Importance of IT Security in Application Development

Importance of IT Security in Application Development

- Historically IT security wasn't implemented into software development.
- We wear different hats
  - IT security professionals aren't always programmers
  - Programmers don't focus primarily on IT security
- Security flaws in software can lead to major risks to your organization:
  - Missing data encryption
  - SQL injection
  - Buffer overflow
  - OS command injection
  - Cross-site scripting
  - etc.

target

$162 Million

THE HOME DEPOT

$56 Million

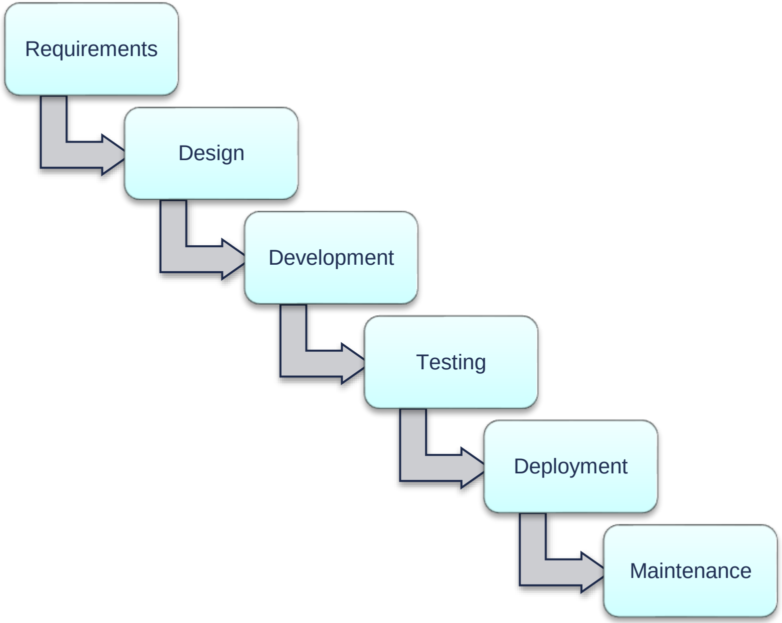
Software Development Lifecycle (SDLC)

Software Development Lifecycle (SDLC)

- Standard business practice for building software applications
  - An end-to-end lifecycle developing applications
  - A project management methodology
- Three commons SDLC models:
  - Waterfall
  - Agile
  - DevOps
- How do we secure the SDLC?
  - The Secure SDLC Framework

Waterfall Model

- A sequential and linear application development process.
- Each phase must be completed before the next phase can begin.
- Phases don’t overlap.
- Earliest and widely-used model.



Requirements

Design

Development

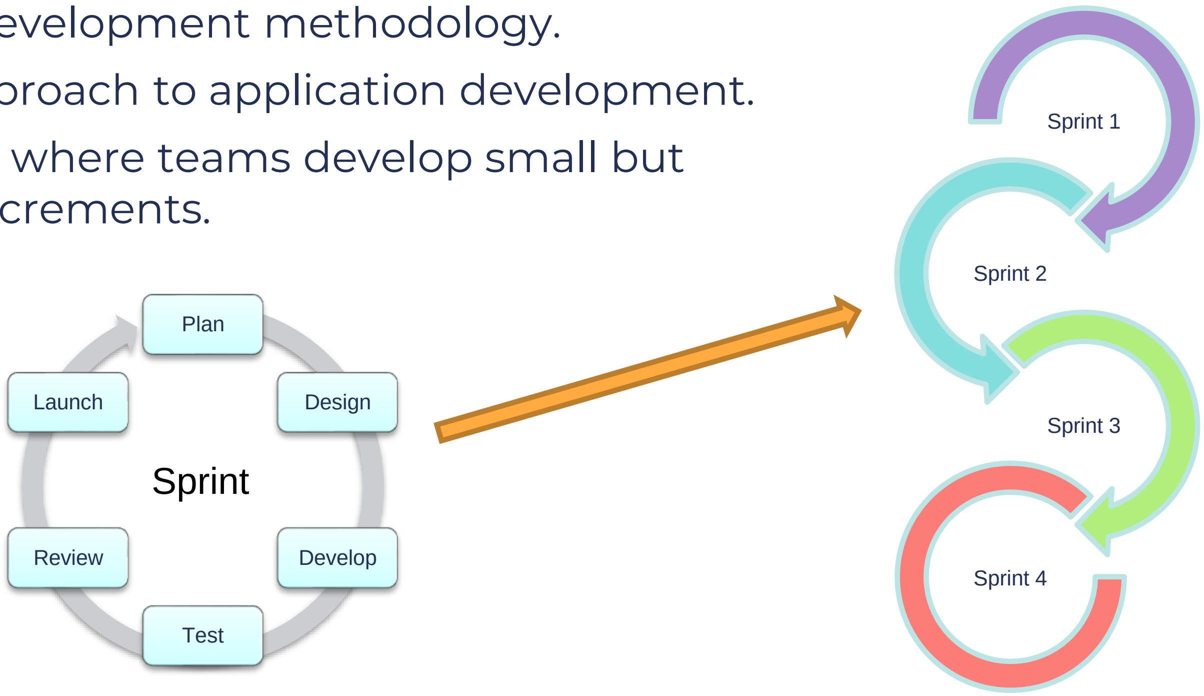
Testing

Deployment

Maintenance

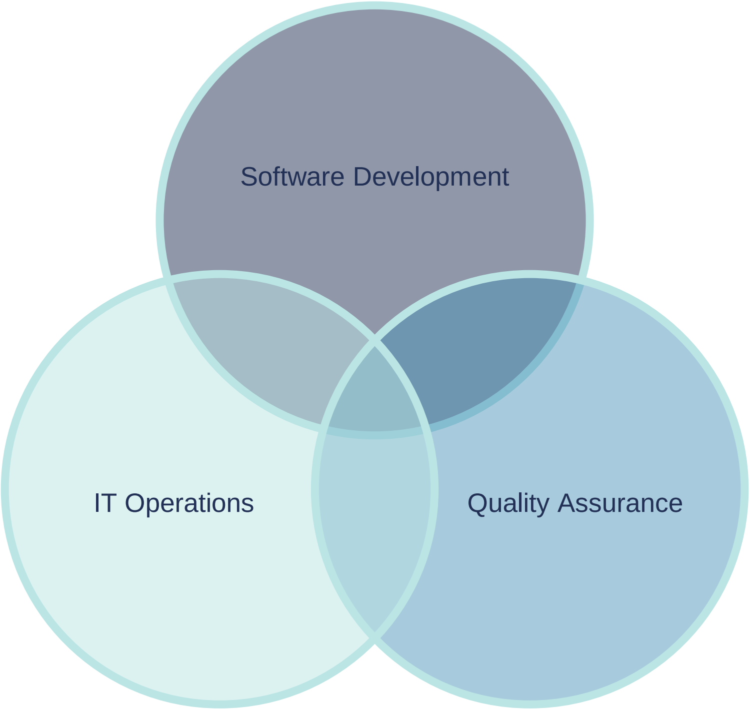
Agile Model

- More flexible development methodology.
- An iterative approach to application development.
- Utilizes sprints, where teams develop small but consumable increments.



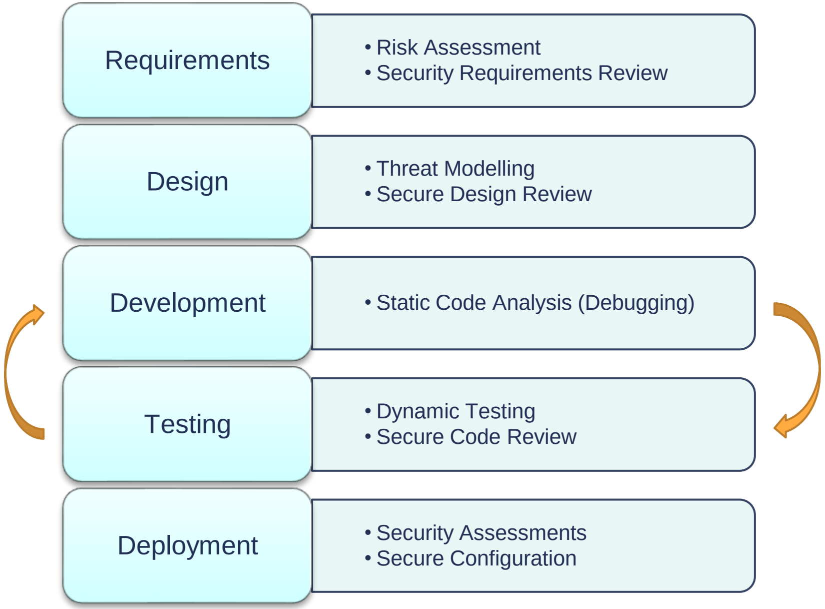
DevOps Model

- Integrates software development, quality assurance (QA) and IT operations teams.
- Historically these various roles worked in silos, slowing down the application development and deployment process.
- DevOps aims to decrease the time required to develop, test, and deploy software.
- DevOps teams aim to have these teams work collaboratively throughout the SDLC Lifecycle.



Securing the SDLC

- A secure SDLC involves integrating security requirements and testing into existing SDLC processes.



Requirements
- Risk Assessment
- Security Requirements Review

Design
- Threat Modelling
- Secure Design Review

Development
- Static Code Analysis (Debugging)

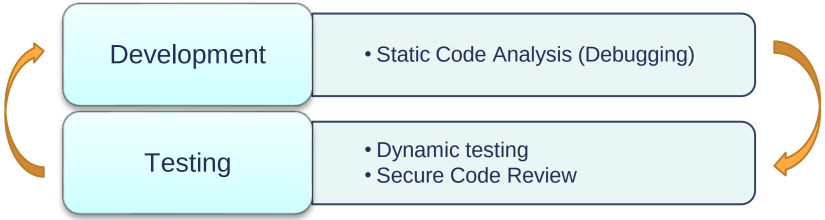
Testing
- Dynamic Testing
- Secure Code Review

Deployment
- Security Assessments
- Secure Configuration

Static and Dynamic Testing

Static and Dynamic Testing

- **Static code analysis** tests the code passively when it's not running.
  - Non-runtime environment.
  - Looks at the code from the inside out (debugging).
- **Dynamic testing** tests the code while executing it.
  - Looks at the application from the outside in.
  - Examines its running states, trying to manipulate it to discover security vulnerabilities.
  - Simulates attacks against an application and analyzes the application's reactions, determining whether it's vulnerable or not.



Development

- Static Code Analysis (Debugging)

Testing

- Dynamic testing
- Secure Code Review

Authorization to Operate (ATO)

Authorization to Operate (ATO)

- Many governmental organizations require an **ATO** before an application can be deployed into production.
  - Private companies and other organizations also use ATOs.
- With an ATO, an authorizing official (AO) or governance body reviews the application’s security authorization package.
  - The package typically includes an executive summary, security plan, privacy plan, security control assessment, and privacy control assessment.
- An ATO signifies completion of an objective third-party evaluation and acceptance of any residual risk of the application to the organization.