

## QED42 Engineering Pvt. Ltd.

QED42 Engineering Pvt Ltd  
702, East Court, Phoenix Market City  
Viman Nagar, Pune – 411014

# Processes at QED42

## OVERVIEW

This document gives an insight into the Processes we follow on projects at QED42.

## Table of Content

- [Processes:](#)
  - [Communication and Project Management](#) - Agile Scrum and Communication Tools
  - [Governance](#) - Change Request Handling, Escalation, Leave/Unavailability, RACI, Capacity Flexibility
  - [Engineering Practices](#)
  - [QA Practice](#)
- [QED42 Advantage](#)
  - [Projects Delivered](#)
  - [QED42 Qualification](#)
  - [Hybrid team experience + References](#)
- [Point of Contact Information](#)



## Processes :

- **Project Management and Communication**
- **Governance - Change request, Escalation, Leave/unavailability, RACi , Capacity Planning**
- **Engineering Practices**

## Communication and Project Management

We have successfully delivered many large Drupal projects using Agile Scrum / Kanban approaches. A typical sprint (2 week Duration) involves following activities:

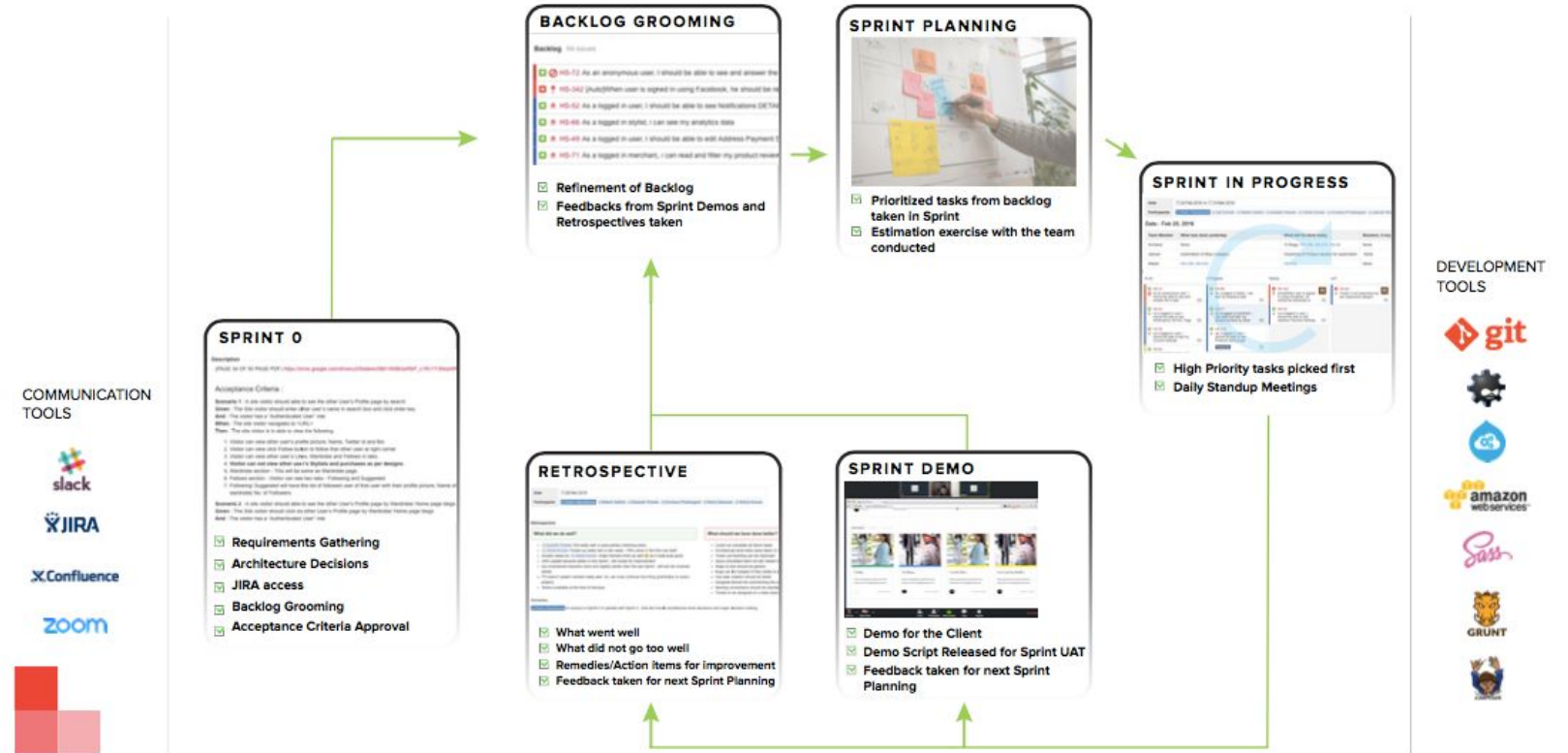
- Sprint Planning -- All hands meet to decide commitment of the team towards the sprint
- Daily Standups -- Less than 15 mins all hands catch up on the status, we maintain standup notes on confluence.
- Development and QA activities as per sprint plan
- Sprint Demo after each [Sprint with a demo script](#) ( applicable in case of new feature development ) that can later involve as CMS guide.
- Sprint Retrospective to identify what went well, what can be improved and action items. We also revisit previous Sprint's retrospective notes to see how our action items for improvement helped and track them on confluence.





[Agile Scrum Overview Illustration Below]

# AGILE SCRUM APPROACH





## Tools

For a geographically distributed scrum team, tools and their disciplined usage is vital for successful execution of project. Below is our recommendation of tools and activities from our past experience on large, long term Drupal projects delivered via hybrid teams:

- **Project Management Activities -- JIRA:**

- Backlog Grooming
- Sprint Management
- Acceptance Criteria -- Having a clear Definition of Done on JIRA tasks before it can be taken in a Sprint
- Estimations and work log
- Status Check using workflow states and JIRA boards
- Feedback and Collaboration using JIRA Comments

- **Documentation -- Confluence:** We propose using Confluence to maintain one central repository for all documentation work, throughout the course of the project. Confluence allows collaboration with ease. A basic tree page structure and constant communication on the tool, will make collaboration simple for developers and content managers. Some examples of pages that we can add to organise our Space would be :

- About the Project/Website
- High-Level Features' List
- Best Practices
- General Checklists
- RACI
- Architecture documentation for custom work
- Test Scenarios for any reusable features

- Sprint Plans
  - Demo Scripts for each Sprint
  - Scrum Notes from the daily Standup meetings
  - Deployment steps (for each release)
  - Manual configuration steps (to be taken care of, at the time of development)
  - Meeting Notes (to keep a track of discussed changes/feedback)
  - Content Manager's guide (a page that will be continuously updated with sprints/features)
  - Google Drive for sharing large files
  - Retrospective Notes (to inspect and perform better with consecutive sprints)
- **Meetings / Real time communication**
    - Zoom -- Standups, Sprint Planning, Demo calls, Sprint Retrospectives and Reviews
    - Slack -- Ad-hoc real time communication. Recently we experimented by inviting few clients on our project slack channels and we found it to be very beneficial, the biggest benefit is for all the stakeholders to have common view of conversations **e.g.** escalations can be avoided if point of escalation is looped into the conversation from the beginning allowing for proactive measures to avoid performance issues.

## Governance -

### Escalation Process

Escalation process outlines the remedies and the process for scenarios where-in Client finds an issue with a QED42's deputed team members over quality, productivity or availability.



**Process:**

- Client notifies QED42 about the nature of issue and any specifics around that
- QED42 and Client set up a call to discuss and understand the issue
- Given the severity of the issue, we mutually decide if to give feedback to the particular team member or to look for replacement.
- If we opt to replace the team member then QED42 to provide replacement within 3-4 weeks.
- If we opt to give feedback, then QED42 to monitor the team member over mutually agreed duration and submit a report to Client to either continue or a decision to go ahead with replacement.

Escalation Point of Contact -- Dipen Chaudhary [ [dipen@qed42.com](mailto:dipen@qed42.com) ]

**RACI**

We suggest to define clear RACI between Client and QED42. This will greatly help responsibility and accountability. A sample RACI can be like this :

	Tasks	Product Owner/Client PoC	PM	Development Team	QA	R-Responsible A-Accountable C-Consulted I-Informed
1	Reportings					
1.1	Escalations	I	A/R	C	C	
1.2	Regular Issue Reporting(including Risks/Contingency Plans) to Stakeholders	I	A/R	C	C	
1.3	Timely Project Status to Stakeholders	I	A/R	C	C	
3	Project Activities					



3.1	Client reports	I	R	C	C	
3.2	Sprint management	A/R	A/R	C	C	
3.3	Sprint planning	C	A/R	R	R	
3.4	Backlog grooming	R	A/R	C	C	
3.5	Technical Grooming	I	I	A/R	I	
3.6	Daily Scrum Meetings	I	A	R	R	
3.8	Sprint Demo	I	A	R	R	
3.9	Sprint Retrospective	R	A	R	R	
3.10	Weekly client meetings	R	A/R	C	C	
3.11	Features and bug tracking, Task assignment	A/R	A/R	R	R	
3.12	UAT Activity and Sign off	A/R	C	C	C	
3.13	Dev resource planning	I	R	C	-	
3.14	Deployment	I	C	A/R	C	
4	JIRA Activities					
4.1	Writing Stories In JIRA	A/R	R	C	C	
4.2	Write Acceptance Criteria	A/R	C	C	C	
4.3	Regular Updating of Stories	A/R	C	R	R	
4.4	Answering Comments in Stories	R	C	R	R	
4.5	Providing Estimations	I	I	A/R	I	





4.6	Prioritization of JIRA stories	A/R	C	C	I	
5.3	Scope Change/CR requests	C	A	C	C	

**\*Please note :** *In case a project manager isn't available/chosen for the project, the responsibilities get divided amongst the Product Owner and the Developers.*

## Capacity Flexibility

- We can make additional developers available on need basis, with a lead time of 2-3 weeks, if required. We are also flexible to replacing a backend developer with a front-end developer or vice versa with similar lead time.

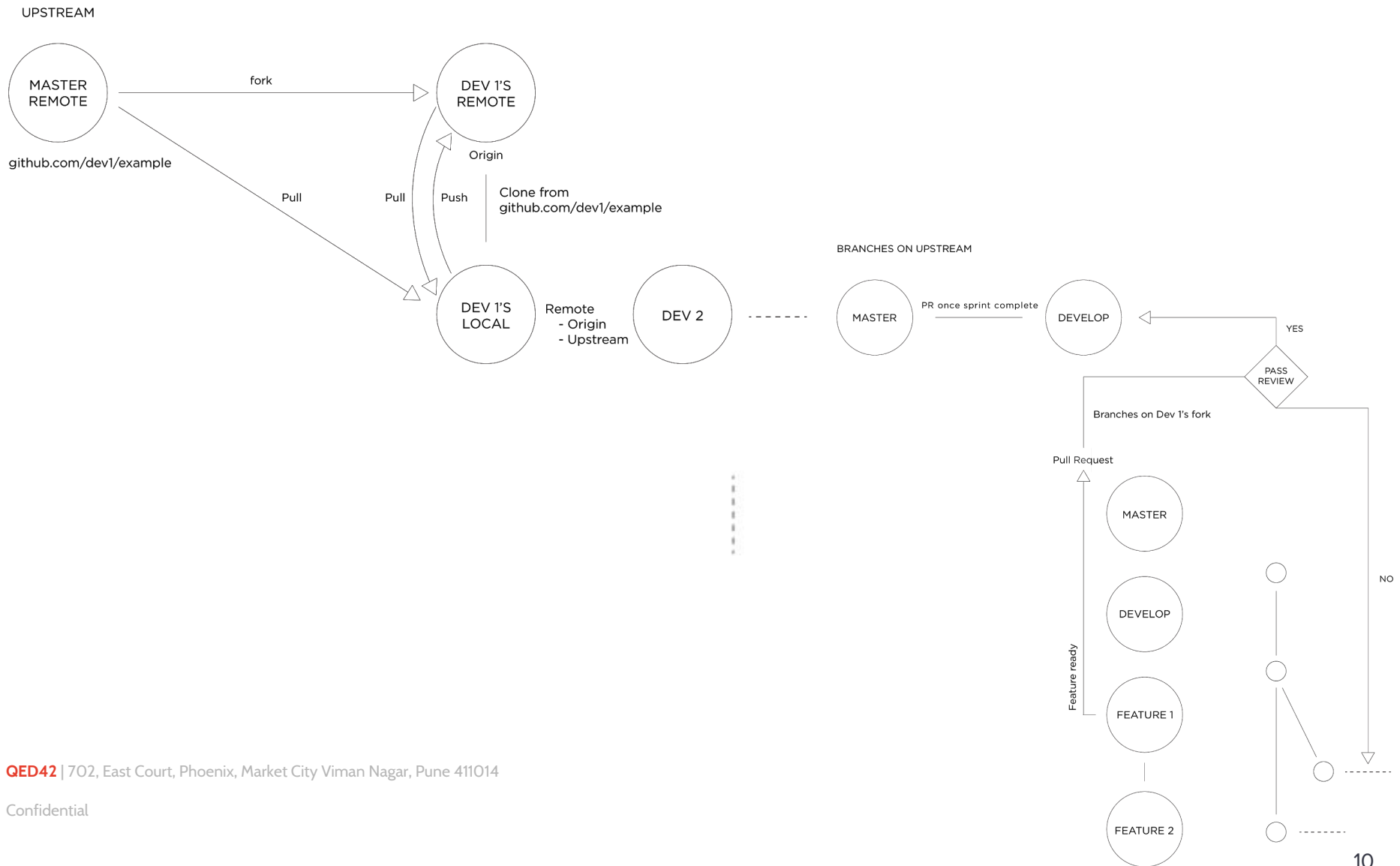
## Leave/Unavailability Management

- In a long term engagement, it would be natural for team members to not be available/ take leaves. To best address such cases of unavailability :
  - We share our holiday calendar for the year in advance and seek the same from our clients to add to our calendars and plan around the holidays
  - In case of long planned leaves from QED42 Team - we try to have a replacement well in advance to avoid any impact on the project. In any situation where a replacement may not be available immediately or we are unable to meet the hourly requirement in a week, we adjust our billing amount accordingly.
  - In case of leaves from Client team - our expectation would be to have prior information and mutually decide on who would take responsibility for decisions/blockers. We also intend to set a plan upfront, in case of unavailability.



## Engineering Practices

We adhere to Drupal Best Practices to ensure Drupal coding standards are maintained through peer code review process. We follow the fork and Pull request model to facilitate code reviews as shown below:





- **Developer Tools**

- Sublime Text or PhpStorm IDE as per developer preference
- OS X Terminal
- Git -- Github / bitbucket
- Drush & Drupal console for Drupal 8.
- Drush coder-review
- Xdebug & Xhprof
- Saas as CSS Pre processor ( Preferred )
- Gulp / Grunt for frontend tooling ( as per project setup )
- Chrome Devtools
- [Drush aliases to work with Acquia cloud](#)
- [Git hooks to enforce some standard checks and clear commit messages](#)

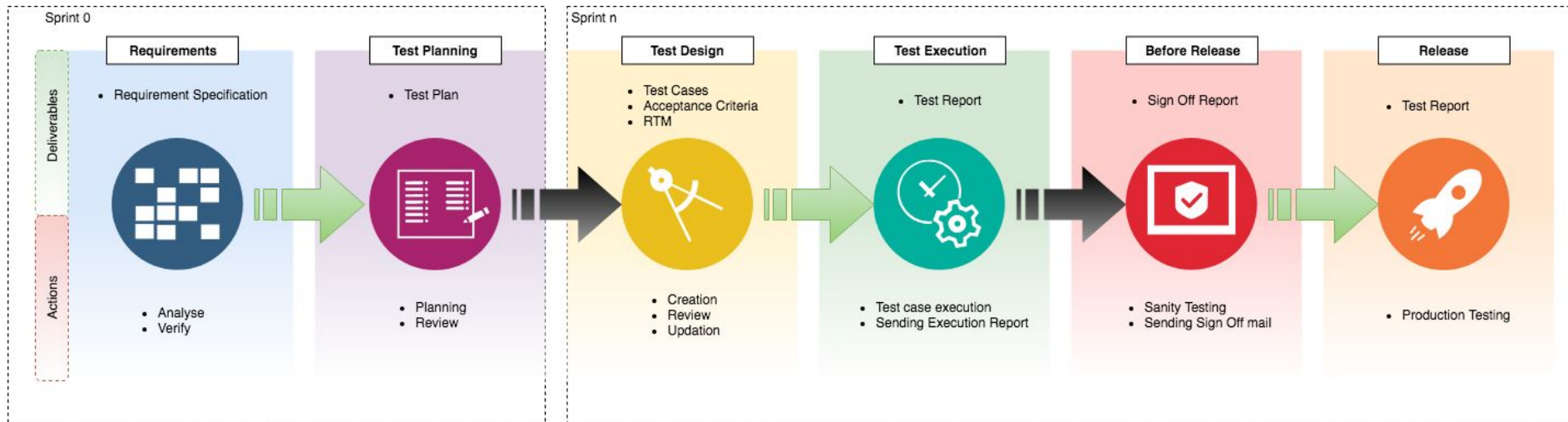
## QA Practice

We have a QA team that's well versed with Drupal CMS and nitty gritty that goes in testing Drupal applications, we provide following services as part of our QA Practice:

- Automation based on Behat and Drupal Mink Extension
- Manual Functional testing
- Manual UI Testing
- Accessibility Testing
- Performance Testing using JMeter and 3rd party services like blaze.io
- API Testing
- Automated Visual Regression Testing [ Experimental ]



## QA Process



### QED42 Qualification

- 100+ Drupal projects Delivered
- 20+ Drupal 8 Projects Delivered
- 3 Decoupled Drupal Projects using JS frontends Angular, React, Vuejs
- 43 Drupal Developers, total team size 62
- 70% Drupal team Acquia Certified Developers
- 100% of our Drupal Developers have contributed to Drupal

- 94 Projects supported on drupal.org

### **Experience in Working with a hybrid and geographically distributed Team**

With geographically distributed teams, time difference if used properly acts as the biggest asset. We utilise the gap to get clarifications done before the off shore team starts their day and for offshore team to submit their queries and deliveries for review before on shore team starts work giving us a potential 16 hr work day. For a successful distributed collaboration, we suggest the following:

- We propose to set a baseline of working methodology that best suits the Project and the People working together and use mutually agreed tools for task and project level communication.
- We need to have a few hours of overlap for daily / weekly calls for status and clarifications depending on the stage of the project.
- We decide a frequency for the catchups depending on the different phases of the project. For instance in Discovery phase we might need daily calls whereas in execution phase weekly calls may suffice with daily status updated on JIRA and shared via email.
- We time box meetings and set clear agenda to make the best out of the catchup calls.
- For optimised communication, we encourage assembling all possible feedback in one go and then have another round for final collaborated feedback
- For deployments, we prefer having a defined schedule and having different environments for dev, internal QA, UAT, Stage and Production so there is no interruption for any of the team roles. For passive deployments like daily dev / stage update Off shore team can handle during their work timings and inform the updates to on shore team of their status where as for critical deployments like milestone production deployments we recommend to use overlap hours for both teams to be available.

- We can have a daily standup/scrum meeting in the overlapping time slot (in terms of time zone), if on shore team is actively participating in execution then we recommend that both on shore and off shore team attend standup, if QED42 is wholly responsible for execution then its not mandatory to join standup but to be available to attend in case of impending blockers requiring on shore team's input.
- Conflict resolution can be handled within the team by using time after a daily scrum to resolve any blockers/conflicts
- We also recommend setting up weekly status check calls on zoom, between the Scrum master and Product Owner to keep the process smooth.

***THANK YOU!***