

Innovative Trends: AI for Software Engineering

Alexandru BRATOSIN

Outline

- **Introduction**
- Software Engineering
- AI in Software Engineering

Introduction

To sustain our world continuous technological evolution we require new and improved professional software.

In order to develop reliable applications, improved software engineering methods must be applied.

Our main topics will be :

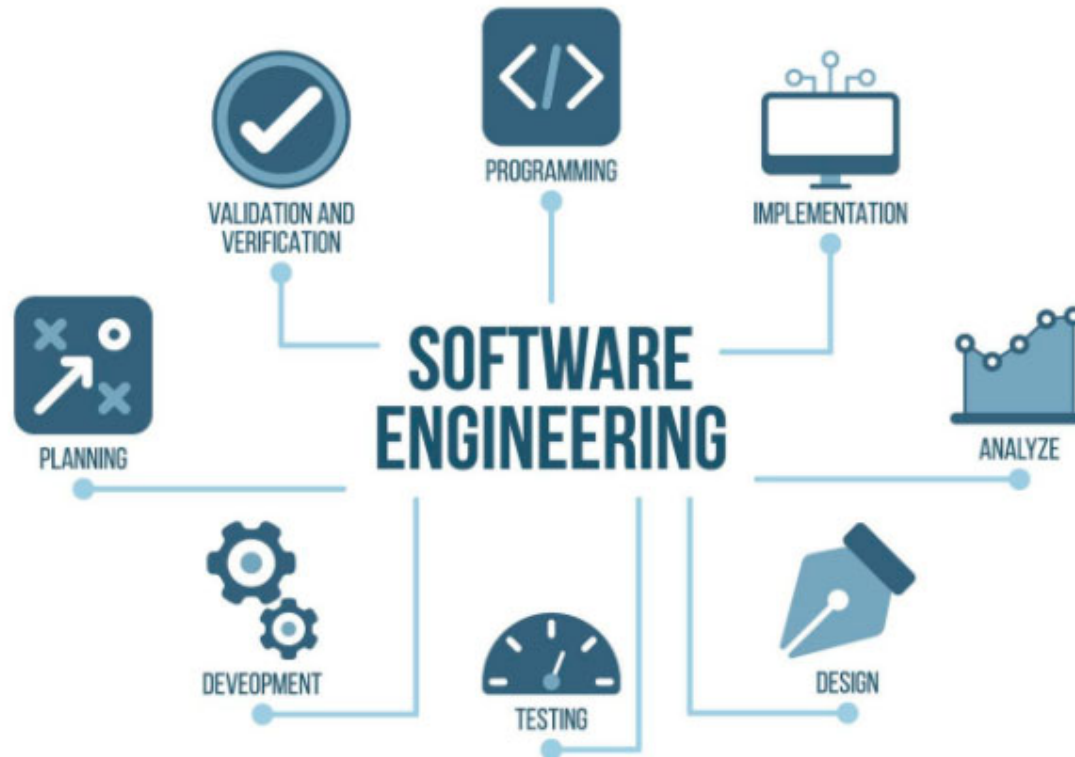
- Software Development Life Cycle
- Software Development Paradigm
- AI in Software Engineering.

Outline

- Introduction
- **Software Engineering**
- AI in Software Engineering

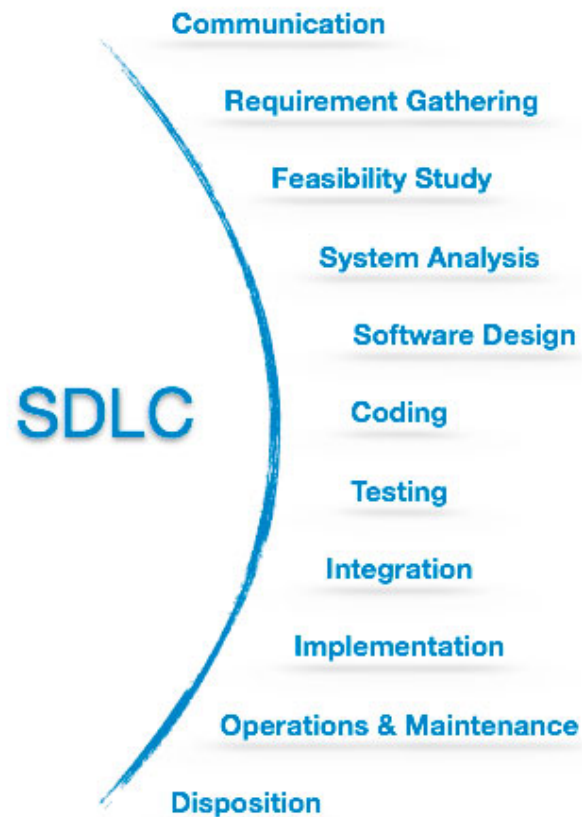
Software Engineering

Software engineering consists of applying well-defined scientific principles, methods and procedures while developing software applications.



Software Development Life Cycle

Software Development Life Cycle (SDLC) is a well-defined, structured sequence of stages in software engineering to develop the intended software product.



Software Development Life Cycle

The base SDLC can be divided in 7 main phases:

- **Phase 1: Planning:**
 - Identification of the system for development
 - Feasibility assessment
 - Creation of project plan
- **Phase 2: Analysis:**
 - Gathering business requirement
 - Creating process diagrams
 - Performing a detailed analysis
- **Phase 3: Design:**
 - Designing of IT infrastructure
 - Designing of system model

Software Development Life Cycle

- **Phase 4: Development:**
 - Development of IT infrastructure
 - Development of database and code
- **Phase 5: Testing:**
 - Writing test cases
 - Execution of test cases
- **Phase 6: Deployment:**
 - Integration & Implementation
- **Phase 7: Maintenance:**
 - Support the system users
 - System maintenance
 - System changes and adjustment

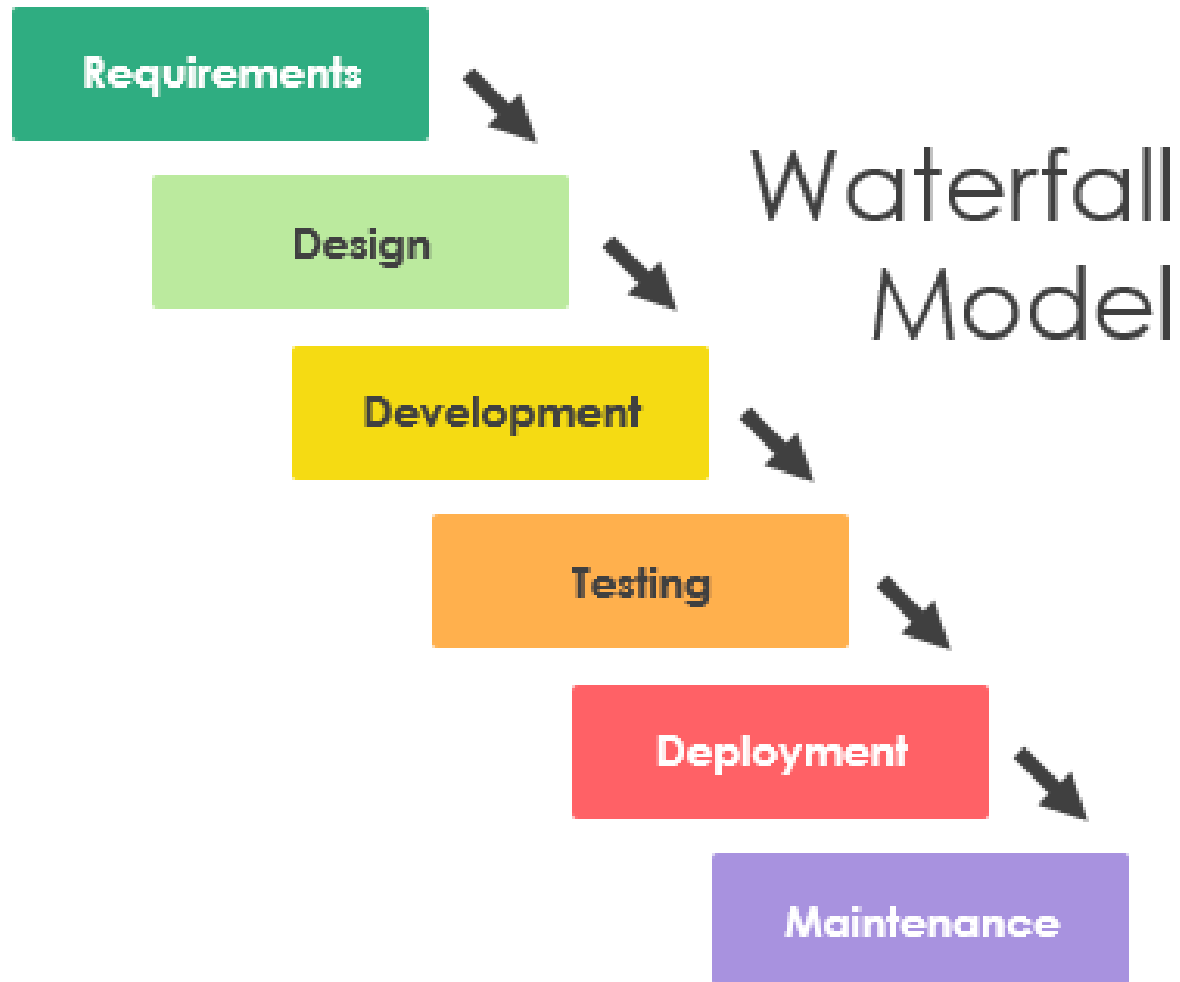
Software Development Paradigm

Due to the wide variety of software and client requirements, multiple types of paradigms were derived from the base SDLC model.

The most commonly used are:

- Waterfall
- Iterative
- V-Model
- Spiral
- Lean
- Agile
- Prototyping

Waterfall

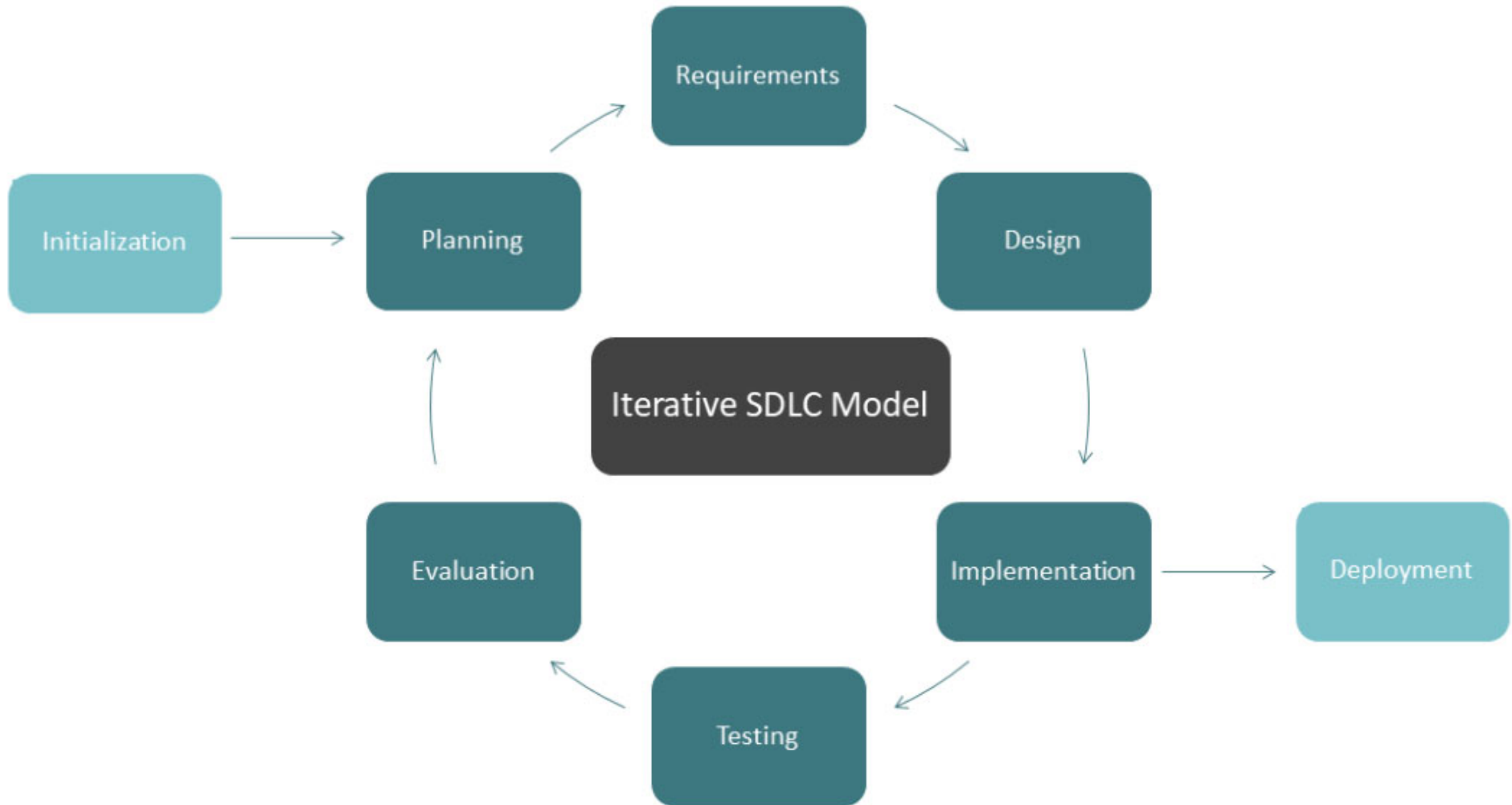


Waterfall

The waterfall model can be used when:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Iterative

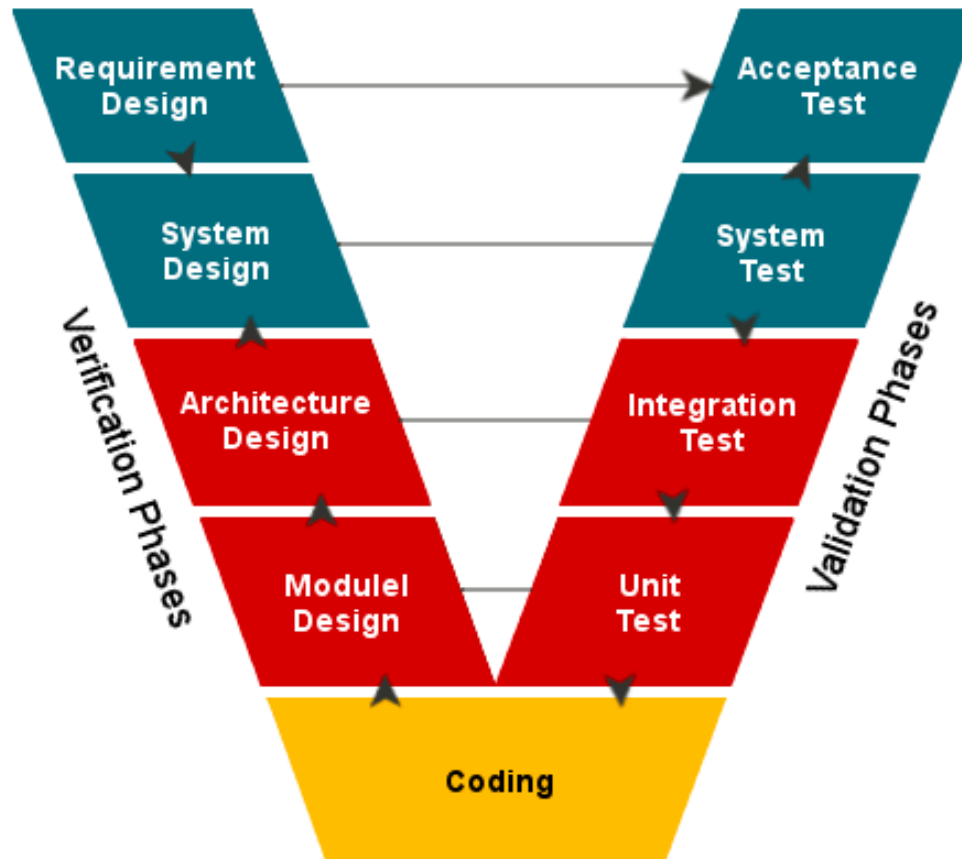


Iterative

Appropriate use case scenario :

- Requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by the development team while working on the project.
- Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which may change in the future.

V-Model

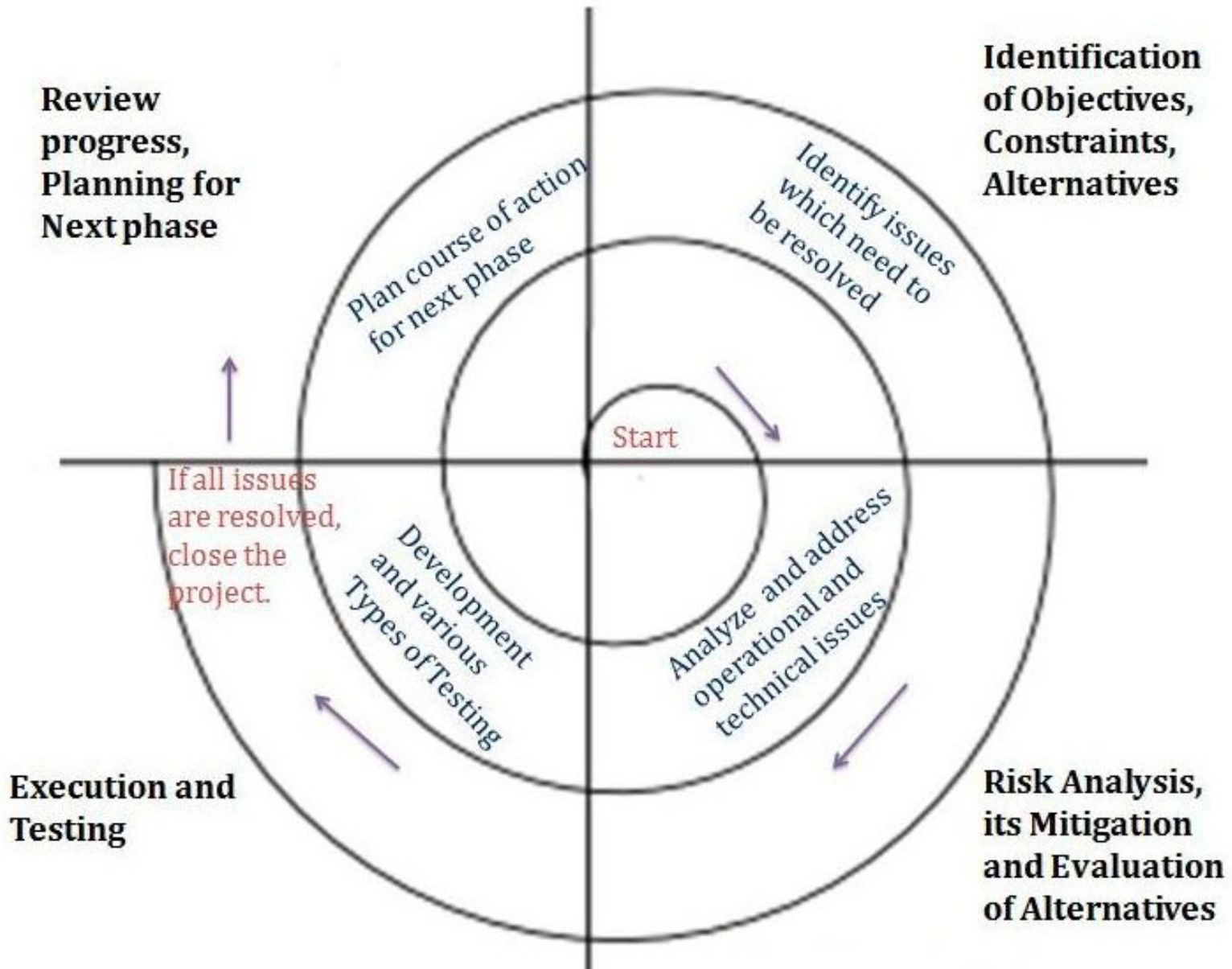


V-Model

The V-Model is best used when:

- Requirements are well defined, clearly documented and fixed.
- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short.

Spiral



Spiral

Typical uses of a Spiral Model:

- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

Lean



Lean

7 LEAN SOFTWARE DEVELOPMENT PRINCIPLES

01

**Eliminate
waste**

02

**Amplify
Learning**

03

**Decide as late
as possible**

04

**Deliver as fast
as possible**

05

**Empower
the team**

06

**Build
quality in**

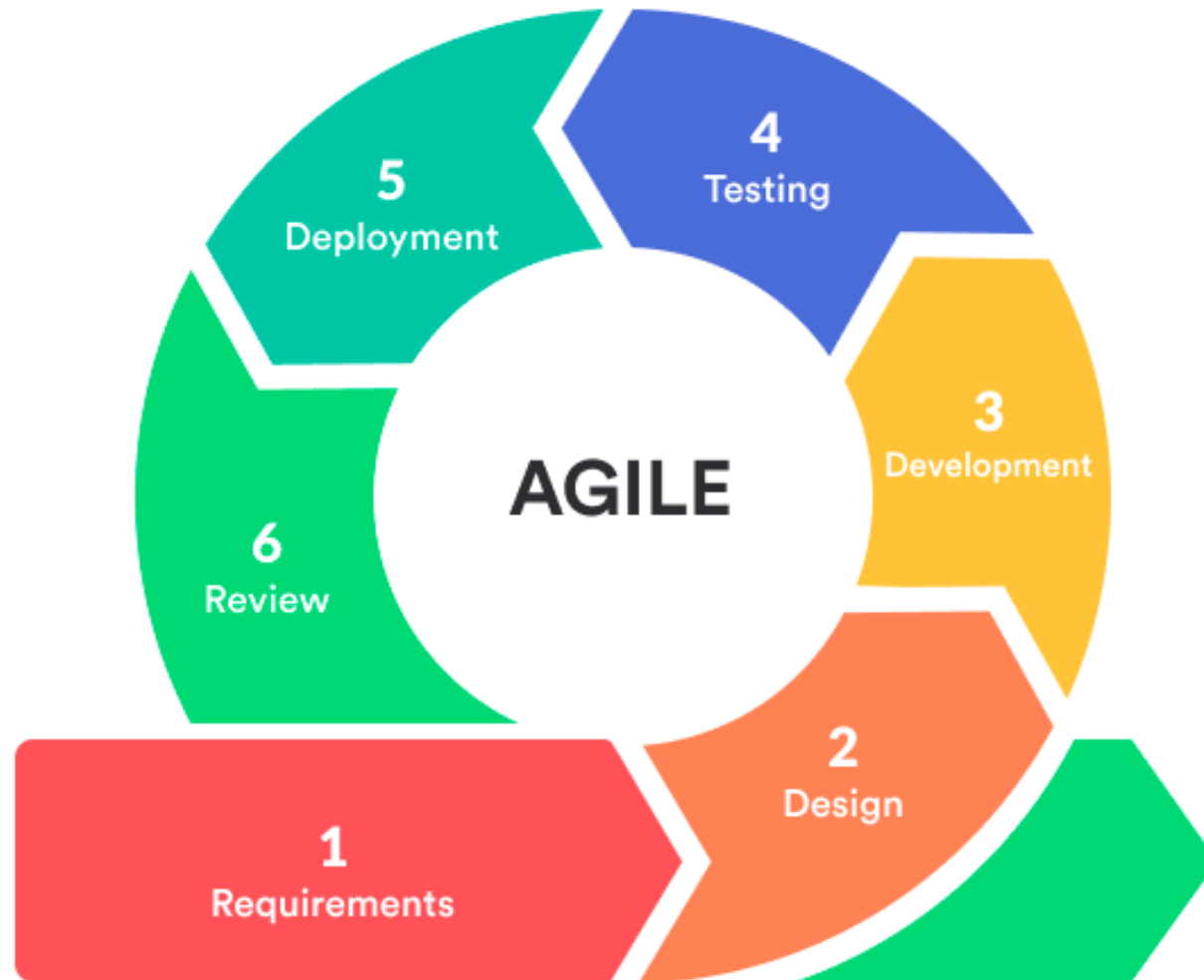
07

**See
the whole**

Lean

- The lean business model is designed to reduce waste in business processes.
- If an organization thoroughly integrates lean concepts into its operations, a likely outcome is a reduced need for cash, fewer errors, higher-quality products, and faster deliveries to customers.
- This approach works well for startup companies, which have little excess cash to invest, as well as for companies interested in improving their competitive positions.

Agile

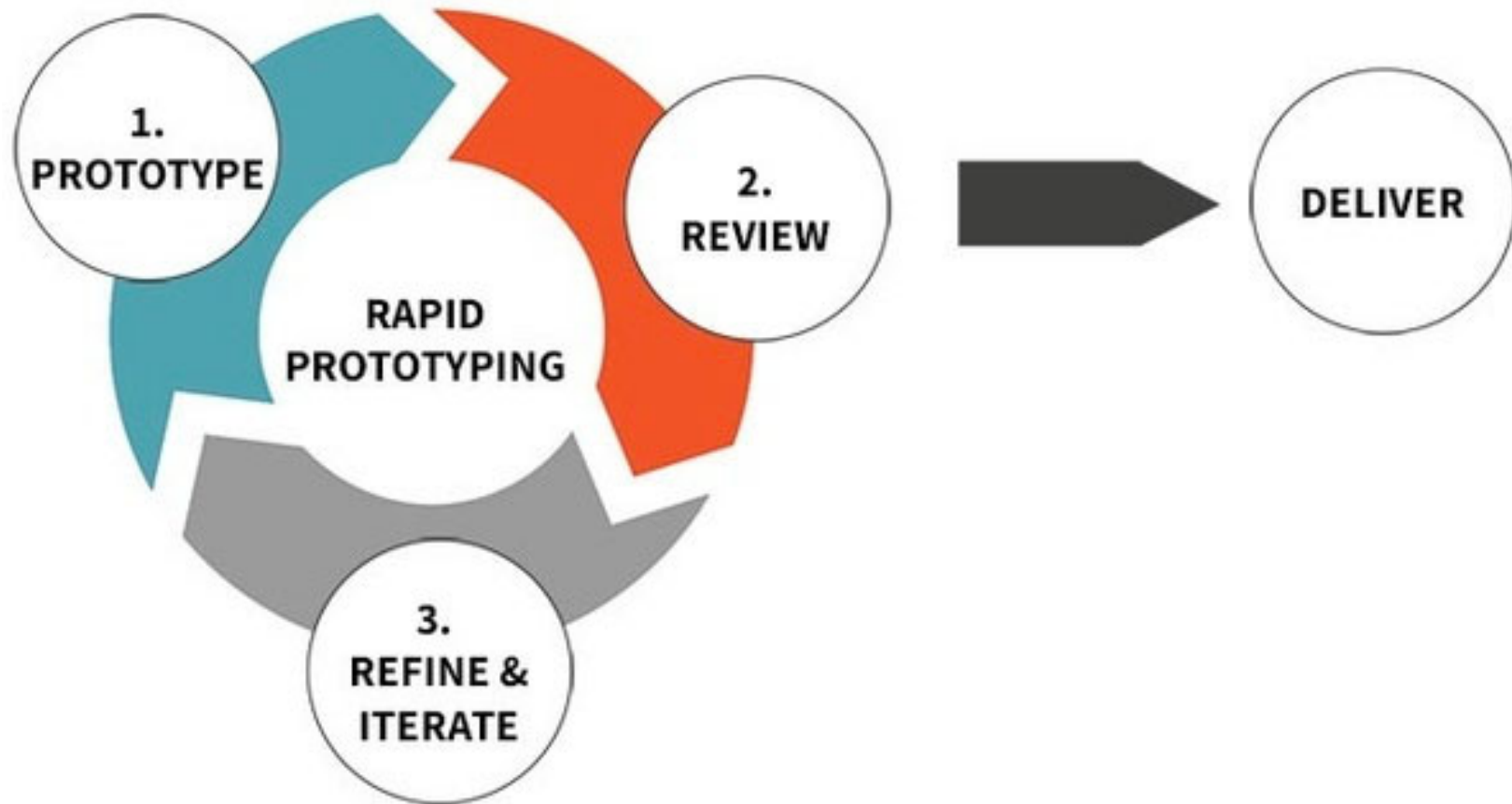


Agile

Agile is made for situations when there are:

- Unclear requirements/deliverables
- Level of participation/input and buy-in from stakeholders
- Cost of change is minimal
- Emphasis on teamwork, transparency and continuous improvement

Prototyping



Prototyping

- Software Prototyping is most useful in development of systems having high level of user interactions such as online systems.
- Systems which need users to fill out forms or go through various screens before data is processed can use prototyping very effectively to give the exact look and feel even before the actual software is developed.
- Software that involves too much of data processing and most of the functionality is internal with very little user interface does not usually benefit from prototyping.

Outline

- Introduction
- Software Engineering
- **AI in Software Engineering**

AI in Software Engineering

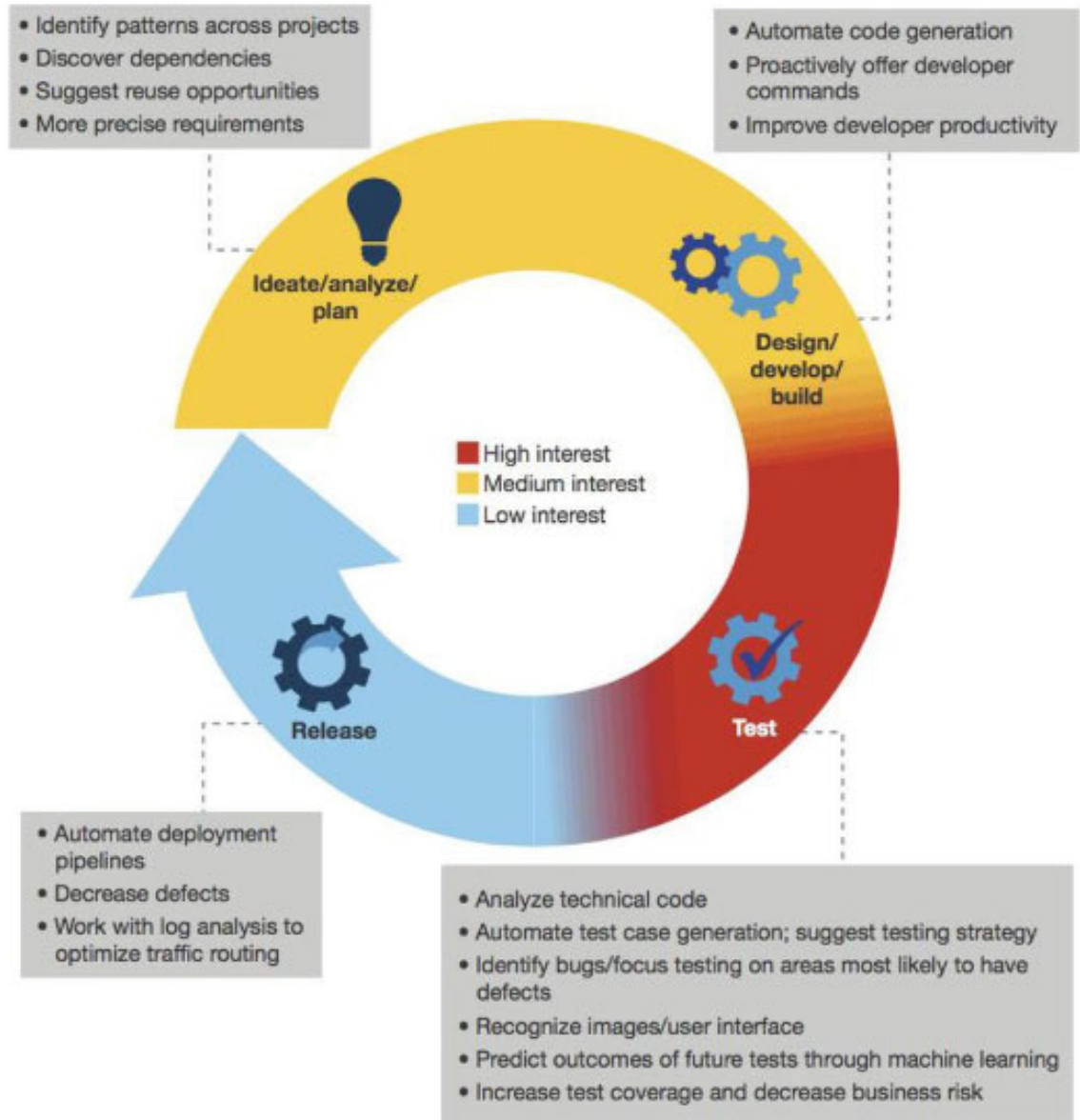
- As discussed above, each of the current methodologies used for developing software have their own utilization scenarios.
- Due to the continuous trend of innovation and development , the demand for software is increasing at a rapid pace, and present software engineering methods must be adapted accordingly.
- AI can be a perfect candidate to improve the speed and process at which the software is developed.

AI in Software Engineering

By training AI on previous project data, it can be used afterwards to help with some core project issues:

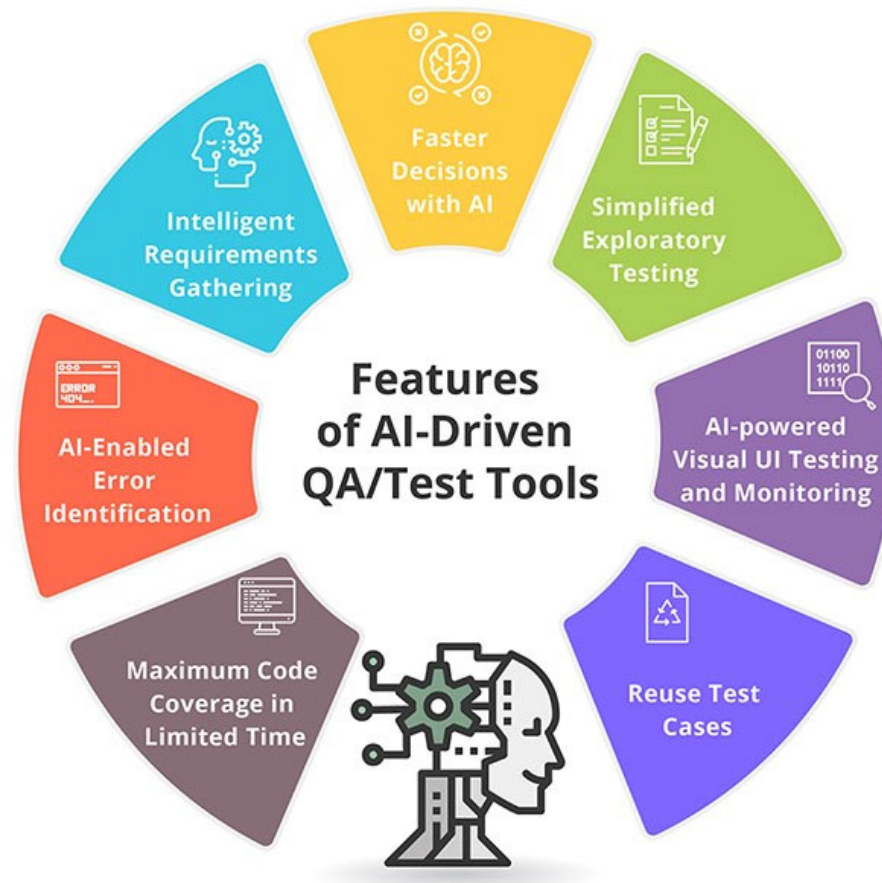
- Rapid Prototyping
- Intelligent Programming Assistants
- Automatic Analytics & Error Handling
- Automatic Code Refactoring
- Precise Estimates
- Strategic Decision-Making

AI in Software Engineering



AI in Software Engineering

- The quality assurance process can be aided as well by AI technology. To diminish the effort required by developers to test, identify errors, bugs and fix them.



AI in Software Engineering

Examples of AI that can help the development process:

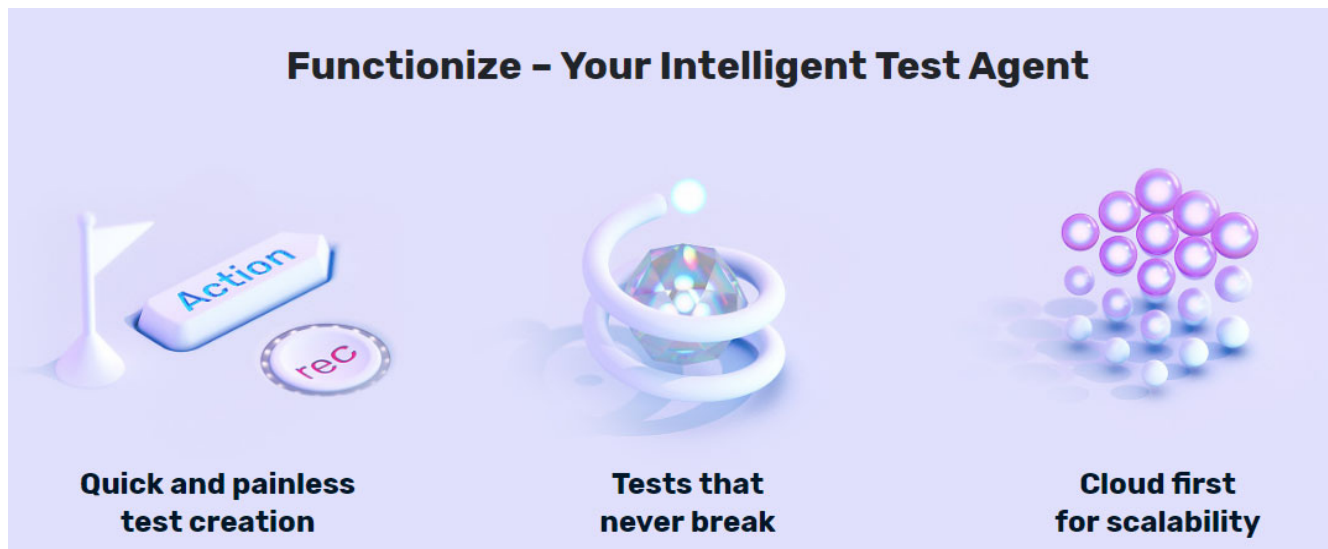
- Applitools
- Functionize
- SapFix
- DeepCode
- Seerene and HPI AI.

AI in Software Engineering

- AppliTools is a GUI tester tool empowered by AI. The AppliTools Eyes SDK automatically tests whether visual code is functioning properly or not.
- AppliTools enables users to test their visual code just as thoroughly as their functional UI code to ensure that the visual look of the application is as you expect it to be.

AI in Software Engineering

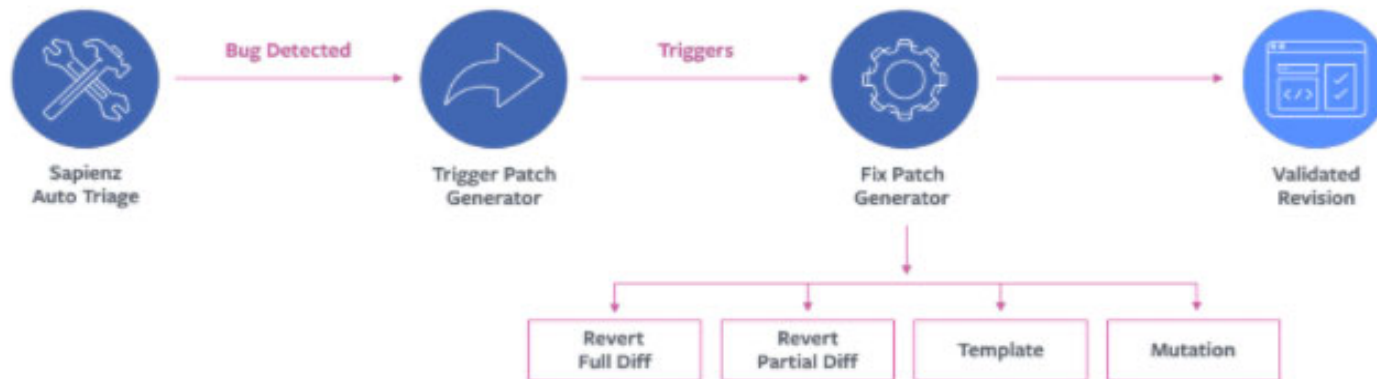
- Functionize - AI-Powered Testing for Agile Teams
- Dramatically reduce time to market
- Reduce the time taken for test creation, maintenance, and analysis.
- No more maintaining test infrastructure.
- Seamless CI/CD integration.



AI in Software Engineering

SapFix is a solution developed by Facebook.

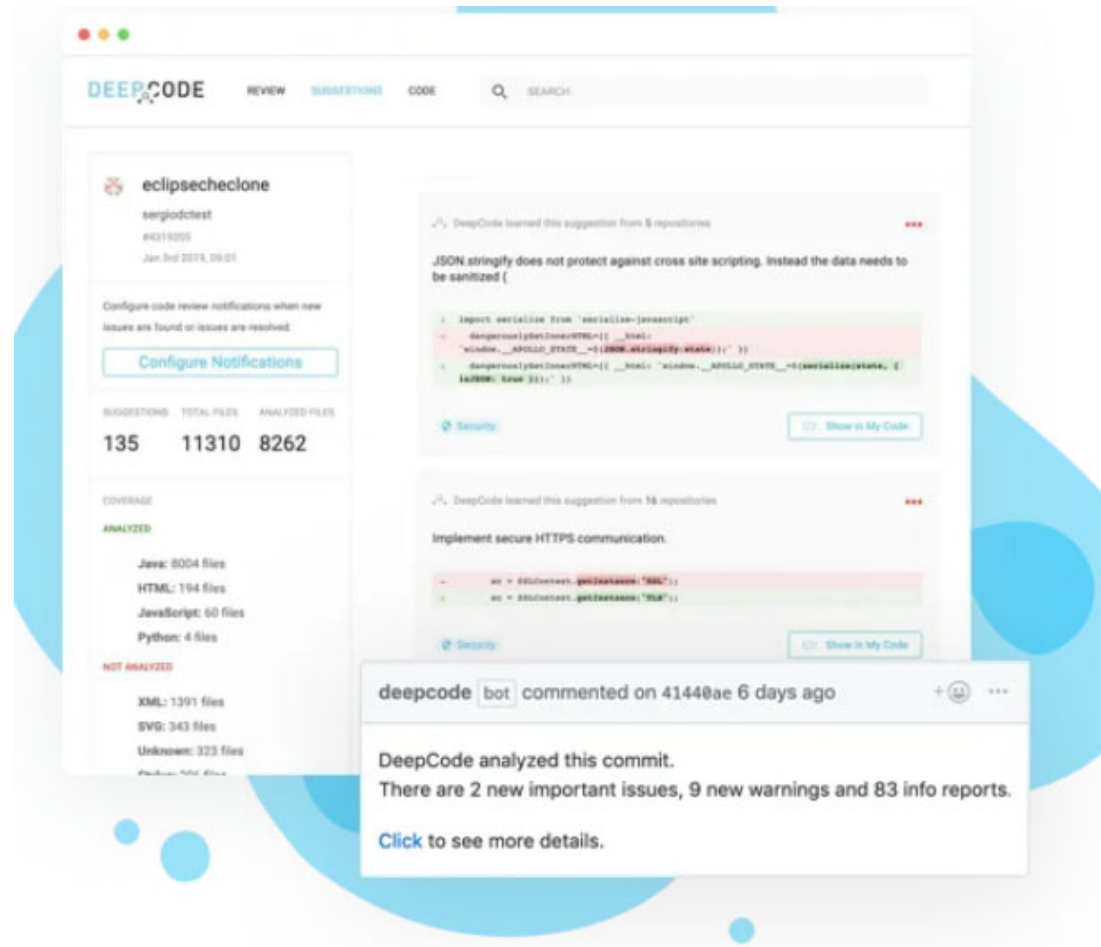
- For simpler bugs: SapFix will create patches that revert the code submission that introduced these bugs.
- For complicated bugs: The tool uses a collection of “templated fixes” that were created by human engineers based on previous bug fixes.
- If human-designed template fixes aren’t up to the job: The tool attempts a “mutation-based fix,” which works by continuously making small modifications to the code that caused the software to crash, until a solution is found.



AI in Software Engineering

DeepCode is a real-time semantic code analysis powered by AI

- Find more critical issues and security vulnerabilities
- Get results immediately
- Benefit from a self-growing knowledge base
- Learn from example fixes of real commits



AI in Software Engineering

Seerene and HPI research AI approaches on the following topics using partner projects:

- AI-based analysis for the deployment planning of development teams
- AI-based computation of source code complexity, technical faults and software quality
- AI-based approaches for the transformation of legacy systems and vendor steering
- AI-based identification of duplicates and clones
- AI-based risk and failure analysis
- AI-based benchmarking
- AI-based statements on expenses and costs

Conclusions

- In order to ease the software development process and to reduce the actual workload of team members AI technology is required.
- AI can be used either during the planning process, development, testing, requirement gathering or design.
- AI will be in an everchanging state improving step by step with each project analyzed and used on.
- The future of software engineering will be led by a combined effort of AI and humans.