installation, and testing of system components
  - A common but unfortunate synonym is systems development (more frequently used to describe the entire life cycle.)
- Systems implementation – the installation and delivery of the entire system into production
  - Day-to-day operation

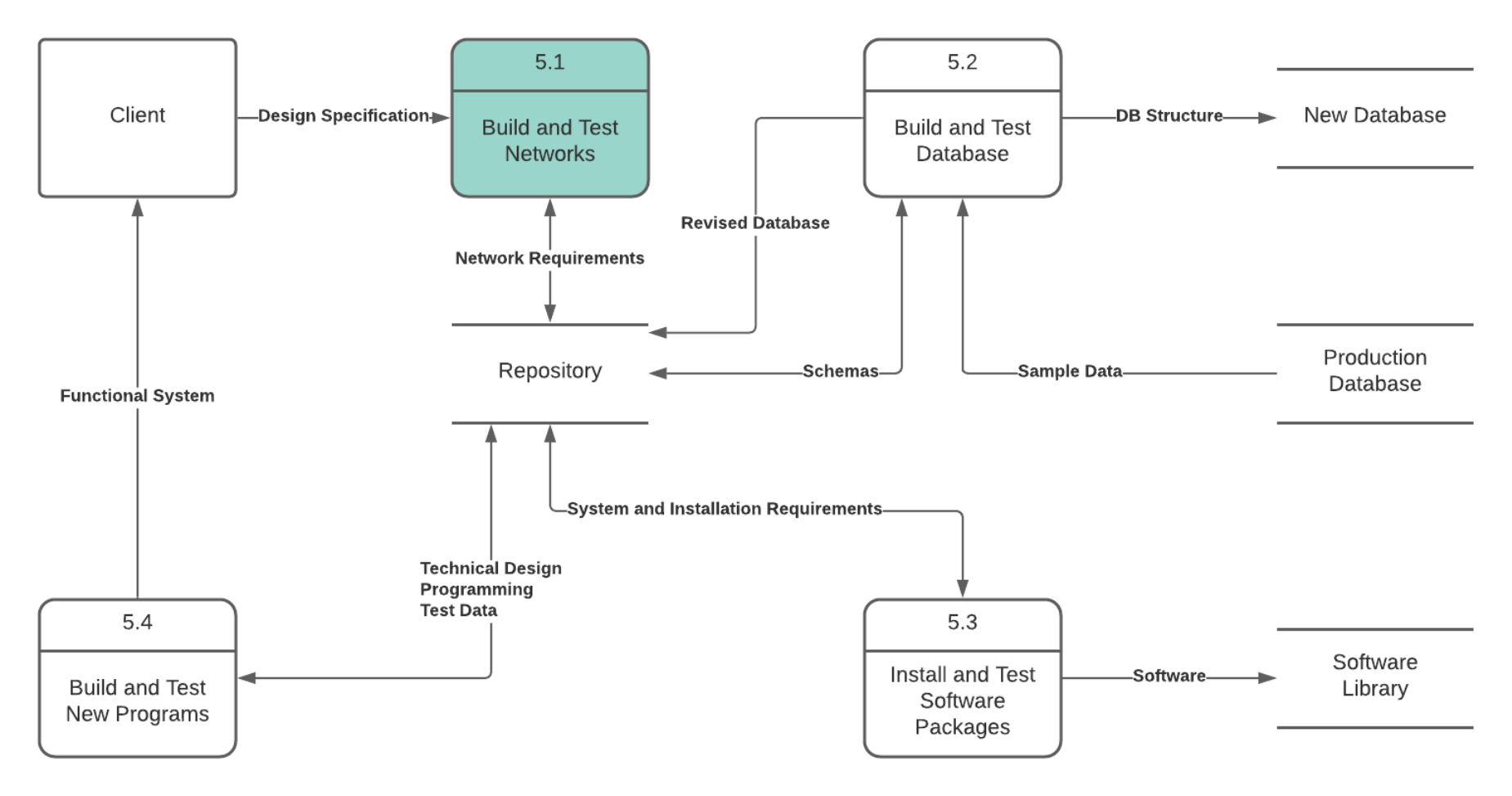
The Context of System Construction and Implementation



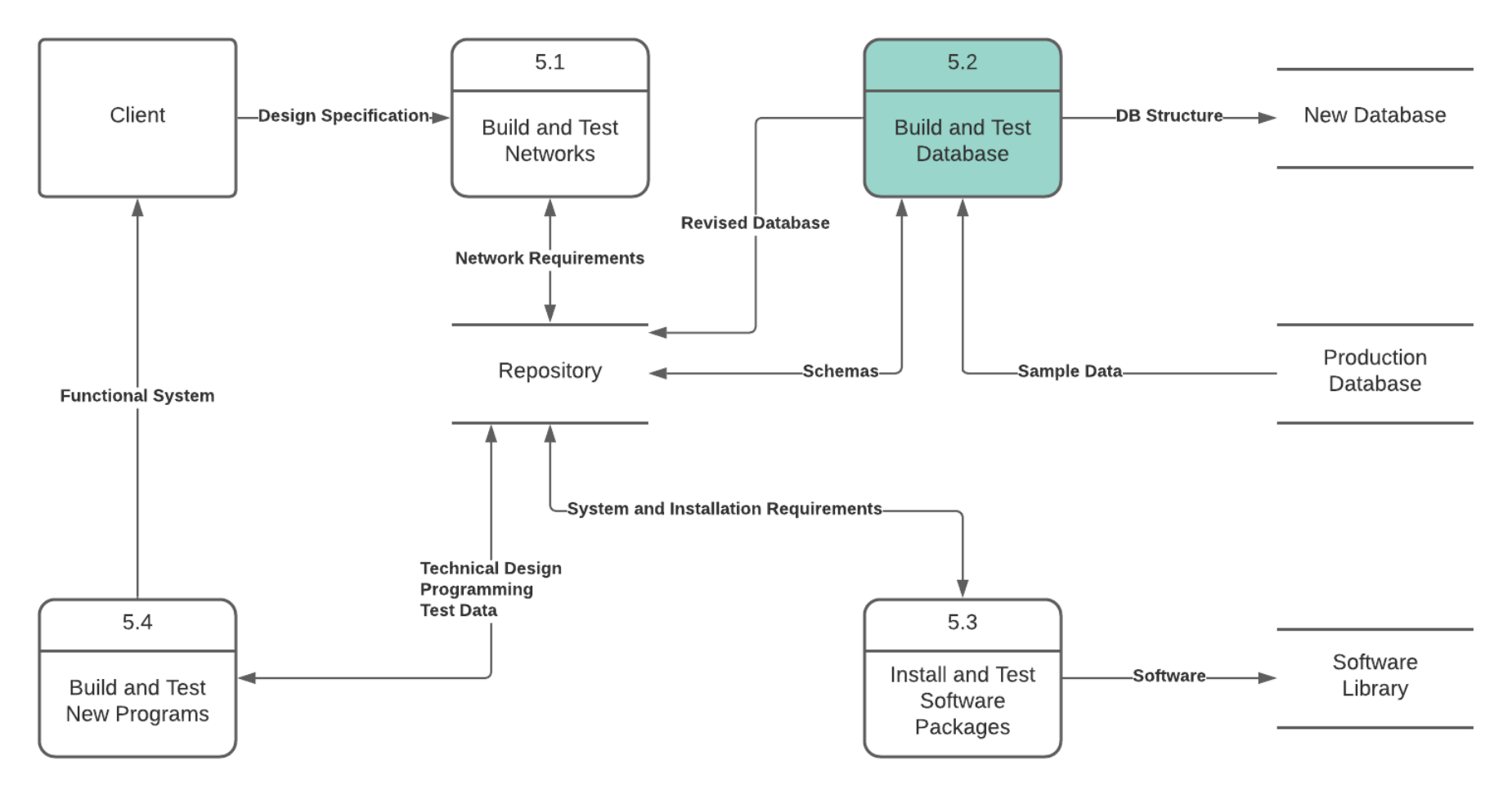
Task for Completing Construction Phase



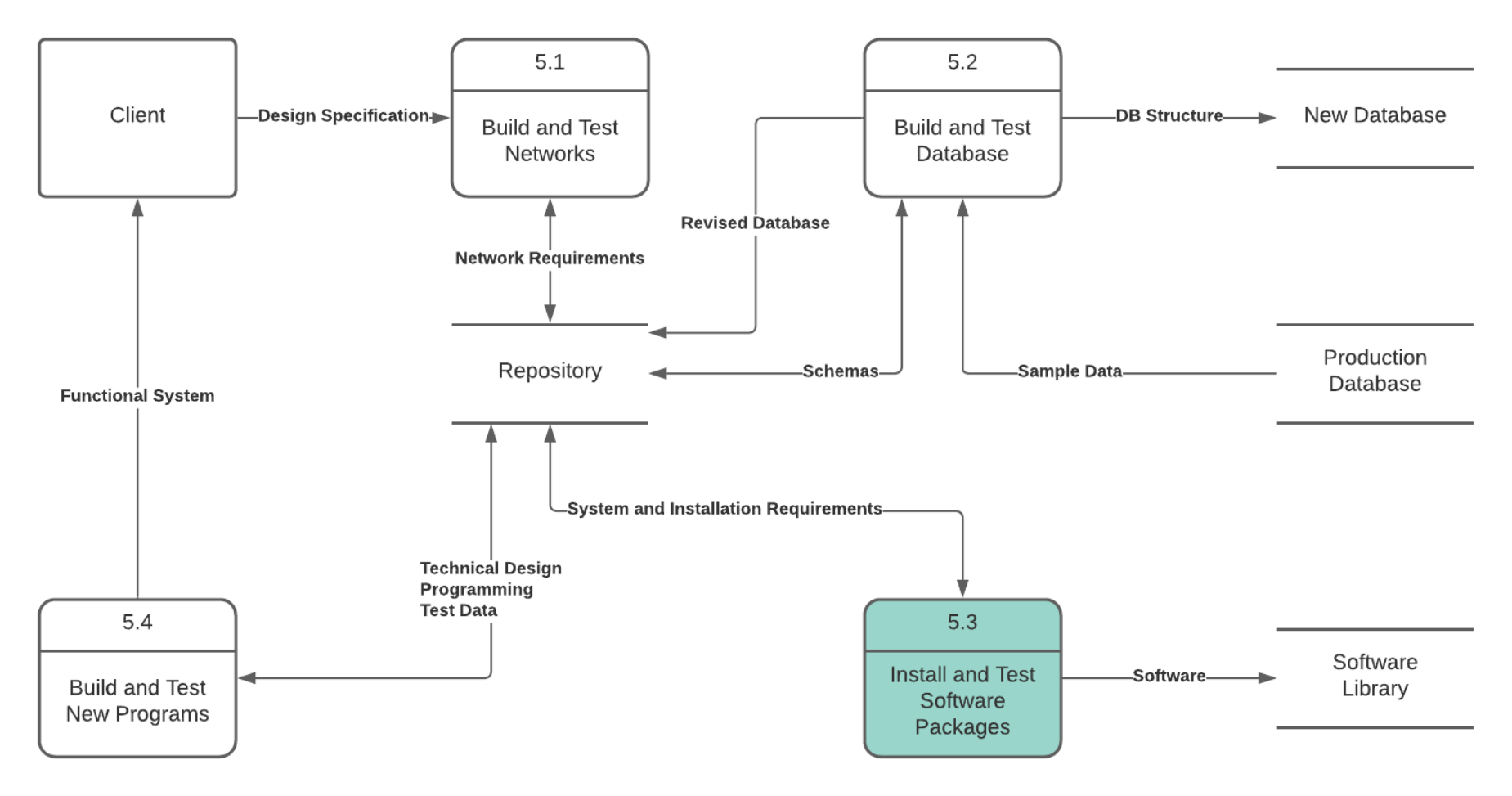
Task for Completing Construction Phase



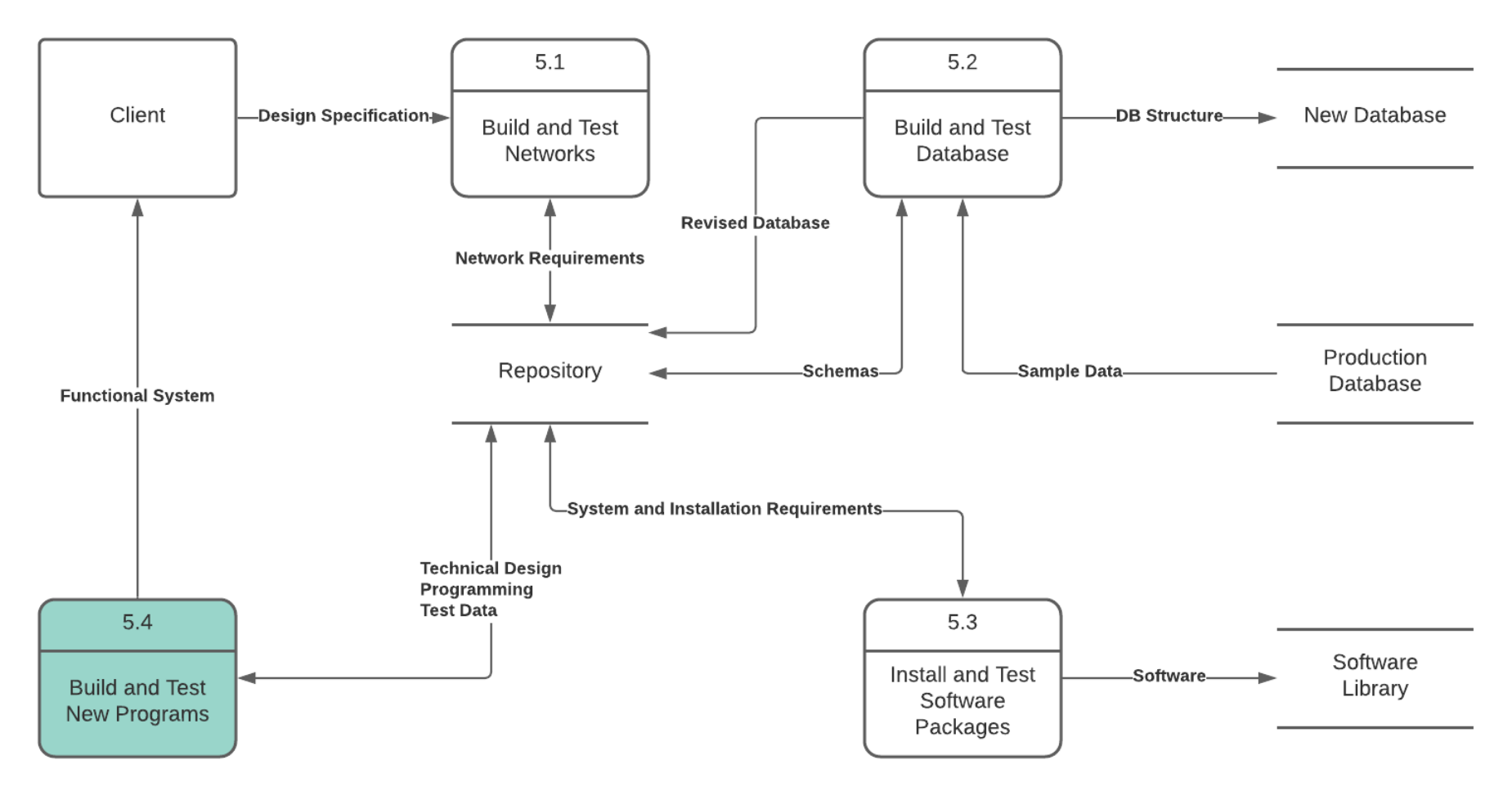
Task for Completing Construction Phase



Task for Completing Construction Phase



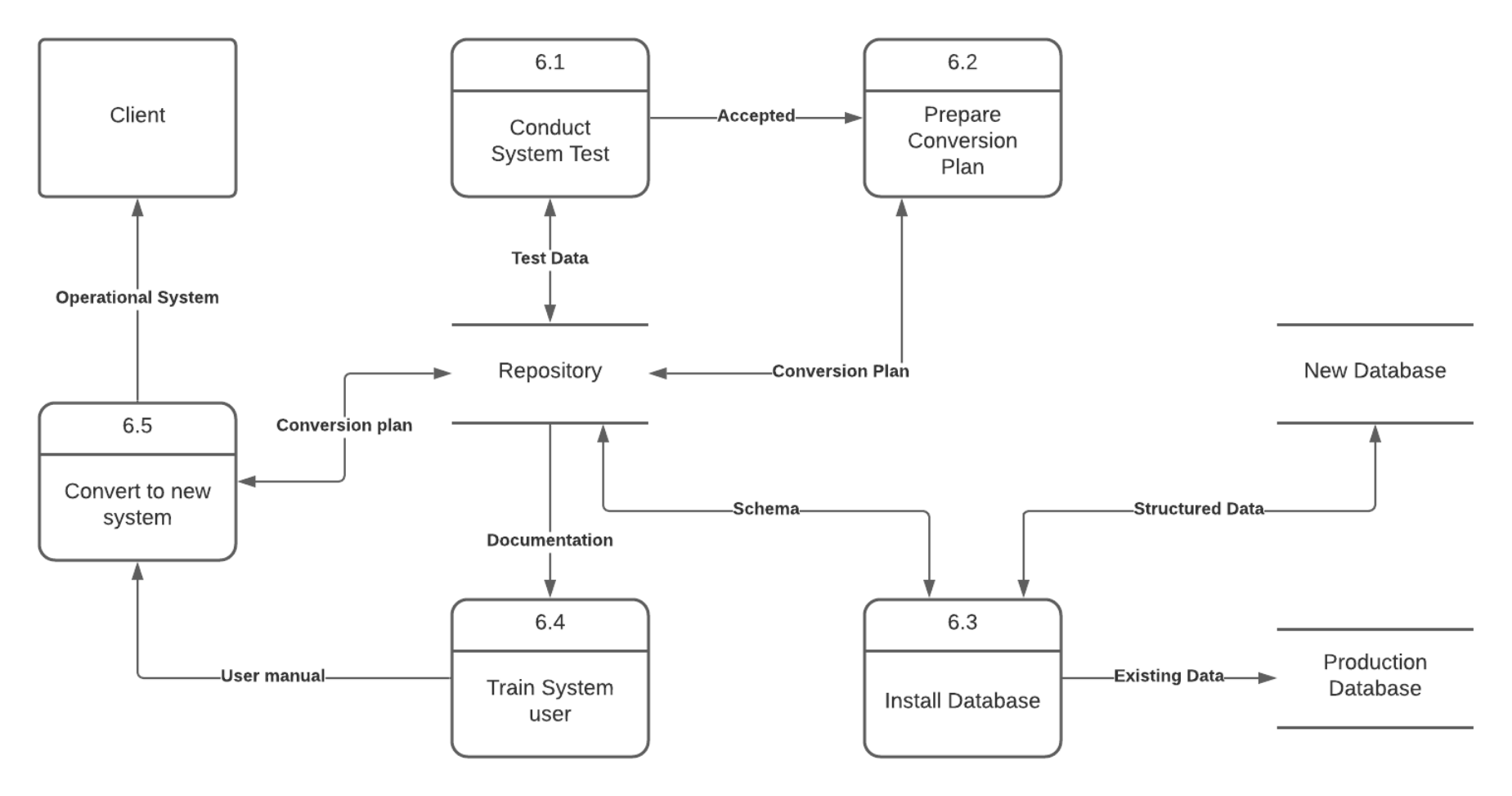
Task for Completing Construction Phase



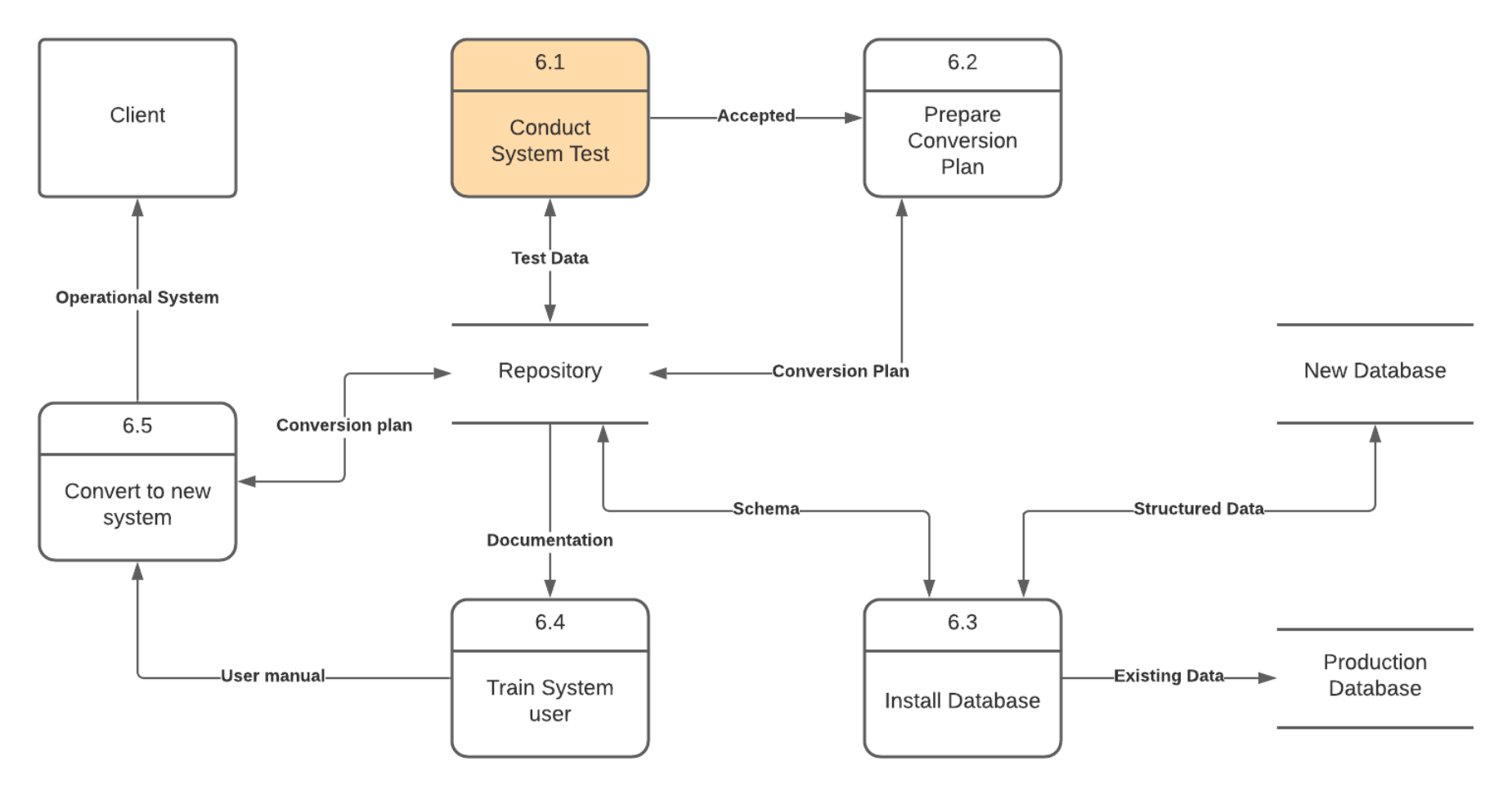
The Four Levels of Software Testing

- Unit Testing – test performed on a function, individual program or even a procedure.
- Integration Testing – test performed on a combined units within a program and test them as a group.
- System Testing – test performed on a system level on a pseudo production environment.
- Acceptance Testing – test performed on a user/consumer level to check if the system is ready for release.

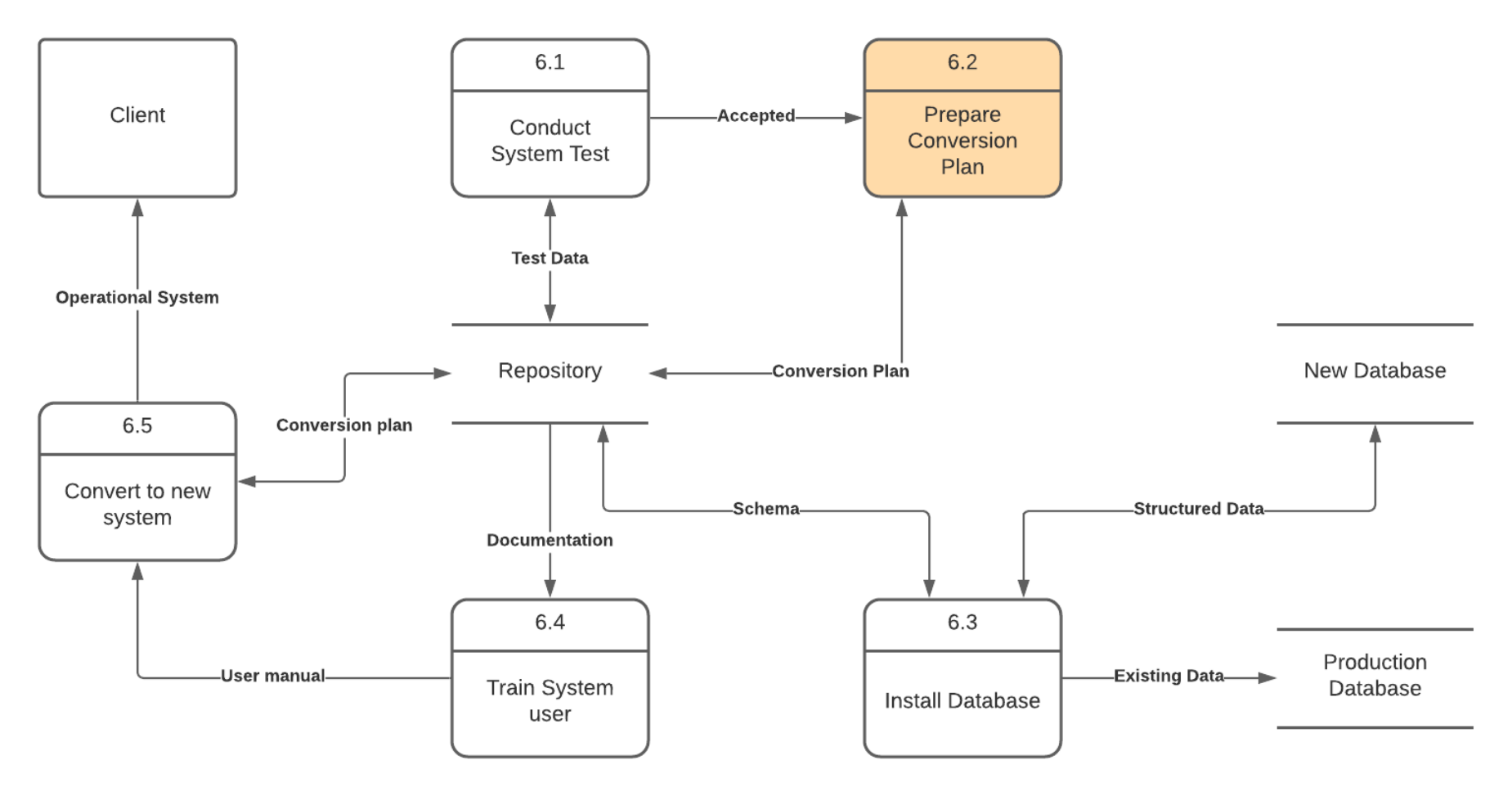
Task for Completing Implementation Phase



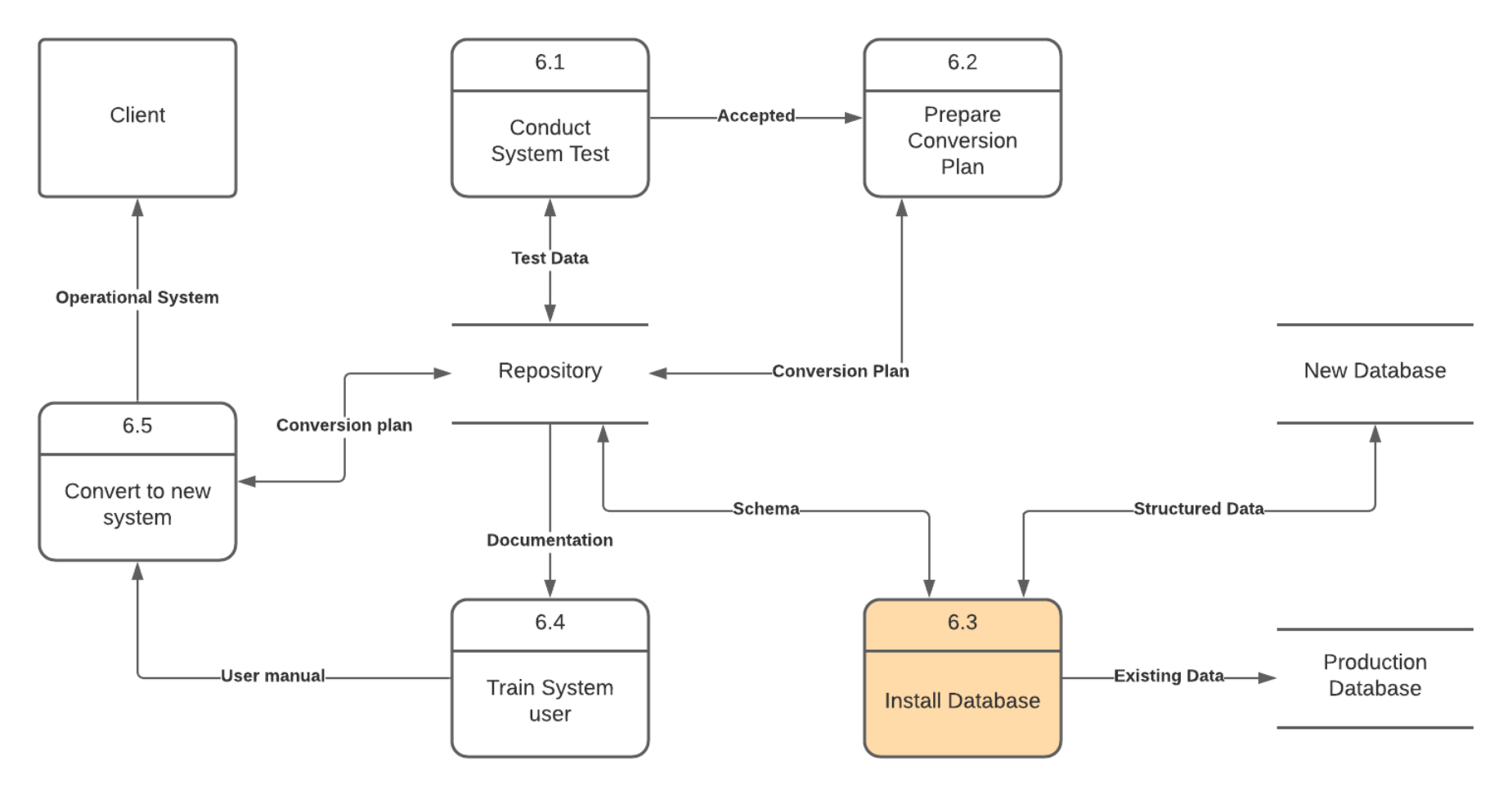
Task for Completing Implementation Phase



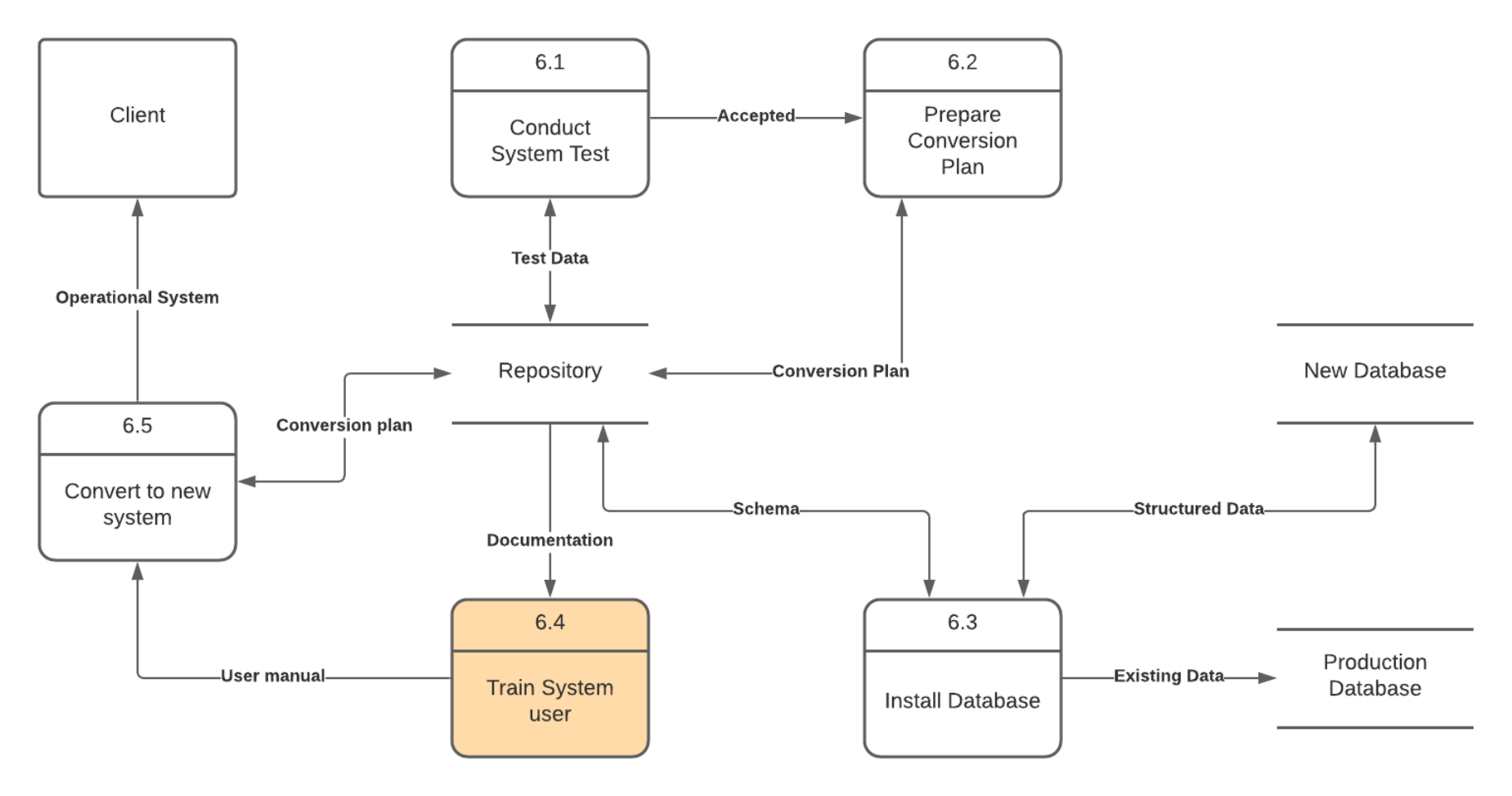
Task for Completing Implementation Phase



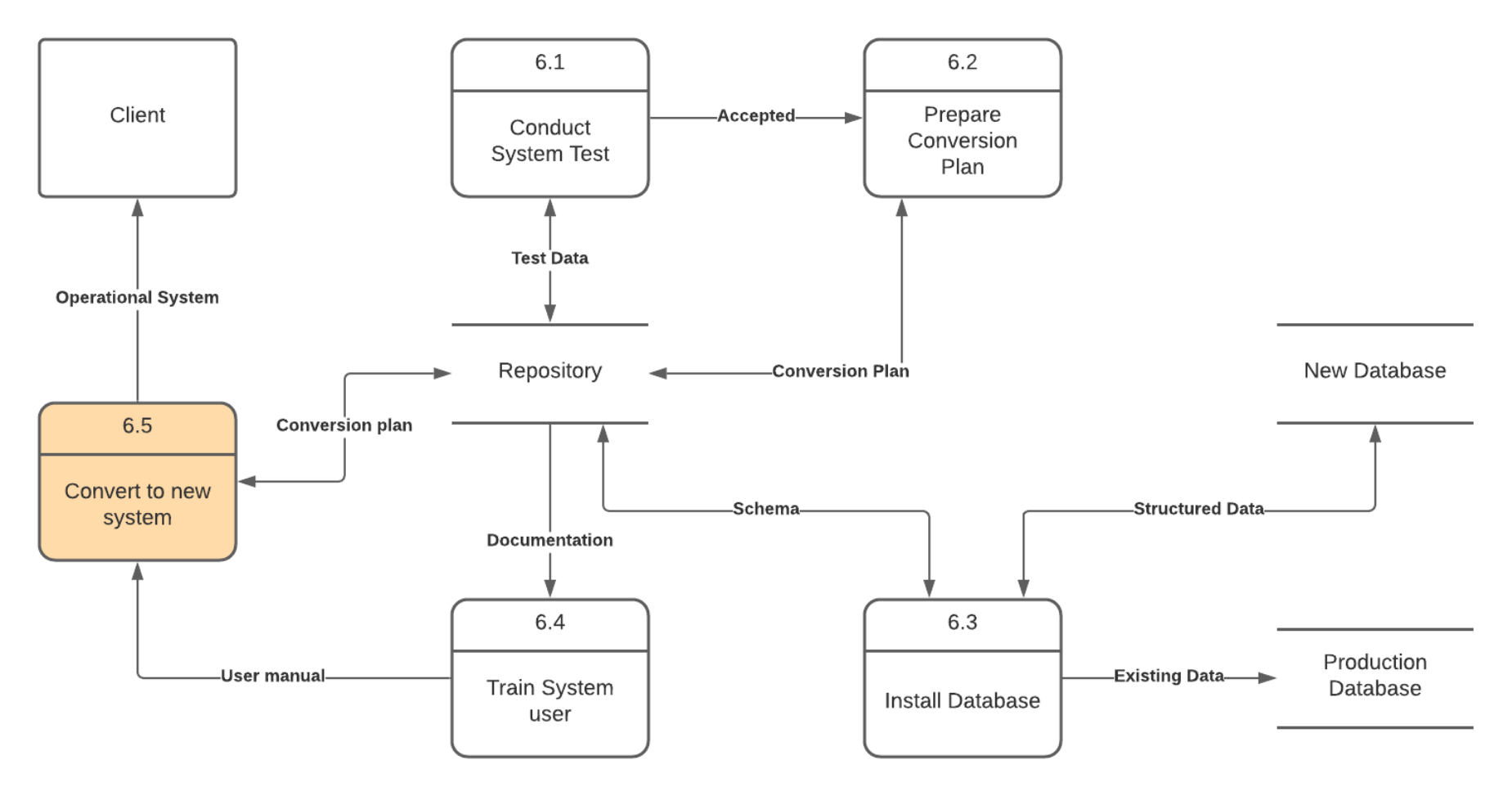
Task for Completing Implementation Phase



Task for Completing Implementation Phase



Task for Completing Implementation Phase



Different Levels of Software Reuse

- Abstraction Level - reuse knowledge of successful abstractions in the design of your software
- Object Level - reuse objects from a library rather than writing the code yourself
- Component Level - where you build your user interface using a framework
- System Level - reuse entire application systems

Host-target Development

- An integrated compiler and syntax-directed editing system that allows you to create, edit, and compile code.
- A language debugging system.
- Graphical editing tools, such as tools to edit UML models.
- Testing tools, such as JUnit that can automatically run a set of tests on a new version of a program.
- Project support tools that help you organize the code for different development projects.

Open Source Development (1)

- Open source development is an approach to software development in which the source code of a software system is published and volunteers are invited to participate in the development process.
- The best-known open source product is, of course, the Linux operating system which is widely used as a server system and, increasingly, as a desktop environment.

Open Source Development (2)

- Two open source issues:
  - Should the product that is being developed make use of open source components?
  - Should an open source approach be used for the software's development?

Open Source Licensing (1)

- The GNU General Public License (GPL). This is a so-called ‘reciprocal’ license that, simplistically, means that if you use open source software that is licensed under the GPL license, then you must make that software open source.
- The GNU Lesser General Public License (LGPL). This is a variant of the GPL license where you can write components that link to open source code without having to publish the source of these components. However, if you change the licensed component, then you must publish this as open source.

Open Source Licensing (2)

- The Berkley Standard Distribution (BSD) License. This is a non-reciprocal license, which means you are not obliged to republish any changes or modifications made to open source code. You can include the code in proprietary systems that are sold. If you use open source components, you must acknowledge the original creator of the code.

Assignment and Quiz for Next Meeting

- Assignment
  - Create a conversion plan to migrate old system to new system
- Short Quiz
  - Topic: Systems Construction and Implementation