Deployment Process

For Rollins Contributors

Rollins
MARKETING TEAM

This document describes the processes and best practices used by Rollins Team contributors. These processes ensure that Orkin.com content created is consistently and accurately deployed.

Use these processes to successfully deploy Orkin.com content.

In This Document

About Software Change Management

Software change management is the discipline of tracking and controlling changes during the software development and deployment process. Some aspects of this discipline include:

Change Control

Change control is a process that ensures changes to software are monitored and controlled, to avoid arbitrary or unapproved changes that may result in disruption of service, errors in the software, or overwriting of code contributed by other developers.

Build Management

A build is the result of converting software code into software artifacts. These artifacts are functioning software tools or, in the case of Rollins Marketing, functioning websites.

Build management ensures that the process and tools are properly implemented and result in quality builds.

Process Management

Process management ensures that all contributors adhere to the established software development process.

Team Building

Promoting a supportive team environment facilitates effective interactions within the software development process.

Defect Tracking

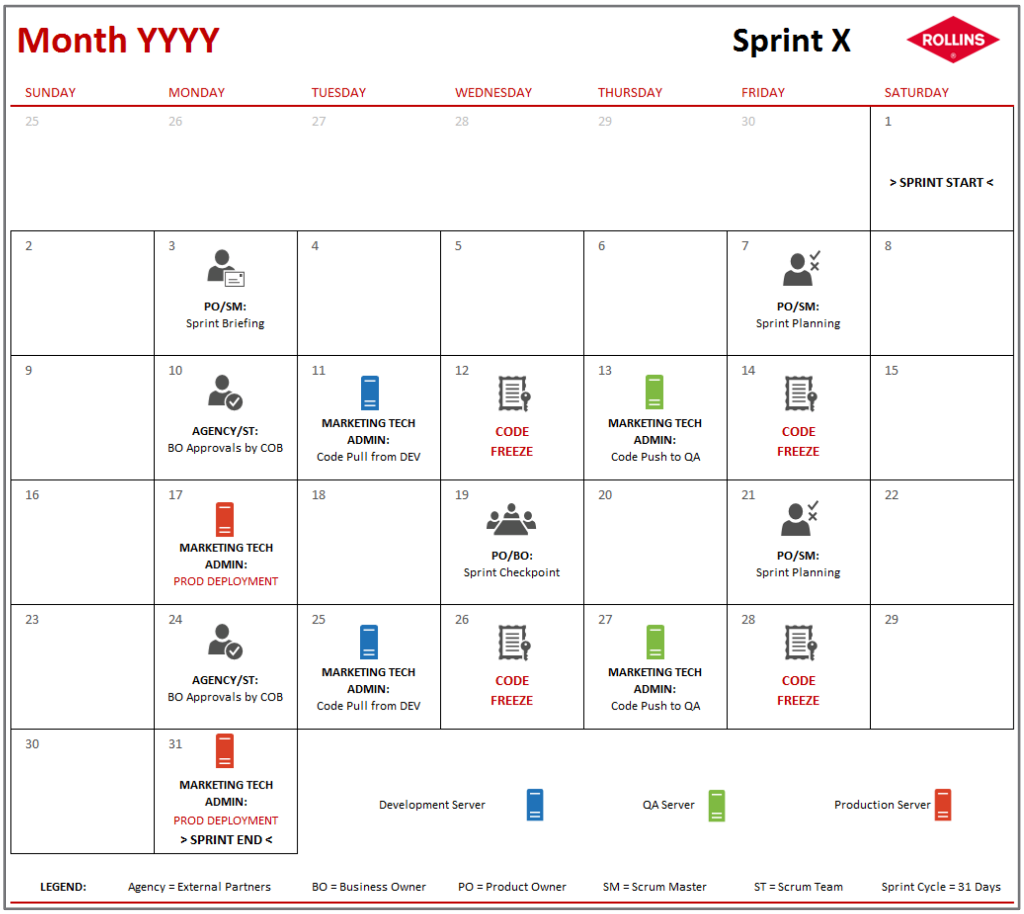
Defect tracking discloses defects and traces them to the source, ensuring defects are mitigated in the future.

Applied Change Management in Rollins Marketing

Rollins Marketing adheres to industry-standard change management concepts. The following section outlines some specifics of the Rollins Marketing process.

Rollins Marketing Sprint Cycle

The Rollins Marketing Technology team and all contributors adhere to a defined Agile sprint cycle. Sprints are four weeks, and changes are deployed to production every two weeks.



Requesting Off-Cycle Deployments

Because the sprint cycle allows two production deployments per month, off-cycle deployments are rarely authorized.

In the event of a crisis that requires an off-cycle deployment, the Business Owner must request the deployment in VersionOne.

The Marketing Technology team requires written authorization from a member of the Rollins Marketing Executive Leadership team before scheduling an off-cycle deployment.

VersionOne

Rollins Marketing uses VersionOne to manage the software development process. This tool tracks requests, tasks, and defects against the Marketing sprint cycle. It is important that all contributors to Marketing software projects plan, track, and update their projects in this central tool.

Change Control and Code Management

The Marketing Team uses GitHub for version control and source code management. Source code for Orkin.com is managed in GitHub repositories and accessed by the Marketing Technology team, outside collaborators, agency partners, and Rollins contributors.

The GitHub service enables the Marketing Team to store source code, manage code revisions, and monitor the history of committed code.

Servers

Several servers hold source code at different stages of the development and deployment process:

- **DEV.ORKIN.COM** – This server is owned and administered by the Marketing Technology team. The server is used by the team to develop code during the development process. Developers view their code in this server environment and identify any necessary changes.
- **QA.ORKIN.COM** – This server is owned and administered by the Marketing Technology team. The server is used by the team to review completed code, to ensure quality and confirm the code is free of defects.
  Only a Marketing Technology team administrator can push code to this server
- **WWW.ORKIN.COM** – This server is the production environment, and contains the code that comprises the public-facing content of Orkin.com. The web-based software on this server is the Orkin.com website. Code deployed to this server is considered complete and released.
  Only a Marketing Technology team administrator can deploy the website to this server.

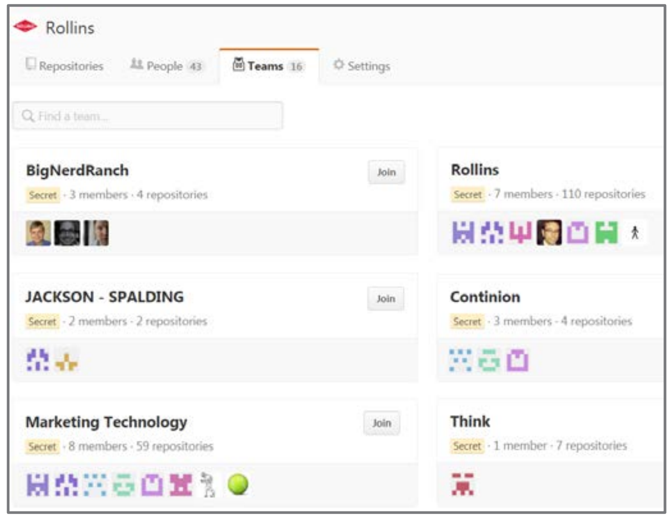
> **IMPORTANT:** Deployment to www.orkin.com is performed twice a month, in sync with the Marketing Technology sprint schedule.

GitHub Administration

The Rollins GitHub Organization hosts several teams, such as:

- Marketing Technology
- Contintion
- Jackson-Spalding

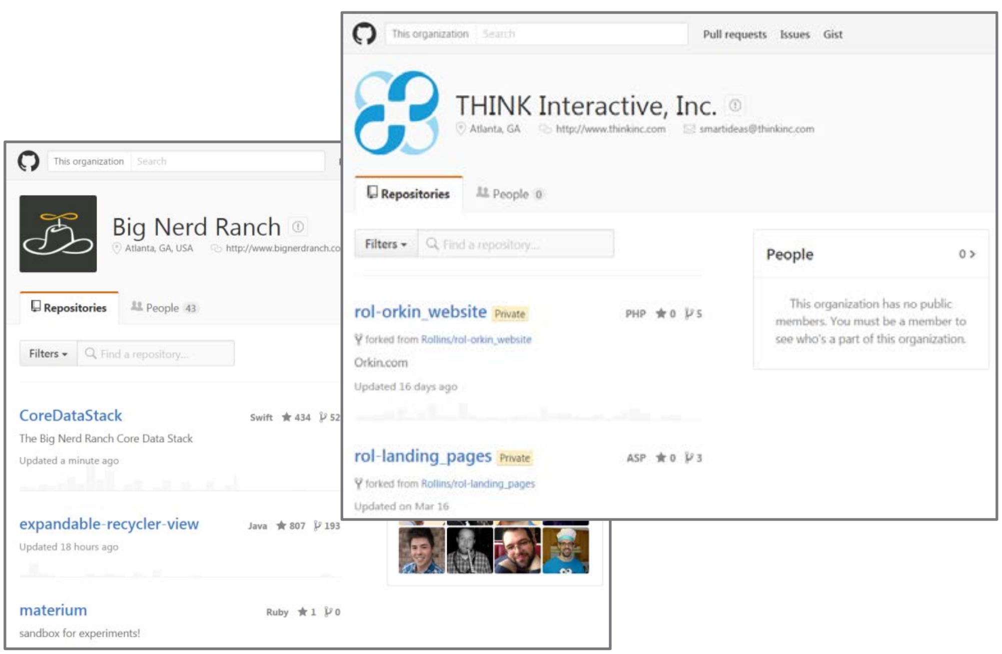
The Marketing Technology GitHub team designates several team members as administrators of Marketing Technology repositories. These administrators can merge Rollins-owned repositories and push and pull code.



Agency Partner GitHub Organizations

Additionally, many agency partners maintain their own proprietary GitHub organizations. These partner organizations host teams specific to the partner's company; for example:

- **BigNerdRanch** – This GitHub organization hosts teams which interact with the Rollins organization, and hosts teams which interact with other customers of the BigNerdRanch company.
- **WebINTENSIVE Software -** This GitHub organization hosts teams which interact with the Rollins organization, and hosts teams which interact with other customers of the WebINTENSIVE Software company.
- **THINK Interactive, Inc -** This GitHub organization hosts teams which interact with the Rollins organization, and hosts teams which interact with other customers of the THINK Interactive company.



Agency partners control and administer the repositories associated with their GitHub organizations and teams. Marketing team members cannot access or modify these partner repositories unless they are granted access permissions by the administrator of the partner team.

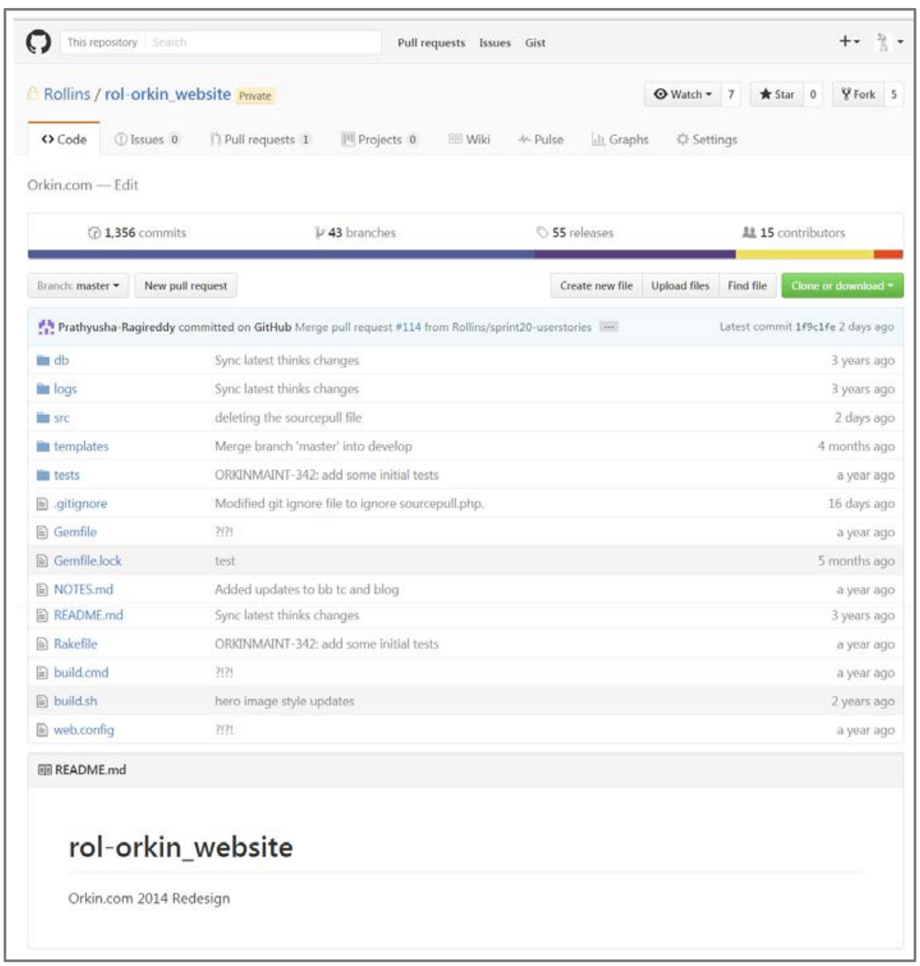
Agency partner GitHub teams outside the Rollins Organization often contribute to the Marketing Technology team repositories.

For example, the BigNerdRanch team accesses some Rollins repositories to contribute code for HomeTeam apps; the ThinkInc team accesses Rollins repositories to contribute to Orkin.com.

Marketing Technology Repositories

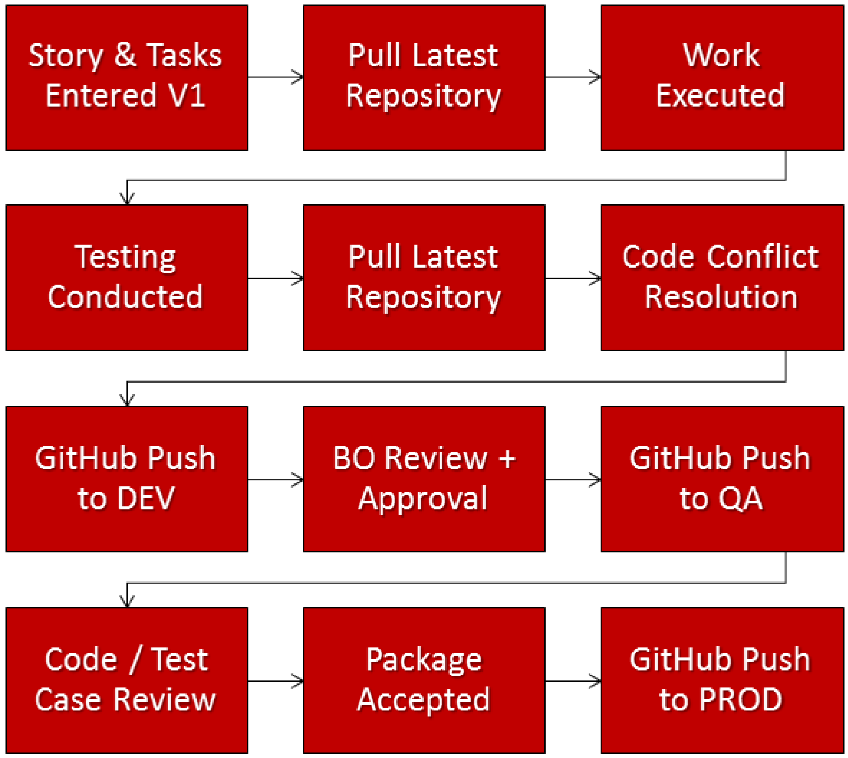
The Marketing Technology GitHub team owns many repositories, which are used to manage source code for various projects.

During the Sprint development cycle, the source code is managed in the `rol-Orkin-website-1st-server` repository, which can be pushed to the dev.orkin.com server.



Rollins Marketing Contributions

Rollins Marketing contributors use change control and development processes that align with the processes used by the Marketing Technology team.



Contributor Workflow

Contributor Workflow Process

Rollins Marketing contributors working with the Orkin.com code use the following steps to include their code in the Orkin.com website:

1. Enter a User Story in VersionOne. The content of this User Story should follow the Rollins Marketing format.
2. Add tasks to the User Story. Tasks should contain robust content explaining the planned development work. Update tasks daily.
3. Prior to beginning code development, clone the dev.orkin.com repository to ensure the planned development work begins with the most recent code changes from the Marketing Technology developers.

   If a clone already exists in your environment, use the clone to pull the latest code from the dev.orkin.com repository. This repository is `rol-Orkin-website-1st-server`.

IMPORTANT: Do not fork or download repositories. Delete any existing forks or downloads from your environment.

---

4. Develop the project in a local environment.
5. Perform quality testing on the project in a local environment.
6. Pull the latest code from the dev.orkin.com repository.

   NOTE: Before pulling the code, make a copy of the files you have changed in your local environment, so you can easily replace content in case of a file conflict.
7. Resolve any file conflicts.
8. Push the completed code to dev.orkin.com repository.
9. Notify the Business Owner, using VersionOne Conversations, that the code is ready for review on dev.orkin.com.

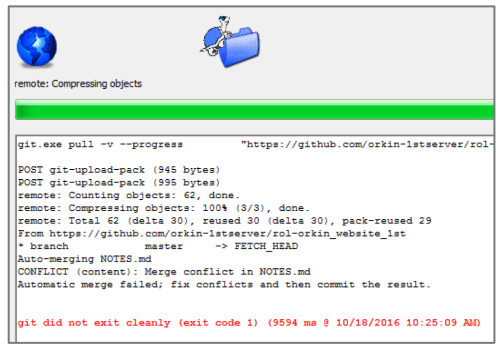
   If the Business Owner requests changes:

   - Pull the latest code from the dev.orkin.com repository.
   - Complete the requested changes in your local environment.
   - Again, pull the latest code from the dev.orkin.com repository to ensure any recent updates to the code are captured.
   - Resolve any file conflicts.
   - Push the completed code to dev.orkin.com.
   - Notify the Business Owner, using VersionOne Conversations, that the code is ready for review on dev.orkin.com.
   - If the requested changes cannot be completed and approved prior to the scheduled deployment date, as defined in the Rollins Marketing sprint schedule, contact the Marketing Technology team for assistance in reverting the changes. Your updates will be moved to the next scheduled deployment in the sprint cycle.
10. After Business Owner approval, use VersionOne Conversations to notify the Marketing Technology team that the code must go to production in the next deployment in the Rollins Marketing sprint cycle. Create a task in the VersionOne user story and attach test cases.

TIP:

Always
Pull, then Push.

To ensure you are always working on the latest code base, pull from the dev.orkin.com repository, resolve any conflicts, then push the code back to dev.orkin.com.



Marketing Technology pushes the code to qa.orkin.com at the appropriate point in the sprint cycle.

Marketing Technology performs quality testing on the code base.

After the final code base is approved, Marketing Technology pushes it to production at the appropriate point in the sprint cycle.

Syncing After Deployment

After the final code base is deployed to production, a Marketing Technology administrator ensures the production code base and the production database are synced to all servers.

This syncing process is performed after the deployment, and is scheduled to occur well after Eastern Standard business hours. This ensures any minor disruptions caused by the syncing process will occur during a low-traffic period.

New development can begin the next business day on dev.orkin.com with the latest code and database that matches production.

Tracking Progress in VersionOne

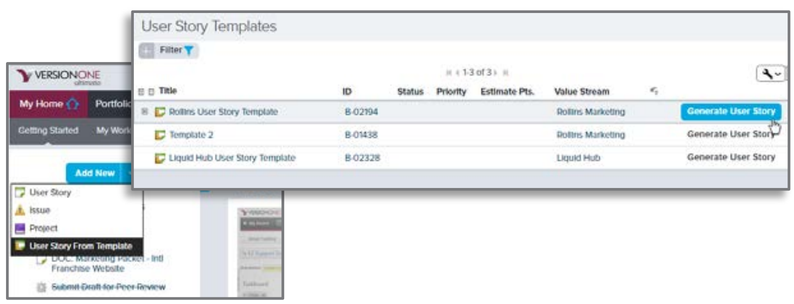
Agency partners use the Rollins Marketing VersionOne tool to track the progress of contributions. This ensures that both Marketing Technology and partners are aware of all contributions.

Adding VersionOne User Stories and Tasks

Add a user story in VersionOne to track your contribution. Follow these guidelines:

- Create your User Story from the pre-defined Rollins User Story Template.

  IMPORTANT: To avoid overwriting the template, change the title of the user story before saving.



- Use a descriptive title for the user story. The user story must be easily identifiable in lists, boards, and reports.
- Leave the Sprint field blank. This will be determined by the scrum master or product owner when the user story is prioritized.

- Use this standard Agile format to describe the user story and define acceptance criteria:

  <u>Executive Summary</u>

  <insert a brief description of the requested work>

  <u>User Story Description</u>

  As a

  I want to

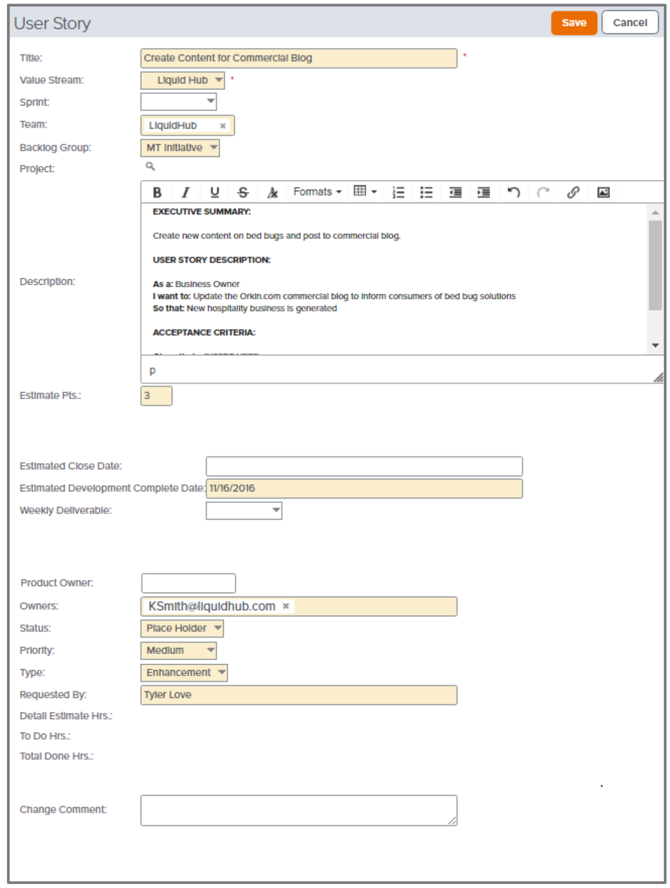
  So that

  <u>Acceptance Criteria</u>

  Given that

  When

  Then



- In the Estimated Development Complete Date field, enter the targeted deployment date, based on the current Sprint calendar.
- Leave the Product Owner field blank. This will be determined by the scrum master or product owner when the user story is prioritized.
- Enter your name in the Owners field.
- Use the default Medium Priority. The scrum master or product owner adjusts this priority as-needed. Contact the product owner to request a higher priority for your user story.
- Select Place Holder in the Status field. This will be changed by the scrum master or product owner when the user story is prioritized.
- If you are unsure of other user story selections; for example, Backlog Group, use the VersionOne Conversations tool to request input from the scrum master or product owner.

Add tasks to the user story to track the phases of your contribution. Follow these guidelines:

- Add a task for each step of the development and deployment process; for example:

  Research and Planning

  Development

  Review

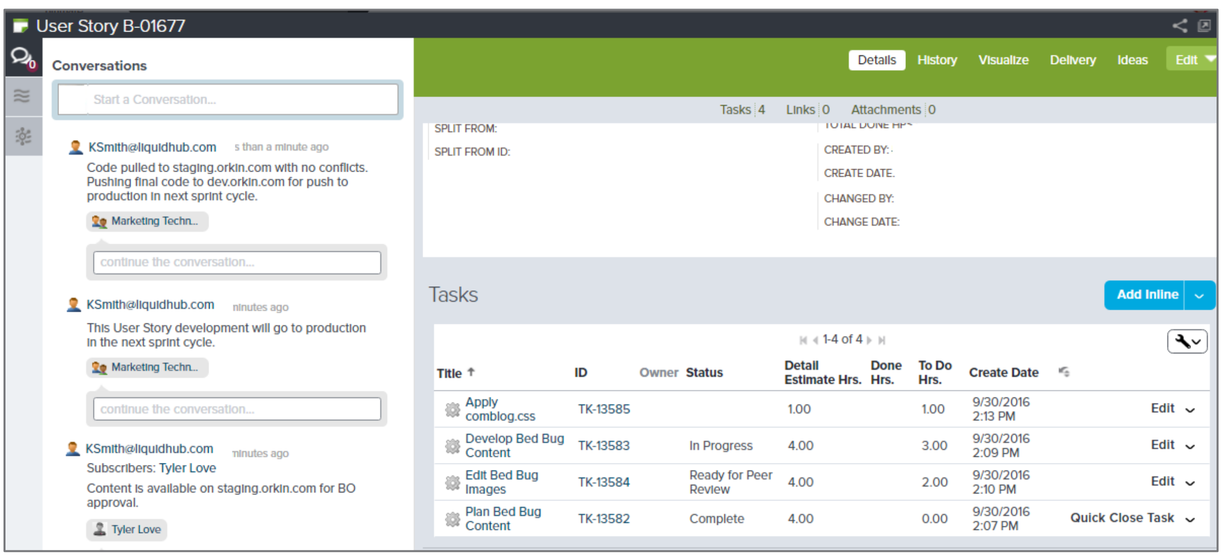
- The task's estimated hours must be four hours or less. Create ordinal tasks to subdivide large phases into short tasks; for example:

  Development Phase One

  Development Phase Two

  Development Phase Three

- Use a short, descriptive title and write a brief description of the task.
- When creating tasks, you can leave Status field blank until the associated work is in progress. Remember to update the Status field as-needed to reflect the current status of the task.
- Add the estimated hours in the Detail Estimate Hrs field.
- Add to-do hours. Typically, the estimated hours and to-do hours are the same, until the associated work is in progress. Remember to update the to-do hours daily after work on the task begins.



Use the Conversations feature, shown in the image above, to add notifications and questions, or to seek Business Owner approval. It is helpful to tag a VersionOne user in the conversation.

Business Owner Approval Process

Rollins Marketing Business Owners review contributions and approve them for deployment to production. Follow these guidelines:

- After developing and testing your contribution in a local environment, push the code to dev.orkin.com.
- Use VersionOne Conversations to notify the Business Owner that the contribution is ready for review.
- Work with the Business Owner on adjustments and revisions. If necessary, push updated code to dev.orkin.com for the Business Owner's approval.

IMPORTANT: Use VersionOne Conversations to track revisions and review cycles.

- Ensure the Business Owner approves the final contribution in VersionOne.

Mitigating Risk in GitHub Version Control

To avoid deploying outdated or incorrect code to production, it is critical that the Rollins Marketing Technology team and all Rollins contributors adhere to these best practices when committing code to GitHub repositories:

- Prior to beginning code development, clone the dev.orkin.com repository.
  If a clone already exists in your environment, use the clone to pull the latest code from the dev.orkin.com repository.
  This repository is `rol-Orkin-website-1st-server`.

IMPORTANT: Never fork or download repositories. Delete any existing forks or downloads from local environments.

- Develop and test projects in a local environment.
- To push the code, you must first pull the latest code from the dev.orkin.com repository.
  This repository is `rol-Orkin-website-1st-server`.
- Resolve any file conflicts resulting from this pull.
- Push the code to dev.orkin.com for review and approval.
- After Business Owner approval of contributions, use VersionOne to request the code be pushed to production.
- Contributors should provide test cases to the Marketing Technology team for code verification.

TIP:

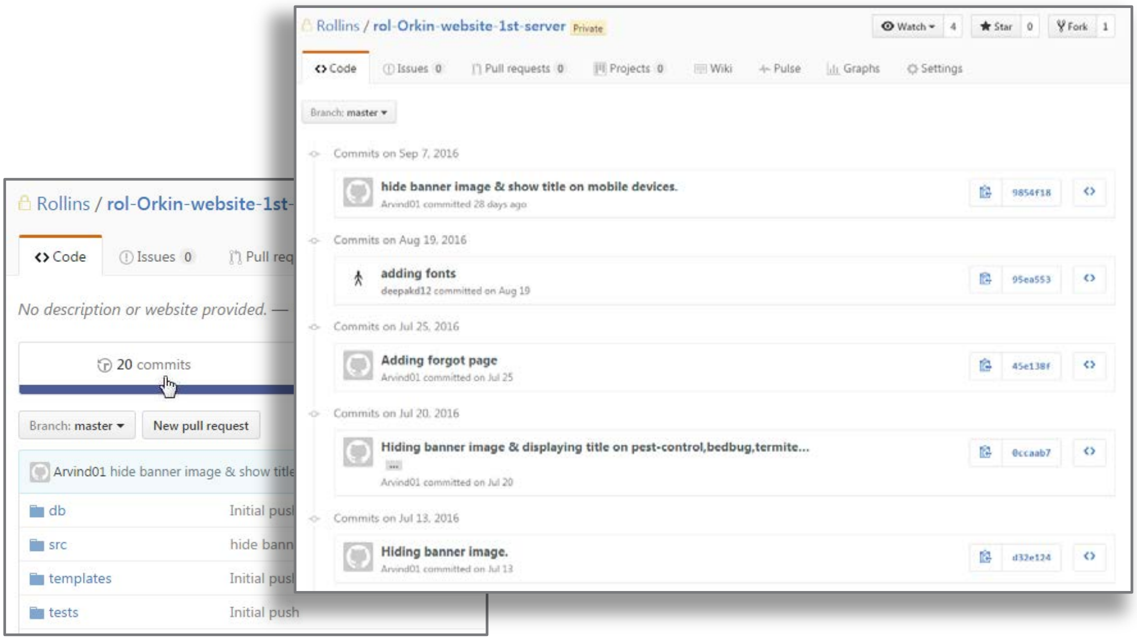
Always Pull, then Push.

To ensure you are always working on the latest code base, pull from the dev.orkin.com repository, resolve any conflicts, then push the code back to dev.orkin.com.

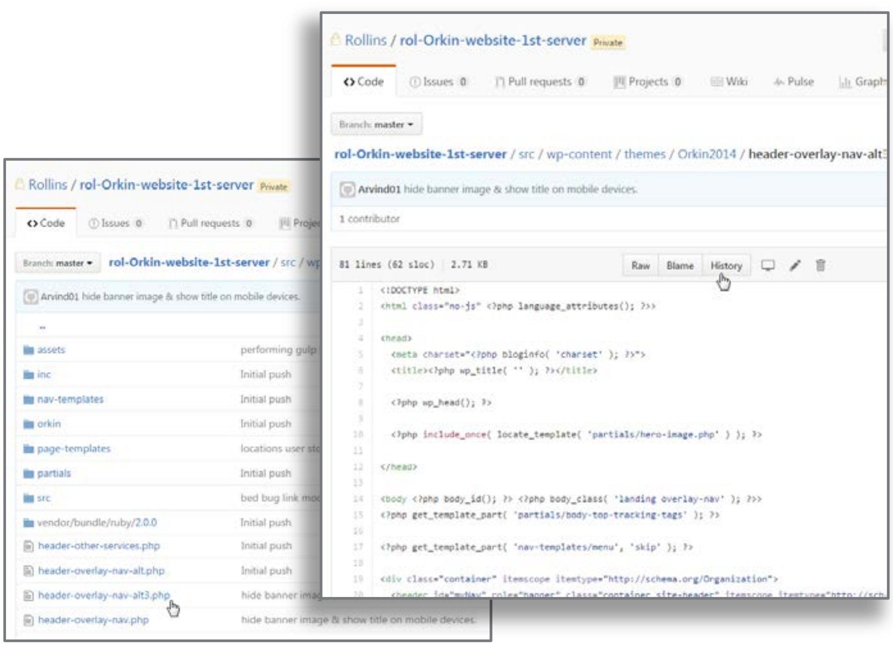
Viewing Commit History

Like most version control tools, GitHub provides two methods of viewing the history of commit actions.

To view the commit history of an entire repository, click the repository's **Commits** tab.

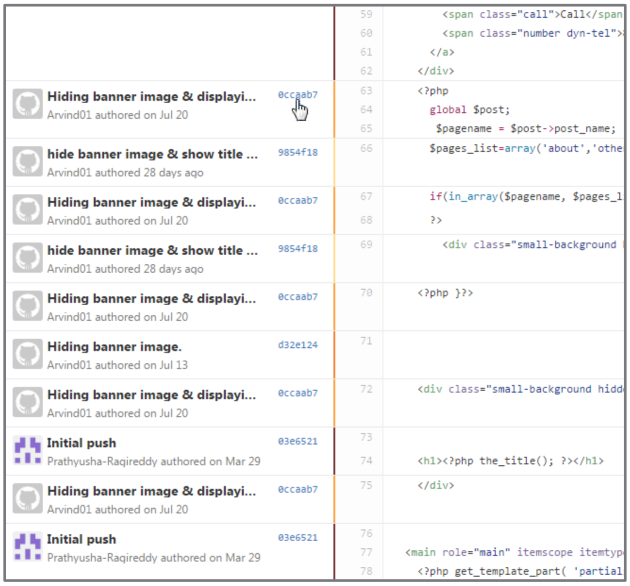


To view the commit history of a single file in a repository, click a file in the repository, then click the **History** button.



To view the change history of a file and compare code changes, click a file in the repository, then click the **Blame** button. A list of historical file changes is displayed.

The image below shows commit 0ccaab7 was made by Arvind01, and the newly-committed file has changes in lines 63, 64, and 65.



Click the file commit number to view a side-by-side comparison of the changes