DATASCAN

Wi Software Development Life-Cycle

May 27, 2020

By:
Michael Hearn, UI/UX Development Manager
Pavel Stanovskyi, Sr. Quality Engineering Manager
Ben Traynor, Sr. Development Manager

415 South Broad Street Alpharetta, GA 30009

In software engineering, the software development life cycle (SDLC), also referred to as the application development life-cycle, is a process for planning, creating, testing, and deploying a software development application. At DataScan, we follow the following six core stages in this cycle:

1) Planning and Requirements Gathering
2) Requirement Analysis
3) Design
4) Development and Implementation
5) Testing
6) Deployment and Delivery

Stage 1 - Planning & Requirements Gathering

The Product Team at DataScan maintains a product backlog, which is a catalog of requests from our Clients, bugs found internal and externally, product roadmap initiatives, technical initiatives, and security findings. Items are pulled from the backlog and prioritized via a process called "Backlog Grooming". As new items are added, frequent grooming occurs, and priorities shift.

The Product Team will then schedule the prioritized backlog items over our core release versions, which occur twice a year. They will then work to create high-level requirements in the form of use cases and business requirement documents. These items will then be broken down even further into Release Candidates (RCs) based on the requirement analysis to allow short term milestone checkpoints for review and flexibility and agility around unforeseen changes that may be needed.

Stage 2 - Requirement Analysis

Now that the product management group has defined high-level use cases and requirements for the *'what'*, what we need to build/change, the technical teams are pulled into the cycle to define the *'how'*, how we will build/change it. During this requirement analysis phase, IT, QA, and the Development teams work together to discuss the technology decisions. For example, what are the security implications of the change, what are the hardware requirements and considerations, what performance implications will the change have, and what functional tests need to be built? Next, a unified decision for a release is formed based on the build time and resources required. The bucket of work defined in each RC is well-established, and the capacity within the team is known. Filling an RC bucket is now similar to playing Tetris, where we fit blocks of features into a container of time.

Stage 3 - Design

During the design phase, work items are broken down further. During this phase, the features are broken into detailed user stories or bite-sized testable pieces of the application, and teams are assigned for each story design. Meetings begin with the stakeholders and implementers, which includes Product Management, QA Engineers, IT, and Developers, all assigned to the specific feature. The team documents all the details of the feature from how it should look and function, to how it will be tested and deployed. Once all the details are known, a level of effort called a story point is assigned. A story point is a numerical measurement we use to determine how many stories the team can complete in a given time frame.

Once we know the story points for a feature, we allocate them to a work sprint. At DataScan, we define our sprints as two weeks; the development and QA engineers agree and commit to completing the selected stories in that two-week time frame. After every two or three working sprints, we follow up with a stabilization sprint. In this sprint, we stop adding new stories and ensure all loose ends are wrapped up, and testing is finalized.

Once a feature is complete, it is presented internally in a cross-company demo session. In this session, we bring the Development, QA, and the Product Management teams together with DataScan Management, Support, and Account Managers to give a demonstration of the new features for review and feedback. The feedback is included in the development life-cycle for future revisions and is one of the ways we ensure continuous improvement at DataScan.

Stage 4 - Development and Implementation

During our implementation stage, we adhere to strict secure software coding and QA standards and testing. We do so by conducting peer reviews, unit tests, and by using a code analysis tool. The following steps are performed during this phase:

Peer Reviews

- Code is reviewed by one or more additional developers before executed into the codebase
- New code is analyzed to ensure it meets the team standards of maintainability, quality, and adheres to the user story
- Analysis of the code changes prevents potential security issues
- Verify that the appropriate unit tests have been created to include the changed/added functionality

Unit Testing

- Programmatic tests were created by the development team that runs during our code build process to gate check that the code works as intended

- Low-level tests allow future changes to a section of code to be validated ensuring that it still functions as the original design intended

Static Code Analysis

- A tool utilized against the code for quality and security checks
- Identifies potential security vulnerabilities and possible programmatic issues
- Helps safeguard code from individual developer bias, human error, and unwanted security holes

Stage 5 - Testing

Once a feature/story is functional and can be deployed successfully to our QA lab, it is ready for additional testing. We utilize a continuous integration environment where we automatically deploy our application several times a day, which allows us to test the latest code changes in smaller bite-size pieces. It also allows us to isolate defects to recent changes, resulting in quick diagnostics and resolution.

An automated regression test is performed after the application is deployed to the test environment. This suite of tests covers the entire system, including accounting & processing, system configuration, loan transactions, etc. We also have robust test automation for our new application programmatic interface (API), which helps us accelerate testing, increase efficiency, and ultimately identify software defects. Once the defect is resolved, it will be re-tested and closed. This process continues until the software is stable and working according to quality standards and business requirements.

Before release certification, we utilize a third party to conduct a dynamic penetration assessment/test to ensure the security of our application and infrastructure. This dynamic penetration test complements the automated static security testing executed as part of the implementation phases.

Stage 6 - Deployment and Delivery

Once all development and testing have been completed, internal teams and core stakeholders will conduct a release signoff. The release is certified and deployed to a pre-production test environment before being made available for Clients. Starting with 2020.01, the deployment will be automated, mimicking our QA repeatable process conducted during testing. Due to the complexity and detail of our code, the automated deployment allows for more consistent and accurate delivery to our clients. Additionally, the automation will allow us to identify repeatable steps needed for Client-specific environments and supported configurations.

Maintenance Patches

After releasing a core version of the product, monthly maintenance patches are scheduled. These maintenance patches include non-invasive high priority bug fixes that have no or

minimal impact on existing functionality, security updates, and findings to our infrastructure and software components. New features or major functionality changes are not included in maintenance patching, and will only be addressed in core releases. Today's cybersecurity threats are both dynamic and sophisticated, and new vulnerabilities are discovered almost every day. Routine monthly patching allows us to act quickly in addressing these concerns.

Every monthly maintenance patch undergoes the same rigorous automated testing as our core releases, through our continuous deployment process and targeted testing on the resolved finding before it's available. Our maintenance releases are cumulative, so the third patch on a core release will include everything from patches one and two.

Clients are highly encouraged to follow a monthly maintenance patch routine associated with their core version. We want to ensure Clients have the safest and most stable version of our product for the best experience. The DataScan Support Team works directly with Clients to establish both core upgrade strategies and monthly maintenance patching execution.