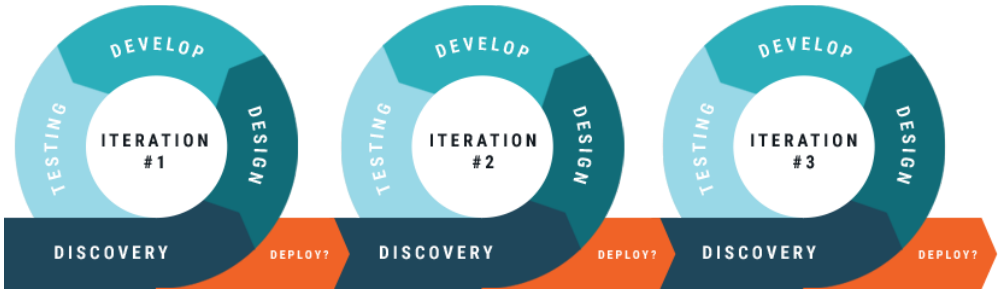
Software Development Life Cycle (SDLC)

SDLC Overview

The Castle engineering team uses an agile software development process. Our engineers work in short iterative sprints which comprise of discovery, design, development, testing and release phases. This allows us to release features quickly with confidence and iterate quickly to improve features over time.



Discovery Phase

The discovery phase comprises of product feasibility and requirements gathering and is typically performed by the VP of Engineering, engineering managers and tech leads.

Design Phase

During the planning and design phase high level designs, user stories and UI mockups are created. Stories are broken up into tasks and estimated for complexity and effort which is measured in time (days). All planning and tracking is performed on JIRA.

Development Phase

During the development phase, engineers choose and implement tasks from the backlog. Engineers rely on peer code reviews and pull requests to ensure the code changes are correct and the quality of code is high. Pull requests are not merged into the main branch until the code reviewer has explicitly approved the change. We maintain all of our code in Github and run integration using CircleCI.

Testing Phase

Engineers write unit tests and integration tests during this phase. Changes are tested locally and verified to work in development environments before releasing to production. We use various testing tools for developing test cases.

Deploy Phase

Engineers push their changes to production with assistance from the Engineering managers when needed. Production pushes are coordinated and managed via Jira, GitHub and CircleCI tools. All deployments are to be done in accordance with Castle's change management policy.

Post Release Phase

We monitor all deployments post release to ensure stability and performance. Our engineering team rotates on-call and are responsible for fixing or rolling back any issues, if they ever occur.

Security

Developers are expected to adhere to Castle's coding standards throughout the development cycle, including standards for quality, commenting, and security. At a minimum, developers are expected to address the common security issues in the OWASP top-10 (www.owasp.org) in the course of their design, development, reviewing, and testing efforts.

Developers performing peer code reviews are required to have taken requisite security training and are to examine the new or revised code and provide feedback. Review must confirm that the code does not violate any security principles or design objectives.

In-scope software must be subjected to standardized tests that include both functionality and security testing. Any security issues detected during testing must be addressed prior to release.

Actions taken by developers on production systems must always be logged and audited.

Access to the private source code repository (in Github and elsewhere) is restricted to authorized personnel, and access controls are enforced to maintain the security of the repository. Developer access to the source code repositories must be approved by appropriate Engineering Management prior to granting any new or additional access rights. All access to source code or changes in access rights to the source code repositories must be logged and is subject to audit.

To satisfy security requirements, following are the guidelines for features and application requirement process:

Establish application security evaluation guidelines using criteria such as:

- Data handling, exposure, and behavior
- External security and compliance requirements
- Interactions with other systems
- User account and privilege management
- Customer security requirements
- Known or expected security weaknesses or exposures
- Security challenges resulting from any unique or non-standard architecture