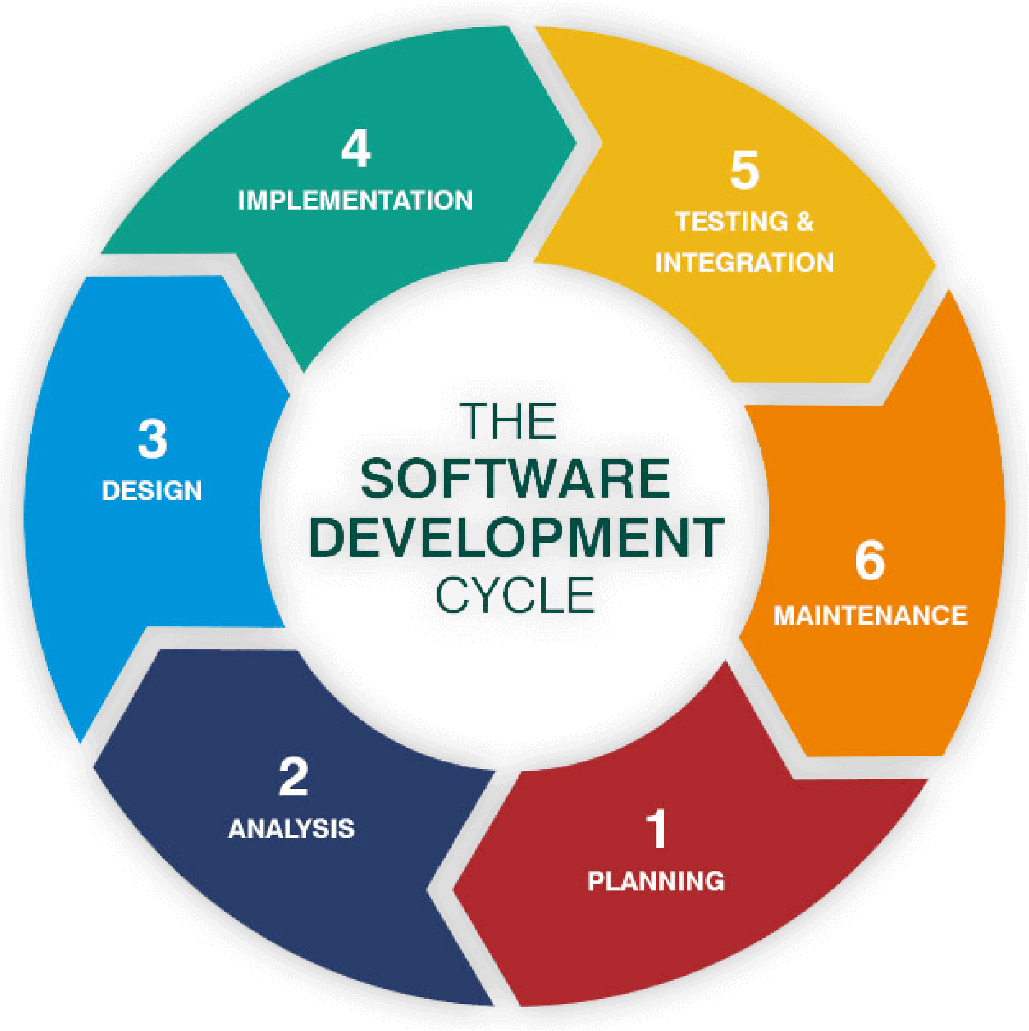
Software Testing

SDLC / STLC

Software Development Life Cycle

- When humans first started writing software, they had little idea how to do it.
- This started the field of software engineering.
- One of the outcomes was the software development life cycle

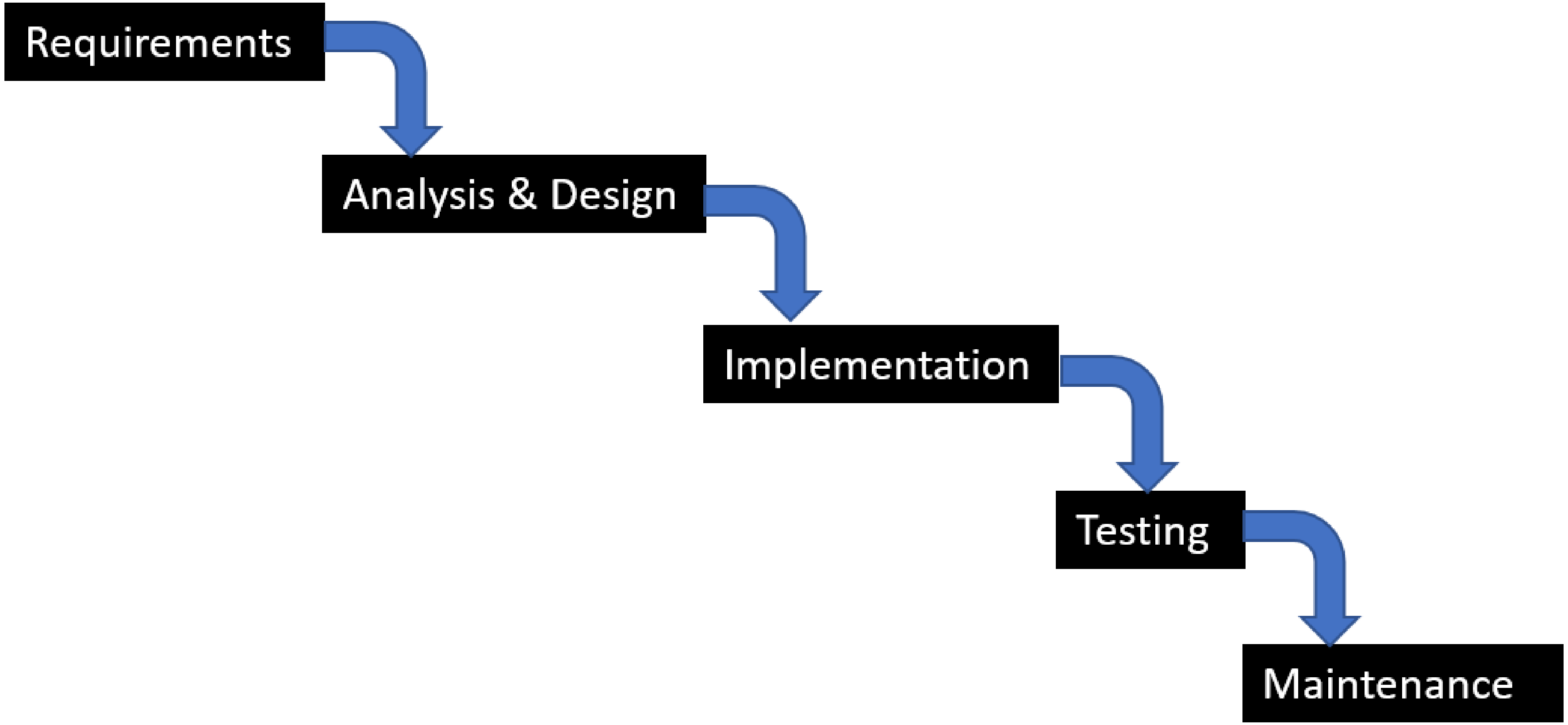
SDLC


THE
SOFTWARE
DEVELOPMENT
CYCLE
1
PLANNING
2
ANALYSIS
3
DESIGN
4
IMPLEMENTATION
5
TESTING &
INTEGRATION
6
MAINTENANCE

SDLC Phases

- **Planning** - plan the entire project
- **Analysis –** understand requirements to map software onto that set of requirements
- **Design –** design a software solution to meet the reqirements
- **Implementation** – build the software
- **Testing and Integration –** install into target environment and test
- **Maintenance** – adapt software to changing requirements and environment

Waterfall Model



Requirements

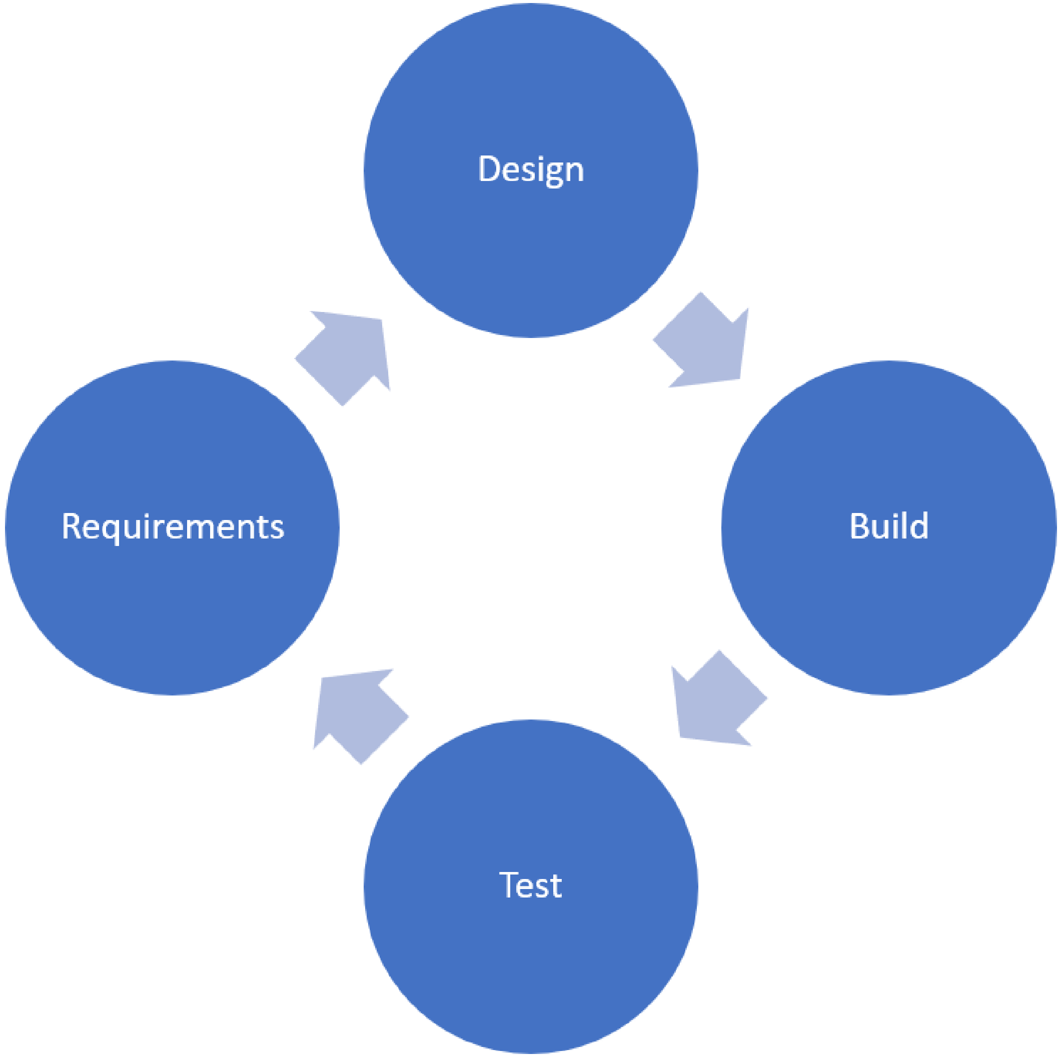
Analysis & Design

Implementation

Testing

Maintenance

Iterative Model



Agile Models

- Respond to change quickly
- Changing requirements are not a disaster
- Phases can happen simultaneously
  - Design can start once some requirements are finalized
  - Implementation can start once some design is finished
  - Test design can start once design is complete
  - Testing can start as soon as some coding is done
- When a problem is found
  - Go back and fix it!

Testing in the SDLC

- You can test
  - **During requirements** – test plan
  - **During design** – unit test design
  - **During implementation** – unit test implementation
  - **During testing**
  - **During Delivery** – acceptance tests
  - **During maintenance** – regression testing and testing new features

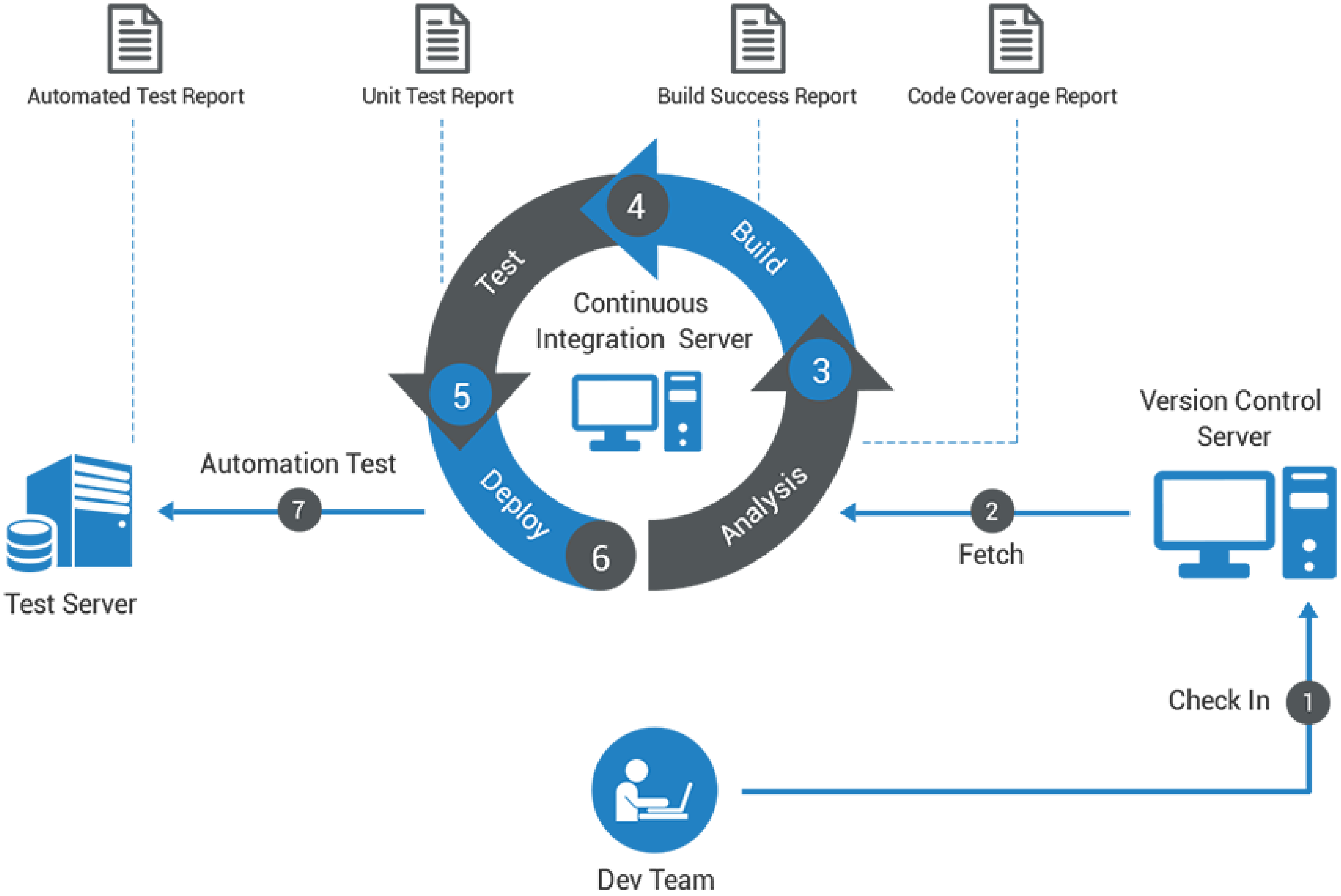
Incremental Development

- The core of the system is built
- Other parts are constructed and added into the core
- Testing is done as new components are built and integrated into the whole

Continuous Integration

- Working software is kept in a repository
- Developers
  - Check it out to add features or fix bugs
  - Test it to make sure it is bug-free
  - Push it to the repository
- The goal is to make sure that the code in the repository is bug-free

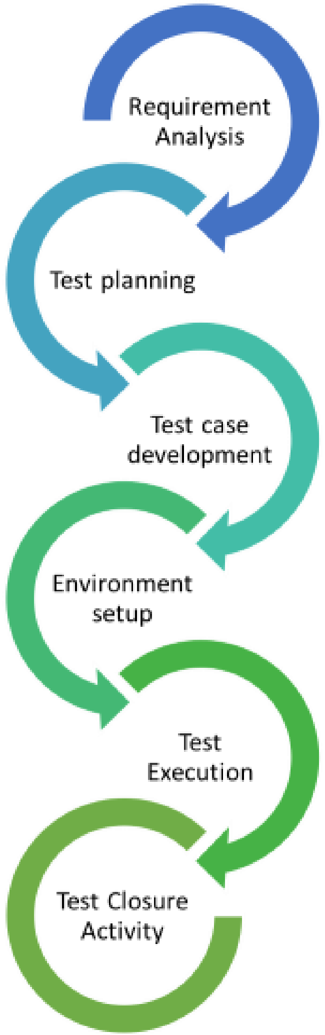
Continuous Integration



Test Driven Development

- writes the tests first and then the code to be tested
- test team usually gets together with the developers and
  - agrees on functions or classes and what they should do
- advantage of test driven development is that there are no surprises for the developers
- can be used as part of continuous integration.

The Software Testing Life Cycle



Requirement Analysis

Test planning

Test case development

Environment setup

Test Execution

Test Closure Activity

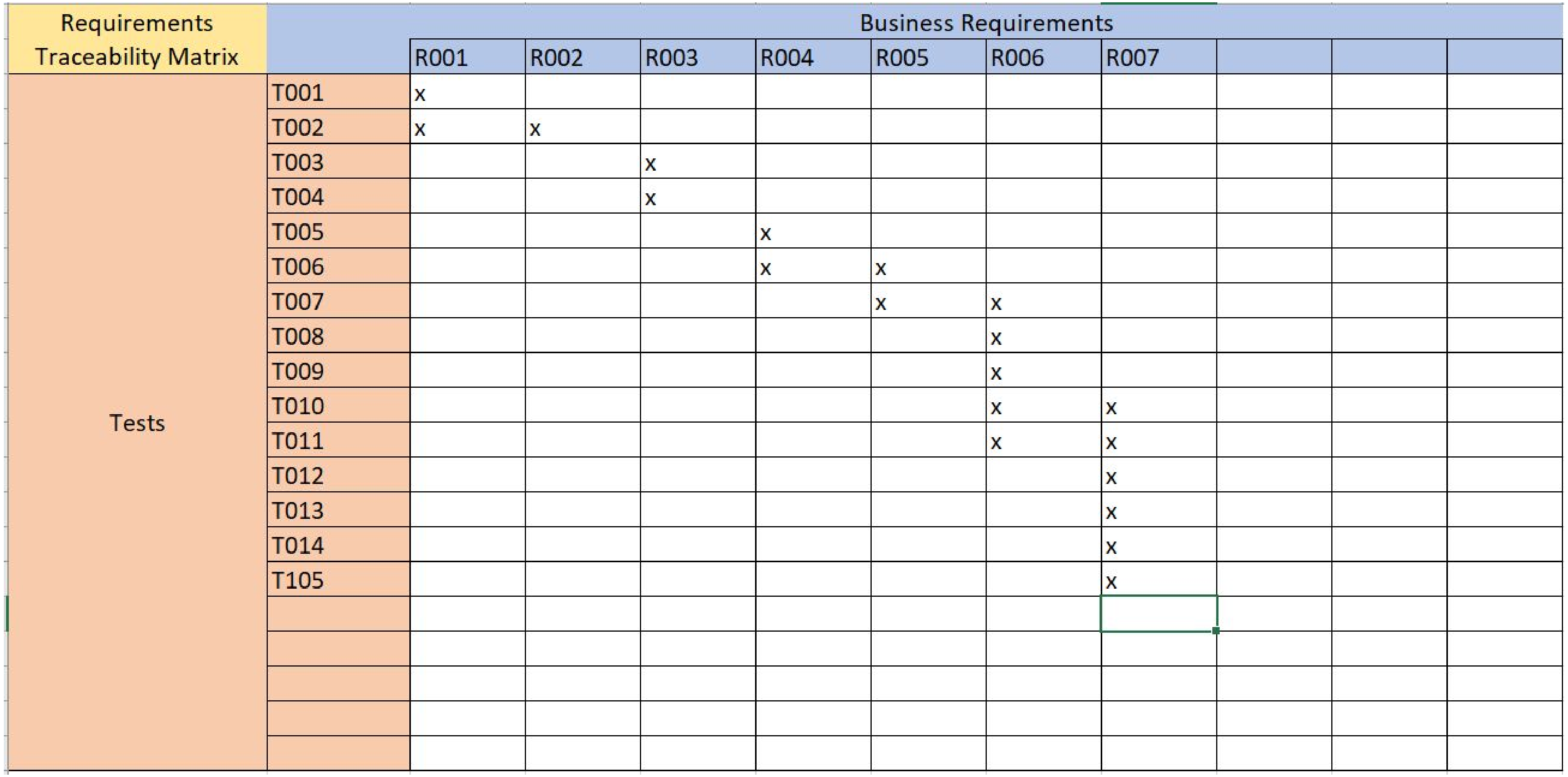
STLC Benefits

- The benefits of using the STLC include:
  - It provides a guide for the testing process, increasing efficiency and consistency,
  - It clearly defines the expectations of each part of the project,
  - It can provide time constraints on the testing,
  - Ensures that software meets requirements before more software is developed,
  - Ensures that all project requirements are met.

STLC Phases

- **Requirements Analysis** -- requirements are examined to identify their testable aspects
  - Entry: A set of requirements and acceptance criteria
  - Exit: Requirements traceability matrix and an automation feasibility report
- **Test Planning** -- produces the test plan document
  - Entry: the requirements analysis and requirements test matrix,
  - Exit: an approved test plan.
- **Test Case Development** -- the actual test cases are created
  - Entry: an approved test plan.
  - Exit: approved test cases, test data, and automation scripts
- **Test Environment Setup** -- the test environment is set up
  - Entry: system design and project architecture.
  - Exit: a functional test environment.
- **Test Execution** -- tests are deployed to the testing environment and executed
  - Entry: all of the exit criteria from the previous steps.
  - Exit: the tests have been performed and test reports generated
- **Test Cycle Closure** -- results in a report which summarizes the results of the tests
  - Entry: test results and reports.
  - Exit: test closure report.

Requirements Traceability Matrix



Requirements Traceability Matrix		Business Requirements									
		R001	R002	R003	R004	R005	R006	R007			
Tests	T001	x									
	T002	x	x								
	T003			x							
	T004			x							
	T005				x						
	T006				x	x					
	T007					x	x				
	T008						x				
	T009						x				
	T010						x	x			
	T011						x	x			
	T012							x			
	T013							x			
	T014							x			
	T105							x			