**Zetdc Development**

## **Scrum**

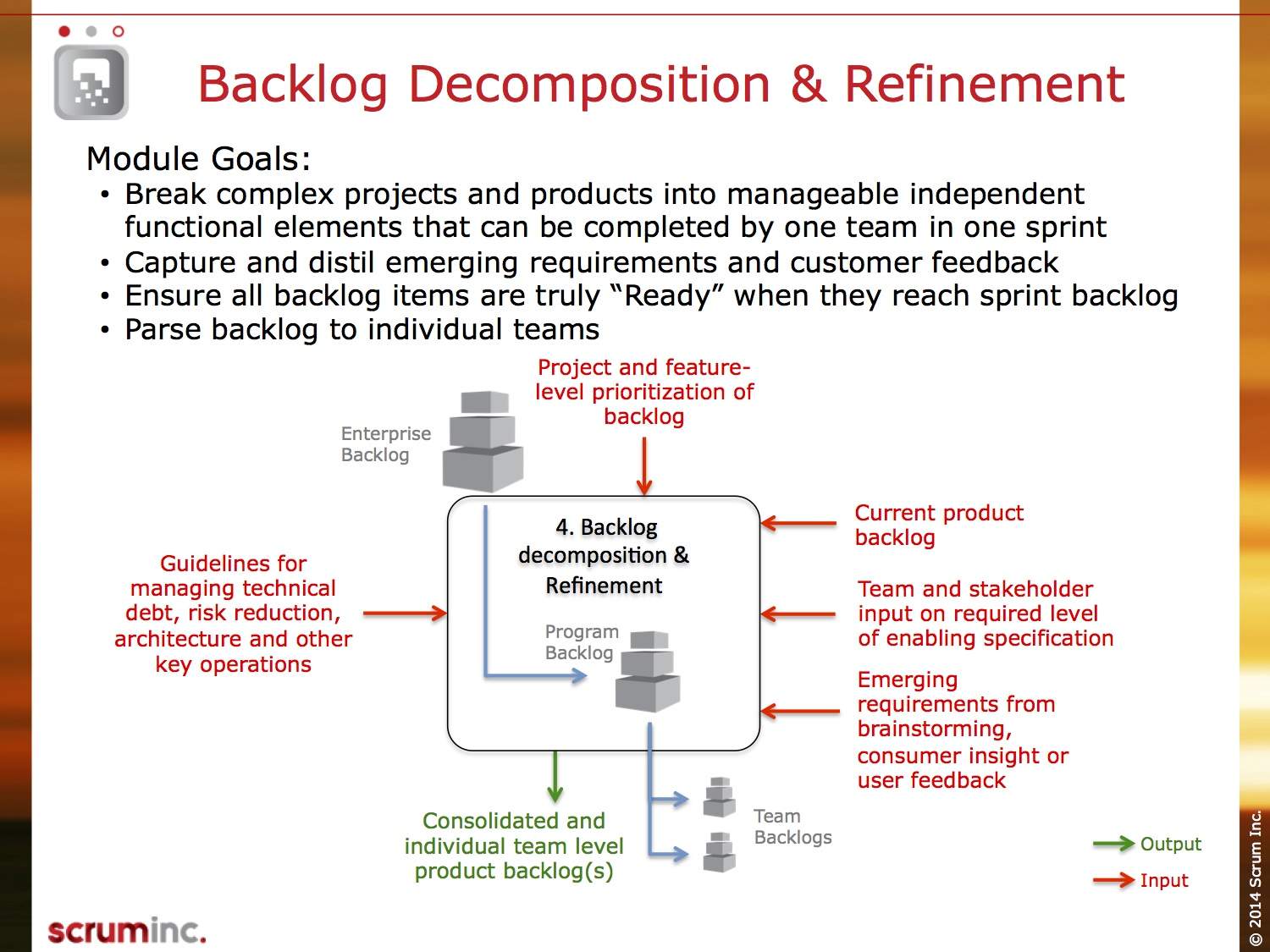
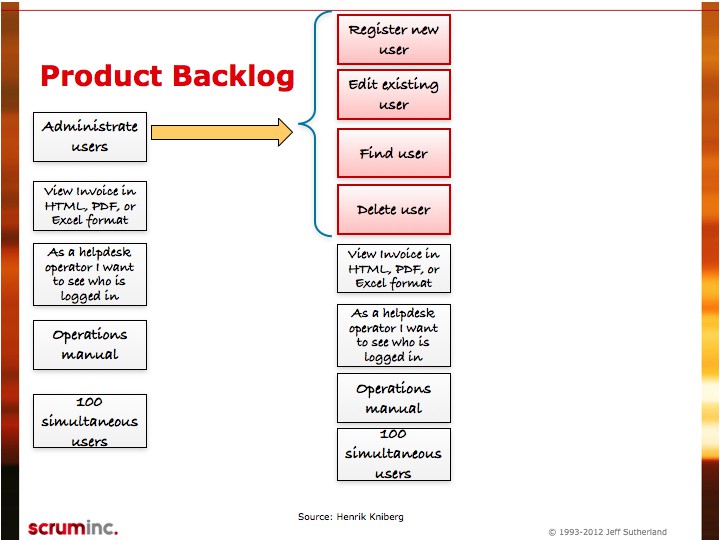
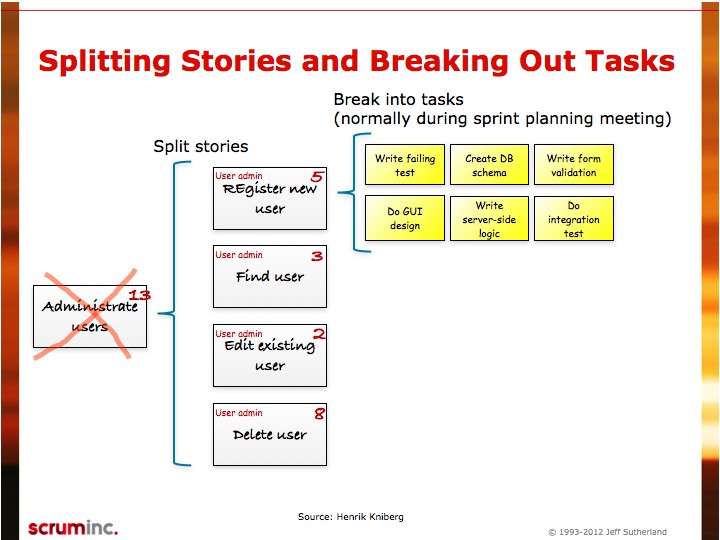
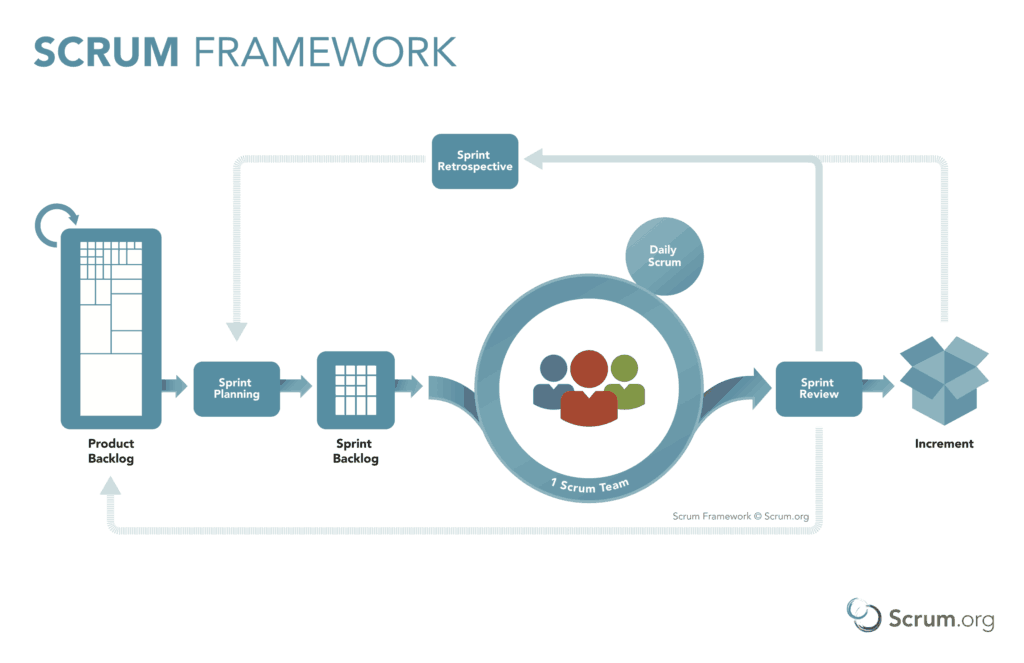
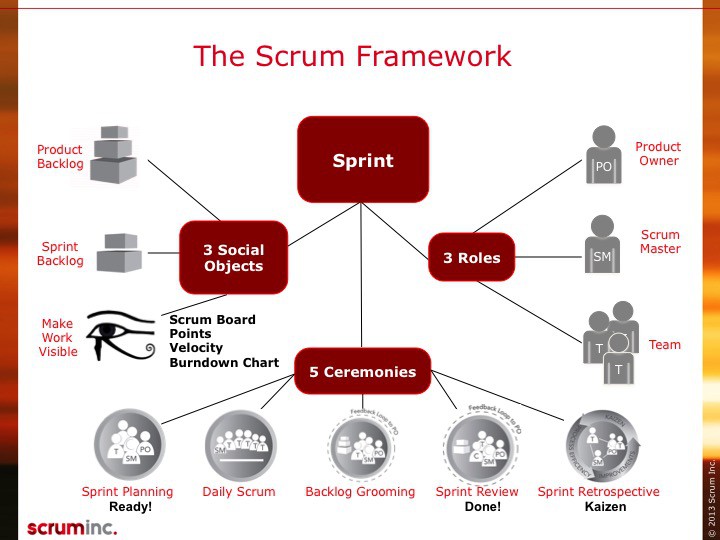
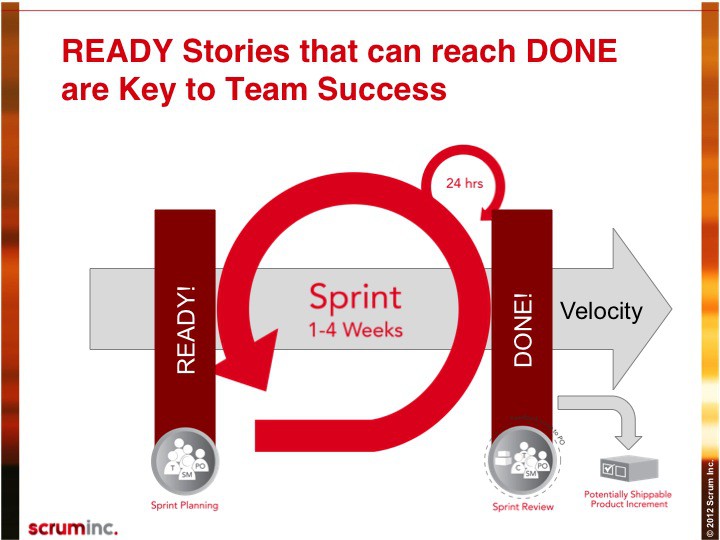
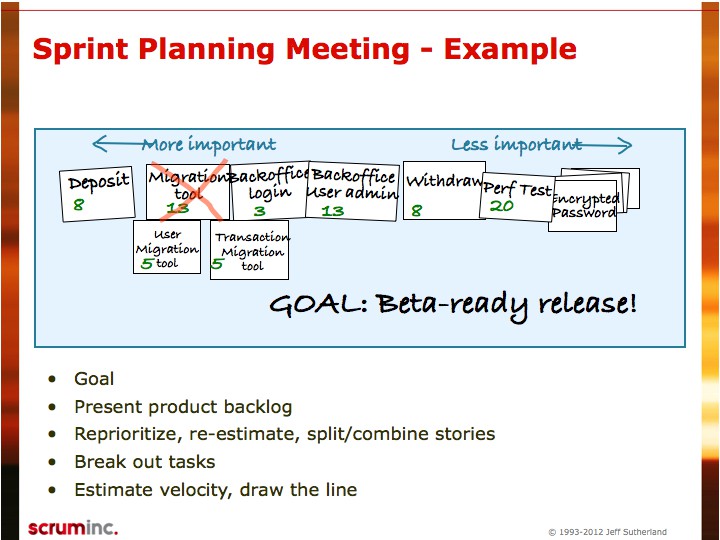
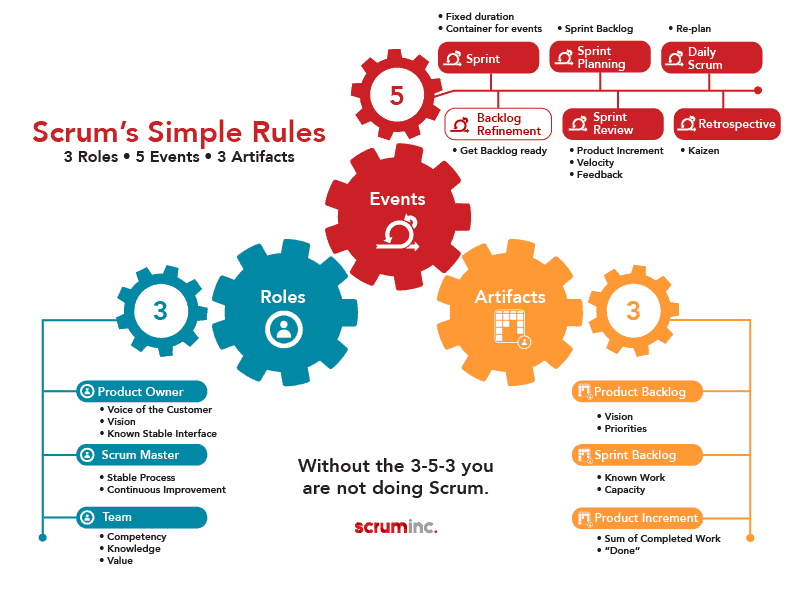
Scrum is a framework that helps teams work together. Scrum is an agile framework for managing knowledge work, with an emphasis on software development, although it has wide application in other fields and is slowly starting to be explored by traditional project teams more generally.

It is designed for teams of three to nine members, who break their work into actions that can be completed within timeboxed iterations, called sprints, no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute time-boxed stand-up meetings, called daily scrums.

* **Product Backlog** is the master list of work that needs to get done maintained by the product owner or product manager. This is a dynamic list of features, requirements, enhancements, and fixes that acts as the input for the sprint backlog. It is, essentially, the team’s “To Do” list. The product backlog is constantly revisited, re-prioritized and maintained by the Product Owner because, as we learn more or as the market changes, items may no longer be relevant or problems may get solved in other ways.
* **Sprint Backlog** is the list of items, [user stories](https://www.atlassian.com/agile/project-management/user-stories), or bug fixes, selected by the development team for implementation in the current sprint cycle. Before each sprint, in the sprint planning meeting (which we’ll discuss later in the article) the team chooses which items it will work on for the sprint from the product backlog. A sprint backlog may be flexible and can evolve during a sprint. However, the fundamental sprint goal – what the team wants to achieve from the current sprint – cannot be compromised.
* **Increment** (or Sprint Goal) is the usable end-product from a sprint. At Atlassian, we usually demonstrate the “increment” during the end-of-sprint demo, where the team shows what was completed in the sprint. You may not hear the word “increment” out in the world, as it’s often referred to as the team’s definition of “Done”, a milestone, the sprint goal, or even a full version or a [shipped epic](https://www.atlassian.com/agile/project-management/epics). It just depends on how your teams defines “Done” and how you define your sprint goals. For example, some teams choose to release something to their customers at the end of every sprint. So their definition of ‘done’ would be ‘shipped’. However, this may not be realistic of other types of teams. Say you work on a server-based product that can only ship to your customers every quarter. You may still choose to work in 2-week sprints, but your definition of ‘done’ may be finishing part of a larger version that you plan to ship together. But of course, the longer it takes to release software, the higher the risk that software will miss the mark.

Below is a list of all the key ceremonies a scrum team might partake in:

1. **Organize the backlog**: Sometimes known as backlog grooming, this event is the responsibility of the product owner. The product owner’s main jobs are to drive the product towards its product vision and have a constant pulse on the market and the customer. Therefore, he/she maintains this list using feedback from users and the development team to help prioritize and keep the list clean and ready to be worked on at any given time.
2. **Sprint planning**: The work to be performed (scope) during the current [sprint](https://confluence.atlassian.com/agile/glossary/sprint) is planned during this meeting by the entire development team. This meeting is led by the scrum master and is where the team decides on the sprint goal. Specific use stories are then added to the sprint from the product backlog.  These stories always align with the goal and are also agreed upon by the scrum team to be feasible to implement during the sprint.  
     
   At the end of the planning meeting, every scrum member needs to be clear on what can be delivered in the sprint and how the increment can be delivered.
3. **Sprint**: A sprint is the actual time period when the scrum team works together to finish an increment. Two weeks is a pretty typical length for a sprint, though some teams find a week to be easier to scope or a month to be easier to deliver a valuable increment. Dave West, from Scrum.org advises that the more complex the work and the more unknowns, the shorter the sprint should be. But it’s really up to your team, and you shouldn’t be afraid to change it if it’s not working! During this period, the scope can be re-negotiated between the product owner and the development team if necessary. This forms the crux of the empirical nature of scrum.  
     
   All the events — from planning to retrospective — happen during the sprint. Once a certain time interval for a sprint is established, it has to remain consistent throughout the development period. This helps the team learn from past experiences and apply that insight to future sprints.
4. **Daily Scrum or Stand Up**: This is a daily super-short meeting that happens at the same time (usually mornings) and place to keep it simple. Many teams try to complete the meeting in 15 minutes, but that’s just a guideline. This meeting is also called a ‘daily stand-up’ emphasizing that it needs to be a quick one. The goal of the daily scrum is for everyone on the team to be on the same page, aligned with the sprint goal, and to get a plan out for the next 24 hours.  
     
   **The stand up is the time to voice any concerns you have with meeting the sprint goal or any blockers.**   
     
   A common way to conduct a stand up is for every team member to answers three questions in the context of achieving the sprint goal:  
     
   •      What did I do yesterday?  
   •      What do I plan to do today?  
   •      Are there any obstacles?  
     
   However, we’ve seen the meeting quickly turn into people reading from their calendars from yesterday and for the next day. The theory behind the stand up is that it keep distracting chatter to a daily meeting, so the team can focus on the work for the rest of the day.  So if it turns into a daily calendar read-out, don’t be afraid to change it up and get creative.
5. **Sprint review**: At the end of the sprint, the team gets together for an informal session to view a demo of, or inspect, the increment. The development team showcases the backlog items that are now ‘Done’ to stakeholders and teammates for feedback. The product owner can decide whether or not to release the increment, although in most cases the increment is released.  
     
   This review meeting is also when the product owner reworks the product backlog based on the current sprint, which can feed into the next sprint planning session. For a one-month sprint, consider time-boxing your sprint review to a maximum of four hours.
6. **Sprint retrospective**: The [retrospective](https://www.atlassian.com/agile/scrum/retrospectives) is where the team comes together to document and discuss what worked and what didn’t work in a sprint, a project, people or relationships, tools, or even for certain ceremonies. The idea is to create a place where the team can focus on what went well and what needs to be improved for the next time, and less about



Modules

Web Application

## Web

1. Create Work Order
2. Create Job
3. Assign Team

## API

1. Add Models to Admin

API

1. Serialize Models
2. Filter Json Response Data

## Testing

### Database

Models

1. Insert Record

DB

1. Update / Modify Record
2. Delete Record

## Workflow

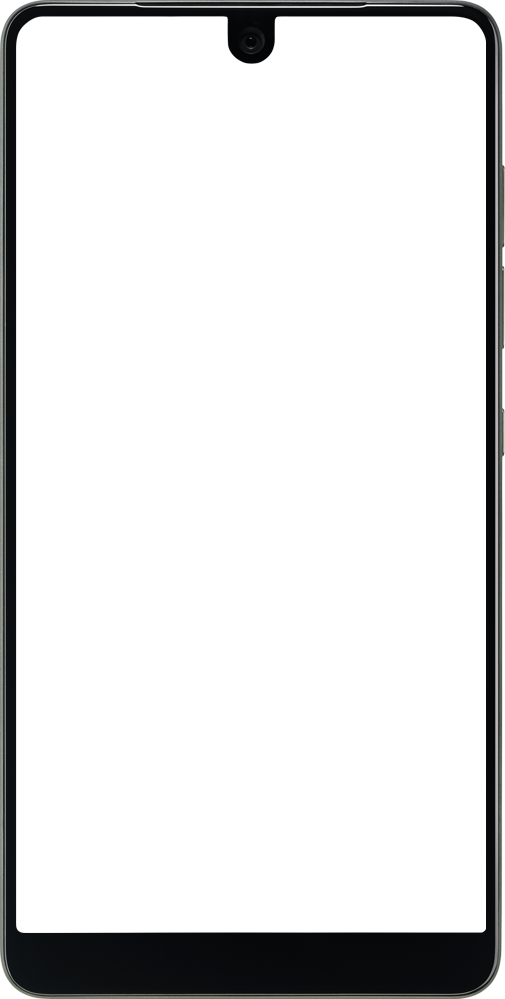
### Registration

* Role Manager

Sync

* Views Filter

## Sync Data



* Update Server Database

## Mobile

DB

1. Interface
2. mobile core
3. E84 form
4. mobile database
5. Synchronising

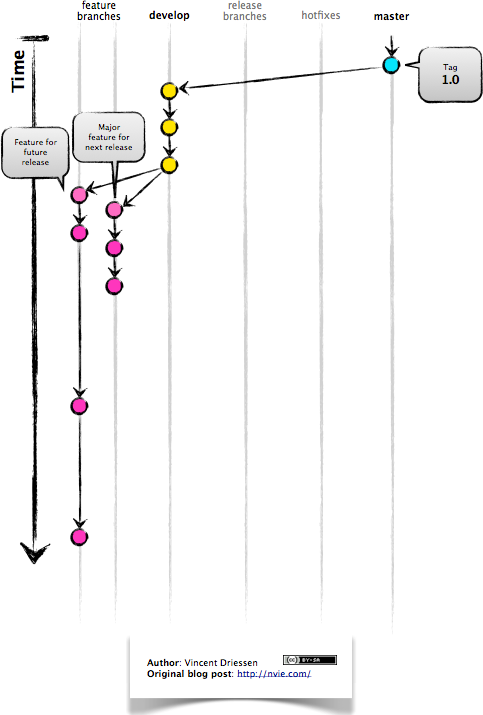
Mobile

Documentation

1. In Code Docs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product Backlog [Project Timeline] | Sprint  Backlog [New]  1 week [Goal] | Doing  [InProgress] | Review | Done  [] |
| Module 1 - Web   * Task 1 – Create Work Order * Task 2 – Create Job  1. Create Views 2. Templates 3. Get Data From Api 4. Send Data To Api 5. Update Data 6. Delete Data 7. Search Record 8. Filter data. Dropdowns 9. E84 View | M1T1  M1T2 | M1T1 – Ishe  M1T2 – KK  M1T3– Persey  M1T4– Tatenda |  |  |
| Module 2- Api   * Task 1 – Api Model Serializers * Task 2 – Admin Model Testing  1. GIS End Point [GET] 2. GIS Data interpretation [Json Serialisation , XML Serialisation] 3. Teams Serializer    1. GET    2. PUT    3. OPTIONS    4. POST 4. Get Core Schema 5. E84 Endpoint(s)    1. GET    2. PUT    3. OPTIONS    4. POST 6. Assets Endpoint(s)    1. GET    2. PUT    3. OPTIONS    4. POST | M2T1  M2T2 | M2T1 – Nigel  M2T2 – Tsitsi  M2T3 – Strive  M2T4 - Sarah |  |  |
| Module 3 - Database   * Insert Record * Update record * Delete Record * Task -Database Crud validation | M3T1  M3T2 | M3T1  M3T2 |  |  |
| Module 4 – Interface Form   * Insert Form * Update Form * Delete Form |  |  |  |  |
| Module 5 – Mobile   * Database Model (Hard Coded Data) * Sql Statements from PostgreSQL |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Git Flow - <https://datasift.github.io/gitflow/IntroducingGitFlow.html>



1. Branches

* Master - Production
* Develop - Development
* Feature/new-feature – Task and new Feature
* Hotfixes – Support Issues

1. Merging Branches
2. Testing and Pull Requests

They are branched directly from a tagged release in the **master branch**, and when finished are merged back into both **master** and **develop** to make sure that the hotfix isn’t accidentally lost when the next regular release occurs.