# “NOVEL INTEGRATED SOLUTION OF OPERATING A FLEET OF DRONES WITH MULTIPLE SYNCHRONIZED MISSIONS FOR DISASTER RESPONSES”

**ResponDrone**

# D7.2 RESPONDRONE Integration Plan

Project Deliverable Report

Deliverable Number: 7.2
Deliverable Title: RESPONDRONE Integration Plan
Author: Juan Perrela
Work Package Number: 7
Work Package Title: General Architecture, Integration, Validation and Testing

This project is funded by the European Union’s H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| RESPONDRONE Project Information | |
|---|---|
| **Project full title** | Novel Integrated Solution of Operating a Fleet of Drones with Multiple Synchronized Missions for Disaster Responses |
| **Project acronym** | RESPONDRONE |
| **Grant agreement number** | 833717 |
| **Project coordinator** | Max Friedrich, DLR |
| **Project start date and duration** | 1st May 2019, 36 months |
| **Project website** | https://respondroneproject.com/ |

| Deliverable Information | |
|---|---|
| **Work package number** | 7 |
| **Work package title** | General Architecture, Integration, Validation and Testing |
| **Deliverable number** | 7.2 |
| **Deliverable title** | RESPONDRONE Integration Plan |
| **Description** | This deliverable describes the working methodologies and the plan that will drive us to a fully integrated ResponDrone Platform.. |
| **Lead beneficiary** | Alpha |
| **Lead Author(s)** | J. Perrela |
| **Contributor(s)** | M. Borkowski, M. Friedrich, J. Lieb, H. Fontes, L. Boudet, J.P. Poli, E. Neeman, T. Gonçalves, R. Biche |
| **Revision number** | V1.0 |
| **Revision Date** | 29/07/2020 |
| **Status** | Final |
| **Dissemination level** | Public |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Document History | | | |
|---|---|---|---|
| **Revision** | **Date** | **Modification** | **Author** |
| 0.1 | 10/07/2020 | Initial Version | J. Perrela,<br>M. Borkowski<br>M. Friedrich, J. Lieb<br>H. Fontes, L. Boudet<br>J.P. Poli, E. Neeman<br>T. Gonçalves, R. Biche. |
| 0.21 | 14/07/2020 | Initial Review | J. Perrela, H. Fontes,<br>M. Friedrich, J. Lieb |
| 1.0 | 28/07/2020 | Final Review | J. Perrela,<br>R. van Oorschot,<br>M. Friedrich,<br>M. Hatziapostolidis. |

| Approvals | | | | |
|---|---|---|---|---|
| | **Name** | **Organisation** | **Date** | **Signature (initials)** |
| **Coordinator** | Max Friedrich | DLR | 29/07/2020 | MF |
| **WP Leaders** | Juan Perrela | Alpha | 29/07/2020 | JP |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Glossary of terms and abbreviations used | |
|---|---|
| **Abbreviation / Term** | **Description** |
| ADS-B | Automatic Dependent Surveillance Broadcast |
| ADS | Air Data System |
| AES | Advanced Encryption Standard |
| AGL | Above Ground Level |
| AHRS | Attitude and Heading Reference System |
| ANC | Airborne Network Control |
| AP | Autopilot |
| API | Application Programming Interface |
| ARP | Address Resolution Protocol |
| ASL | Above Sea Level |
| CEA | Commissariat à l'énergie atomique et aux énergies alternatives |
| CPU | Central Processing Unit |
| CRUD | Create Read Update Delete (in database management) |
| DEM | Digital Elevation Model |
| DHCP | Dynamic Host Configuration Protocol |
| DLR | Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center) |
| DNS | Domain Name System |
| DSS | Decision Support System |
| ETH | Ethernet |
| FLARM | Flight Alarm |
| DTM | Digital Terrain Model |
| FTP | File Transfer Protocol |
| GIS | Geographical Information System |
| GCS | Ground Control Station |
| GPS | Global Positioning System |
| GPU | Graphical Processing Unit |
| HMI | Human-Machine Interface |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Glossary of terms and abbreviations used | |
|---|---|
| **Abbreviation / Term** | **Description** |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| HW | Hardware |
| IAI | Israel Aerospace Industries |
| IAS | Indicated Air Speed |
| ICD | Interface Communication Document |
| ICMP | Internet Control Message Protocol |
| IMU | Inertial Measurement Unit |
| INESC TEC | Institute for Systems and Computer Engineering, Technology and Science |
| INHA | Inha University |
| IP | Internet Protocol |
| LTE | Long Term Evolution Network, 4G wireless communications standard |
| MVP | Minimum Viable Product |
| NCC | National Command Center |
| NFZ | No Fly Zone |
| OSCC | On-site Command Center |
| PL | Payload |
| PTU | Pan and Tilt Unit |
| PWA | Progressive Web Applications |
| PWM | Pulse-Width modulation |
| PFD | Primary Flight Display |
| QNH | QNH is an aeronautical Q code, indicating the atmospheric pressure adjusted to mean sea level |
| RAM | Random Access Memory |
| RCC | Regional Command Center |
| RF | Radio Frequency |
| RPM | Revolutions Per Minute |
| RS | Recommended Standard |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Glossary of terms and abbreviations used | |
|---|---|
| **Abbreviation / Term** | **Description** |
| RTSP | Real-Time Streaming Protocol |
| SBC | Single Board Computer |
| SDR | Software Defined Radio |
| SNMP | Simple Network Management Protocol |
| SQL | Structured Query Language |
| SSH | Secure Shell |
| SSL | Secure Sockets Layer |
| SW | Software |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TMM | Team and Mission Management |
| TX | Transmission |
| UAV | Unmanned Aerial Vehicle |
| UDP | User Datagram Protocol |
| USB | Universal Serial Bus |
| UTM | Universal Transverse Mercator |
| VOD | Video on demand |
| VPG | Video Processing at Ground |
| WFS | Web Feature Service |
| WMS | Web Map Service |
| WMTS | Web Map Tile Service |
| XMPP | Extensible Messaging and Presence Protocol |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

### *Disclaimer*



### *Copyright message*



This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## Table of Contents



This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## Table of Figures



## Table of Tables



This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

# 1. Executive Summary

This deliverable introduces the integration plan by explaining the framework to be used for the continuous integration of the different components that the ResponDrone platform is formed of. The framework will be based on Scrum framework, although some adjustments are needed to adapt it to the multi-organistation, international delocalised and more asynchronous (but still coordinated) character of H2020 projects.

After explaining the methodology, a break-out of the desired functionalities is presented, resulting in different lists of preliminarily identified developments. Also, dependencies between the different developments are studied and identified. All this information will be used in the first sprint meeting to identify priorities. The developments and their interdependencies are shown graphically in Section 7.

# 2. Introduction

The objective of this deliverable is:

- To explain the framework that will be used
- To translate the identified wishlist of features into specific developments
- To identify the dependencies between the developments and their components, in order to be able to establish priorities.

The deliverable will then generate a starting position as well as a set of management rules for the integration process. As a result, the consortium will then be able to select which tasks should start when and generate a framework that can adapt to changes.

# 3. Methodology

The integration and development process of all the pieces conforming with ResponDrone is a complex and large task that involves many parties from different companies. It implies dealing with a high number of constraints, some of which are external (for example, priorities inside companies, availability of resources, closing periods, a global pandemic etc) and therefore cannot be controlled or foreseen.

Consequently, if we opt for the development of a closed plan for the integration from the beginning of the project, although it would be feasible, it would probably fail or require a lot of adaptations during its execution. To be able to adapt to changing scenarios, we decided to adopt an agile framework based on Scrum, tailoring it to the particular charactersics of ResponDrone.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

Scrum is a useful framework for complex projects with changing scenarios or requirements. It offers a high degree of freedom, transparency across the board, and needs just minor changes to match ResponDrone needs.

It is a **framework** by which people can address **complex adaptive problems**, while productively and creatively delivering products of the **highest possible value**. Scrum is:

- Lightweight
- Simple to understand (but difficult to master)
- Effective in team collaboration on complex products. It divides a complex problem into simple, small increments.

Scrum is a well-known framework, based on 3 different team roles, 4 prescribed events, and 3 artifacts:

- **Teams** are self-organizing and cross-functional. The adoption of teams is designed to optimize flexibility, creativity, and productivity. The roles are:
    - Product Owner
    - Scrum Master
    - Development Team
- **Prescribed events** are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events. Events are:
    - Sprint Planning
    - Daily Scrum
    - Sprint Review
    - Sprint Retrospective
- **Artifacts** represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts are specifically designed to maximize the transparency of key information so that everybody has the same understanding of the artifact. Defined artifacts in Scrum are:
    - Product Backlog
    - Sprint Backlog
    - Increment

As we are performing adaptations over the Scrum framework, we can not use Scrum to refer to it, so the naming needs to be adapted as well. The roles assigned during ResponDrone are:

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 1. Assigned roles | | |
|---|---|---|
| **Role** | **Assigned to** | **Company** |
| **Product Owner** | Juan Perrela | Alpha |
| **Framework[1] master** | Richard Van Oorschot | Alpha |
| **Development team** | Technology partners | |

For more information about the Scrum roles, events, or artifacts, please consult *Annex 1: The Scrum Guide.*

### 3.1 Adaptations needed

Scrum is a framework designed for intracompany development teams, working under only one management, and dedicating all or most of their time to the project or product development in question. On the other hand, the working environment in H2020 projects as ResponDrone is much more heterogeneous. Teams working in H2020 projects usually have additional responsibilities in their companies or research centers, which must be reconciled with the responsibilities demanded by, in this case, ResponDrone. Additionally, the work and its coordination must be done, mainly, remotely. This, together with the location in different time zones, working times and shifting weekly calendars, complicates a lot the execution of the Daily Scrum.

To overcome that problem, we need to adapt the periodicity of the daily scrum to a feasible scale. The main adaptation needed is **the change of the Daily Scrum to a Weekly follow-up.** Therefore, as we are changing one of the basics of Scrum, we cannot say that we are using a Scrum framework, but something similar to it which shares a lot of its fundamental concepts.

As a consequence of these adaptations and although the similarities are obvious, we cannot claim to be working under a Scrum framework.

### 3.2 Sprint duration

The initial sprints of ResponDrone will be one-month long. Nevertheless, this will be reviewed in each Sprint planning, based on the estimated need of control and duration of the sprint backlog developments.

[1] Framework master is the equivalent to SCRUM Master.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## 4. Initial Product Backlog – User Stories

The Product Backlog is an ordered list of known features that will be needed for the development of the product. It is the single source of requirements for any changes to be made to the product. (Schwaber & Sutherland, 2017)

The backlog has been created on the basis of:

- D3.2 Preliminary Web-based Architecture & Concept
  - Table 1. Users Requirements
- D7.1 General Architecture Design
- D15.5. RESPONDRONE Functional Design
  - Table 3. Functional specification of the system

A priority number is assigned to each feature. Those with priority "1" are included into the Minimum Viable Product, and will be developed in ResponDrone, while those with priority "2" are not foreseen to be included in this stage, but their development can be evaluated if the time and budget constraints allow. We're also including them here to ensure that the basic framework is built in such a way that it's ready for expansion in that direction.

**Table 2. Product Backlog – User Stories**

| As a | Wishlist | So that… | Feature | Priority |
|---|---|---|---|---|
| **Commander / Decision Maker** | Aerial imaging. | I can improve my situational awareness. | THALES & Alpha Cameras on UAVs. | 1 |
| **First Responder** | Imaging analysis, entities geotagging and representation. | I automatize and speed up image evaluation. | Thales Video Processing. | 1 |
| **First Responder** | Backup communication cells dynamically adjustable to the coverage needs. | I can maintain communications with teams and victims on the ground. | INESC TEC Comms Relaying. | 1 |
| **Commander / Decision Maker** | GIS-based platform. Visual | Information is geo-tagged and organized over a map. | IAI Map-based UI. | 1 |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 2. Product Backlog – User Stories | | | | |
|---|---|---|---|---|
| **As a** | **Wishlist** | **So that…** | **Feature** | **Priority** |
| **Commander / Decision Maker** | Validation tools must be implemented. | The information used and orders given are reliable. | Alpha UAV order validation. | 1 |
| **Commander / Decision Maker** | Improve decision making by analysing available data. | Decisions are taken better and faster. | CEA Decision Support Service. | 1 |
| **First Responder** | 4D air traffic management: flight plan analysis and definition. | Flight safety is higher. | DLR TrafficManagement. | 1 |
| **First Responder** | Near plug and play system. Minimum number of setups before operation. | Deployment is executed quickly. | IAI: Cloud based web platform, (no installation needed). | 1 |
| **First Responder** | Materials and medical equipment delivery on request using UAVs. | Materials are delivered in time and where needed. | Alpha UAV delivery. | 1 |
| **First Responder** | Aerial localization of victims and objects. | We can respond as soon as possible. | Alpha & Thales Victims search. | 1 |
| **Commander / Decision Maker** | Multiple drones gathering information and working at the same time. | We can execute multiple missions at the same time. | Alpha & DLR Multi UAV flight control. | 1 |
| **Commander / Decision Maker / First Responder** | Effective information-sharing services for a wide range of people and teams. | Information flows are effective and there are no losses in the communication channels. | IAI Chat Service. | 1 |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 2. Product Backlog – User Stories | | | | |
|---|---|---|---|---|
| **As a** | **Wishlist** | **So that…** | **Feature** | **Priority** |
| **Commander** | Use the minimum number of operators and Ground Control Stations. | The use of human resources and needs of equipment is optimized. | Alpha Multi UAV capabilities. | 1 |
| **Commander** | Access roles with different permission levels implemented. | First Responder's hierarchy is implemented in the platform. | IAI Access Control. | 1 |
| **Commander / Decision Maker** | First Responders and other response resources to be located on the map. | We can assign resources to missions efficiently and we can control on-going missions. | IAI Web UI representation Thales FR and objects detection. | 1 |
| **Commander / Decision Maker** | 2D and 3D dynamic mapping. | Decisions are made over updated information. | IAI Mapping services. | 1 |
| **Decision Maker** | The gathered information should be stored and uploaded to the cloud . | The information is accessible remotely, not just at the emergency location. | IAI Repositories. | 1 |
| **Decision Maker** | Information platform should be cloud-based, and accessible remotely when Internet connection is available at the emergency location . | Remote decision makers can be aware of what is happening at the emergency site. | IAI Cloud based platform. | 1 |
| **Systems Integrator** | Modular services and repository-based web platform. | The system is expandable and customizable. | IAI Repositories, IAI Services. | 1 |
| **Systems Integrator** | Open standard interfaces that allow easy integration of new modules. | Integration of new items is as straight-forward as possible. | Use of RESTful APIs. | 1 |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 2. Product Backlog – User Stories | | | | |
|---|---|---|---|---|
| **As a** | **Wishlist** | **So that…** | **Feature** | **Priority** |
| **Systems Integrator** | Off-the-shelf, easily customizable, and upgradable wireless communications hardware when possible. | We can easily adapt to fast-changing technologies and usage of spectrums. | INESC TEC Customizable Radio for Communications Relay. | 1 |
| **Systems Integrator** | Open Source SW when possible. | License fees are avoided, and code is exposed to avoid information leaks. | Use of Open Source code and platforms. | 1 |
| **First Responder** | Hot-swap payloads for UAVs. | I can easily adapt to changing scenarios. | Alpha Payload exchangeability. | 2 |
| **First Responder** | Voice must be supported as peer-to-peer communication method. | I can easily, quickly, and effectively communicate with team members and stakeholders. | Voice channels integration (radio, cellphone). | 2 |
| **Commander / Decision Maker / First Responder** | Minimum risk route planner for on-ground first responders. | We avoid unnecessary risks for not being able to consider as many parameters as a machine would do. | DLR Route planner for people or First responders. | 2 |
| **Commander / Decision Maker** | Collect information from the public. | We have additional sources of information. | Public Web Application to provide and access to info. | 2 |
| **Commander / Decision Maker** | Data validation tools. | We can filter reliable information from the public. | Data validation tools for external information. | 2 |
| **Commander / Decision Maker** | HMI should allow going back and forth in time. | We can re-check past events and have predictions for future emergency development. | Timeline back and forward feature. | 2 |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 2. Product Backlog – User Stories | | | | |
|---|---|---|---|---|
| **As a** | **Wishlist** | **So that…** | **Feature** | **Priority** |
| **Commander / Decision Maker** | Platform must be able to work without internet connection, without losing information. | System is operational when no Internet access is available. Information reaches remote decision makers. | On-site instances of Web platform and critical systems with buffering and quick distribution after re-establishing communications | 2 |
| **Commander / Decision Maker** | Scalable processing power. | Processing power is not a limitation. | Distributed (cloud) processing for heavy tasks. | 2 |
| **Commander / Decision Maker** | Create reports of the event. | We can easily report to other stakeholders. | Create report menu. | 2 |
| **Systems Integrator** | Must support diverse external information sources. | The system adapts to each First Responder's sources of information. | Integration with external sources with public APIs. | 2 |

## 5. User Stories break-out

Below we increase the definition of the user stories by better describing the components that will answer each feature.

Later, we will further increase the definition of the components development by defining a first batch of inputs, outputs and subtasks required for each component. In this way we continuously drill down deeper during the process, getting more concrete at every step.

| Table 3. User Stories to components break-out | |
|---|---|
| **Feature** | **Component** |
| THALES & Alpha: Cameras on UAVs. | Payload – On board video processing and camera |
| Thales: Video Processing. | Ground Component – Video processing and broadcasting<br>Payload – On board video processing and camera |
| INESC TEC: Comms Relaying. | Ground Component - Airborne Network Control (ANC)<br><br>Payload - Communications Relay |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 3. User Stories to components break-out | |
|---|---|
| **Feature** | **Component** |
| IAI: Map-based UI. | Web Platform – Front-end (web-app, Mobile app) (IAI) |
| Alpha: UAV order validation. | Ground Component – Multidrone G-Case/GCS |
| CEA: Decision Support Service. | Web Service - Decision Support (CEA) |
| IAI: Web UI representation of risk-based decision support. | Web Platform – Front-end (web-app, Mobile app) (IAI)<br>Web Service – Decision Support (CEA) |
| DLR: Traffic Management. | Ground Component (TMM) |
| IAI: Cloud based web platform. | Web Platform – Front-end (web-app, Mobile app) (IAI)<br>Web Services<br>Repositories |
| Alpha UAV delivery. | Payload – Payload drop system |
| Alpha & Thales Victims search. | Payload – On board video processing and camera<br>Ground Component – Video processing and broadcasting |
| Alpha & DLR Multi UAV flight control. | Ground Component (TMM)<br>Ground Component – Multidrone G-Case/GCS |
| IAI Chat Service. | Web Services |
| Alpha Multi UAV capabilities. | Ground Component – Multidrone G-Case/GCS |
| IAI Access Control. | Web Platform – Front-end (web-app, Mobile app)<br>Web Services |
| IAI Web UI representation Thales FR and objects detection. | Web Platform – Front-end (web-app, Mobile app)<br>Web Services<br>Ground Component – Video processing and broadcasting |
| IAI Mapping services. | Web Services |
| IAI Repositories . | Repositories |
| IAI Services. | Web Services |
| Use of RESTful APIs. | RESTful API Implementation |
| INESC TEC Software Defined Radio for Communications Relay. | Payload - Communications Relay |
| Use of Open Source code and platforms. | All |

Moreover, the Web Services, Web Front-End and Repositories are complex components. A further break-out of them is presented in their respective sections.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## 5.1 Web Platform – Front-end (web-app, Mobile app)

The front-end applications are comprised of the following applications.

### 5.1.1 OSCC web application

The OSCC (On Site Command Center) web application allows presentation and manipulation of geo-referenced situation pictures, the creation and management of tasks, the creation and management of various drone missions, the real-time monitoring of resources and missions, sensors' control and informal collaboration (chat, voice messaging, geo referenced drawing). Its interface is shown in Figure 2:

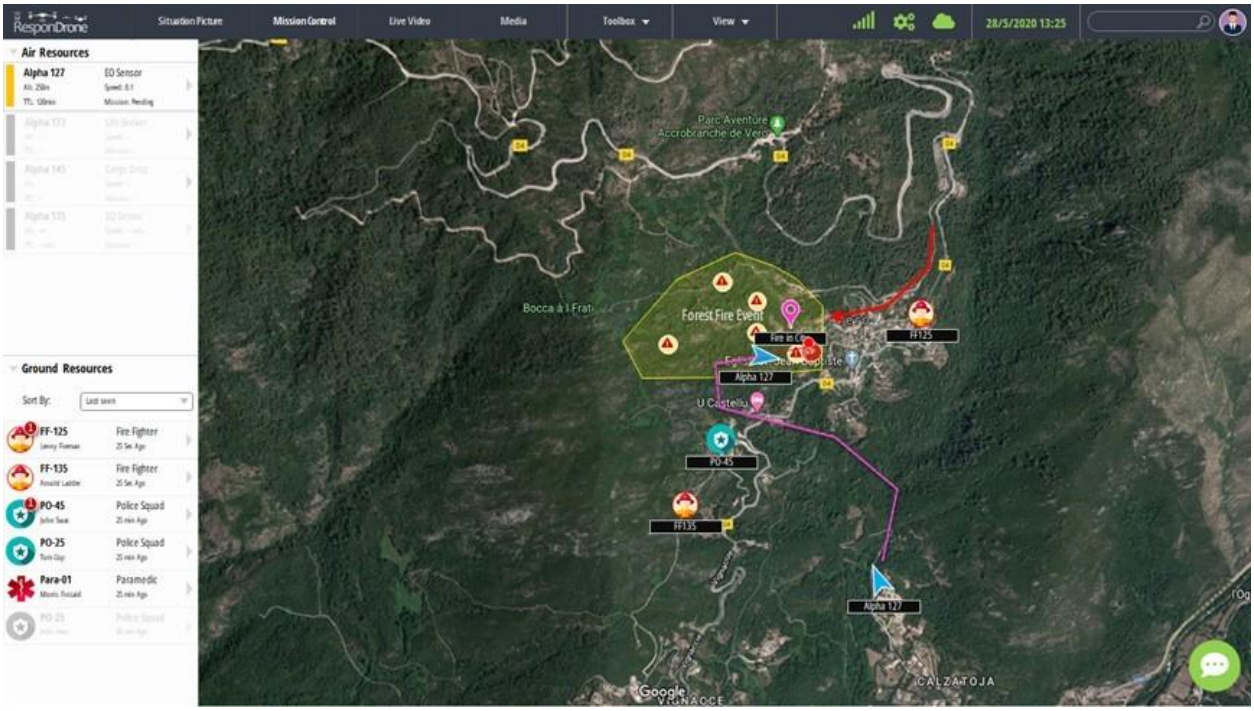


*Figure 1: OSCC web app*

OSCC web application allows the following functionality blocks, presented in Figure 2:

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/


*Figure 2: OSCC functionality blocks*

1- Call center and situation assessment block

This block allows the input, update and dissemination of situation picture based on:

- External reports received from the general public (via phone or web) or from the FRs (via their mobile apps)
- FR locations received from their mobile devices GPS location
- Events created by the OSCC operators based on situation analysis of reports, locations and other info
- Ability to ask for the risk-based decision support service

2- Resource management and dispatch provides:

- Ability to create and dispatch tasks to the various FR teams
- Ability to request different types of air missions

3- Mission monitoring and operation provides:

- Ability to monitor the entire air and ground operation (mission and task status etc.)
- Ability to control the sensors
- Ability to perform informal communication (chat, share graphic overlays etc.)

### 5.1.2 FR mobile application

The FR (First Responder) mobile application allows the FR to perform the following:

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/


*Figure 3: FR mobile App*

- Send reports to the OSCC
- Receive and manage tasks from OSCC
- View and understand ground and air situation picture
- Request air missions
- Share graphic overlays
- Chat and collaborate
- View live streams
- Control the sensor (pan, tilt, zoom, etc.)

The following table describes mobile and web applications main features and components:

**Table 4. Front-end features to components break-out**

| Feature | Components |
|---|---|
| Reports management | Mobile reporting component.<br>OSCC reporting component.<br>OSCC Reports management component.<br>Upload media component. |
| Events management | OSCC event creation component.<br>OSCC events management component.<br>Point, polyline, polygon creation tool.<br>Reports linking component |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 4. Front-end features to components break-out | |
|---|---|
| **Feature** | **Components** |
| Tasks management | OSCC task creation component.<br>OSCC Task dissemination and management component.<br>Point, polyline, polygon creation tool. |
| Ground & Air situation picture | OSCC Map client (Cesium / Leaflet)<br>Mobile Map client (Cesium / Leaflet)<br>Ground entities presentation component (Tasks, Overlays, FR locations, Events, geographic overlays)<br>Air entities (Drones, Routes, NFZs, Gimbal look at point). |
| Air missions requests management | Air missions request component.<br>Point, polyline, polygon creation tool. |
| Air mission request monitoring | Air mission request process monitoring component. |
| Payload control | Payload ownership control management component.<br>Payload software controls (pan, tilt, zoom…) component. |
| Video image presentation | Streaming video player component. |
| Video analytics presentation | Video metadata presentation component. |
| Risk-based DSS request and presentation | Ground entities presentation component (Tasks, Overlays, FR locations, Events, geographic overlays). |
| FR Location management | Mobile location broadcast component. |
| Media management | Media cloud-based file system.<br>Media upload component.<br>Media search component. |
| User account management | User management component.<br>User authentication component. |
| Technical status monitoring | Cloud status component.<br>Comm status component.<br>API status component. |
| System messages management | OSCC messages presentation module.<br>Mobile messages presentation module. |
| Chat, Textual & voice messaging | OSCC XMPP chat client component.<br>Mobile XMPP chat client component. |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 5 Web / Mobile Front-end integrations | | |
|---|---|---|
| **Source** | **Data** | **Destination** |
| OSCC | Air mission request:<br>• Resource ID.<br>• Type.<br>• Location (point, line, polygon).<br>• Additional mission related parameters. | TMM<br>Repository: Mission requests |
| TMM<br>Repository: Mission | Air mission:<br>• Route.<br>• Sensor data.<br>• Other mission-related data. | OSCC |
| OSCC, FR | Gimbal control:<br>• Gimbal commands (R/L/U/D).<br>• Zoom level.<br>• Other gimbal related data. | Ground Component – Video processing and broadcasting |
| Ground Component – Video processing and broadcasting | RTSP video stream + metadata in stream. | OSCC, FR |
| OSCC | Risk-based DSS request:<br>• Area definition (lat/long min/max).<br>• Spatial resolution.<br>• Temporal horizons.<br>• Other dss related data. | DSS |
| DSS | Risk-based DSS outputs. | OSCC |
| All | Generic user message (mechanism enabling generic system messages distribution). | OSCC, FR |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 6. Web / Mobile Front-end development preliminary list of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| WFE.cloud | Cloud infrastructure setup. | 2 weeks | |
| WFE.oscc | Development of OSCC application:<br>• Situation picture management.<br>• Mission planning management.<br>• Mission monitoring management.<br>• Simulators and mock data. | 12 weeks | |
| WFE.frapp | Development of FR app:<br>• App.<br>• Simulators and mock data. | 10 weeks | |
| WFE.mreq | Mission request integration. | 3 weeks | WFE.oscc |
| WFE.mmon | Mission monitoring integration. | 2 weeks | WFE.oscc |
| WFE.gimb | Gimbal control integration. | 2 weeks | WFE.oscc |
| WFE.vid | Video integration. | 2 weeks | WSR.video |
| WFE.dss | DSS integration. | 2 weeks | DSS.api.test |
| WFE.repo.int | Repository integration. | 2 weeks | WRP.serv |
| WFE.test | System testing. | 3 weeks | WFE.mreq<br>WFE.mmon<br>WFE.gimb<br>WFE.vid<br>WFE.dss<br>WFE.repo.int |

## 5.2 Web Services

The following table describes the main web services and components:

| Table 7. Web Services features to components break-out | |
|---|---|
| **Feature** | **Components** |
| 2D Mapping services | Geoserver geo-data and maps service. |
| 3D mapping services | Hivemapper web application and service. |
| Video streaming management | Azure media services. |
| Video archiving and indexing | Azure media services. |
| Media archiving and management | Web based file system service. |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 7. Web Services features to components break-out | |
|---|---|
| **Feature** | **Components** |
| XMPP chat services | XMPP chat server. |
| User authentication services | Authentication services. |
| Text to speech | Google text to speech. |
| Geocoding | Google geocoding services. |
| Discovery | Services discovery. |

| Table 8. Web Services integrations | | |
|---|---|---|
| **Source** | **data** | **Destination** |
| Geoserver | WMTS / WMS / WFS data.<br>DEM. | OSCC, FR, Video Processing Ground, DSS, ANC. |
| RTSP Server | RTSP stream.<br>Metadata. | Azure cloud service |
| OSCC, FR, Others | Media files. | Media cloud file system |
| FR, OSCC | XMPP chat data. | XMPP server |
| FR, OSCC | Authentication data. | Authentication service |
| OSCC, FR | Text. | Google text to speech |
| OSCC, FR | Address. | Google geocoding |
| All | Services configuration. | Discovery services |

| Table 9. Web Services preliminary list of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| WSR.gser | Geoserver setup and integration. | 1 week | |
| WSR.xmpp | XMPP setup dev and integration. | 4 weeks | |
| WSR.azure | Azure media services setup Dev and integration. | 4 weeks | |
| WSR.oauth | O-Auth setup and integration. | 2 weeks | |
| WSR.t2s | Text to speech integration. | 1 week | |
| WSR.gcod | Geocoding integration. | 1 week | |
| WSR.disc | Discovery service integration. | 1 week | |
| WSR.video | Video services development. | 4 weeks | |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 9. Web Services preliminary list of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| WSR.test | Final testing and validation. | 2 weeks | WSR.gser<br>WSR.xmpp<br>WSR.azure<br>WSR.oauth<br>WSR.t2s<br>WSR.gcod<br>WSR.disc<br>WSR.video |

### 5.3 Repositories

The repositories are externalized as a service and allow other modules to manipulate, consume and query the data in each repository. The following table describes the main repositories:

| Table 10. Repositories to components break-out | |
|---|---|
| **Repository** | **Components** |
| Mission requests | Mission requests CRUD service.<br>Mission request query service. |
| Routes | Routes CRUD service.<br>Routes query service. |
| Missions | Mission CRUD service.<br>Routes query service. |
| Geo-JSON Geo data | GeoJSONSON CRUD service.<br>GeoJSON query service. |
| Geo Video analytics data | Geo-video analytics CRUDservice.<br>Geo-video analytics query service. |
| Weather | Weather CRUD service.<br>Weather query service. |
| Air traffic data | Air traffic CRUD service.<br>Air traffic query service. |
| NFZs 4D | NFZ 4D CRUD service.<br>NFZ 4D query service. |
| Discovery | Discovery CRUD service.<br>Discovery query service. |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 10. Repositories to components break-out | |
|---|---|
| **Repository** | **Components** |
| UAV telemetry | UAV telemetry websocket room. |
| FRs telemetry | FR telemetry websocket room. |
| Sensors Telemetry | Sensors telemetry websocket room. |

The publishers and consumers of information for these repositories are specified in the following sections, along with the information they are using. It can be also consulted in Section 1. Updated Architecture..

| Table 11. Repository development preliminary list of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| WRP.serv | Development of entity repository service<br>• CRUD service.<br>• Query service.<br>• Gateway simulators.<br>• Unit testing. | 3 weeks | |
| WRP.tele | Development of telemetry service:<br>• Telemetry rooms.<br>• Telemetry archiving and logging. | 2 weeks | |
| WRP.int | Integration with TMM, Thales, CEA. | 3 weeks | WRP.serv<br>WRP.tele |
| WRP.test | Testing. | 2 weeks | WRP.int |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

### 5.4 Ground Component (TMM)

The TMM component is a ground-based software system responsible for managing both the trajectories of UAV as well as the paths of First Responders during missions. In doing so, the aim of the TMM component is to minimize the overall risk and integrate the mission management into a higher-level unmanned traffic management (UTM) system such as U-space.

The TMM component is running on a stand-alone machine on the disaster site. On a conceptual level, the TMM is responsible for the creation of UAV trajectories (lists of 4D points) based on requests from end-users.

The overall flow of information is as follows: End users submit mission requests, representing a user-based view of what the First Responder requires from the ResponDrone system (e.g., “provide aerial view of coordinates [Lat/Lon] from the north”). Based on this mission request, the TMM is responsible for creating a mission plan. This is done by contacting the respective component responsible for the given mission type. For EO/IR requests, this is the Video Processing component (see Section 5.7). For communications/connectivity requests, this is the ANC component (see Section 5.9). For payload drops, this is the payload drop component (see Section 5.5). The result of this communication is the mission plan, which is a concrete list (or single entry) of 4D positions (latitude, longitude, altitude and time), representing the required trajectory/position of the UAV. From this, the TMM creates a flight route, which represents the actual trajectory to be used by the UAV in flight. This flight route takes into account the risk model described in deliverables D2.1 and D2.2, including traffic, no-fly zones, etc.

**Table 12. TMM Inputs**

| Source | Input |
|---|---|
| OSCC | End-User Mission Requests. |
| Ground Video Processing | Mission Plan (as a response to a request by the TMM, based on an EO/IRmission request):<br>• List or envelope of acceptable points (lat/lon/alt) for fulfilling a given EO/IR-based mission. |
| ANC | Mission Plan (as a response to a request by the TMM, based on a communications mission request):<br>• List or single acceptable point (lat/lon/alt) and heading for the given UAV to fulfil a given communications-based mission request. |

This project is funded by the European Union’s H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 12. TMM Inputs | |
|---|---|
| **Source** | **Input** |
| Payload Drop | Mission Plan (as a response to a request by the TMM, based on a payload-drop mission request):<br>• Acceptable point (lat/lon/alt) to fulfil the payload drop – trivially, this can simply be the same coordinate as the mission request (with a safe altitude). |
| GCS | Stream of telemetry data representing the current state of the UAVs (see Section 5.6 for details). |
| Telemetry Repository: FR Location | Location of First Responders. |
| Repository: Geo JSON-JSON Geographic data | Geographic information associated to a mission (areas, paths…). |
| Repository: No Fly Zones 4D | No Fly Zones. |

| Table 13. TMM Outputs | |
|---|---|
| **Output** | **Destination** |
| EO/IR Mission requests:<br>- Point-to-coordinate (target GPS position + view direction).<br>- Follow path.<br>- Area scan (gimbal at nadir direction).<br>- Visual servoing (target: label + bounding box). | Video Processing Ground |
| Request for a communications mission plan<br>• "Fixed Point": 3D coordinate as the center of the area of interest.<br>• "Area Coverage": Polygon with requested coverage area.<br>• "Follow First Responders": List of first responders to follow. | ANC |
| Request for a payload drop mission plan:<br>• Requested payload drop target (lat/lon). | GCS |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 13. TMM Outputs | |
|---|---|
| **Output** | **Destination** |
| • Points of interest<br>• Flight Plans<br>• Landing plans (point and heading)<br>• Request of flight mode change<br>• No Fly zones (circular)<br>• Preflight configuration parameters. | GCS |
| Mission Plans | Repository: Missions<br>OSCC |
| UAV telemetry data (as received from GCS) | Telemetry Repository: UAV |
| Auto-generated No Fly Zones | Repository: No Fly Zones 4D |

**Table 14. TMM Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| TMM.mock | Development of component mocks for interface interaction (either self-developed simple mocks, extracted from API definition, or provided by interfacing partners). | 3 weeks | API.gcs-tmm<br>API.rep<br>API.anc-tmm<br>API.vpg-tmm |
| TMM.impl.api | Implementation of TMM-side of interfaces. | 2 weeks | API.gcs-tmm<br>API.rep<br>API.anc-tmm<br>API.vpg-tmm |
| TMM.impl.rm | Implementation of risk model into the TMM. | 8 weeks | |
| TMM.e2e.mock | End-to-end tests with mocks. | 4 weeks | TMM.mock<br>TMM.impl.rm |
| TMM.e2e.real | End-to-end tests with real interfacing components. | 3 weeks | TMM.impl.rm<br>ANC.cov<br>WRP.test<br>GCS.ext.test<br>VPG.icd.test |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

### 5.5 Payload – Payload drop system

Payload drop will consist in a release mechanism, activated with a pilot-controlled switch, which will allow to deliver goods (e.g. medicines, first-aid kits, tools, walkie-talkies) in a short time to a determined location with limited site accessibility.

The system will be designed to be flexible and usable with different droppable items that could use parachutes or other tools for softening the impact when hitting ground and also reducing the risk of injuring people or damaging other resources.

Moreover, the action of dropping an object should not affect the flight stability, taking into account effects such as the UAV weight variation and avoiding the pull release mechanism for parachutes or ensuring that, if a pull is required, the UAV and the AP can handle this force and that it does not affect flight safety or the AP control loops stability.

**Table 15. Payload Drop System Inputs**

| Source | Input |
|---|---|
| Visionair | Release command. |

**Table 16. Payload Drop System Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| PDS.prod | Payload design and manufacturing:<br>• Box or container.<br>• Release mechanism. | 5 weeks | |
| PDS.config | Visionair configuration for Payload release and test. | 2 weeks | |
| PDS.halt | Payload HALT (vibrations, impact, corrosion,etc). | 2 weeks | PDS.prod |
| PDS.fenv | Mission flight envelope definition. | 2 weeks | PDS.halt |
| PDS.test | Payload in-flight tests. | 2 weeks | PDS.config<br>PDS.test |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## 5.6 Ground Component – Multidrone G-Case/GCS

The design and adaptation process of the G-Case to multi-UAV configurations resulted in the configuration presented in Figure 4:

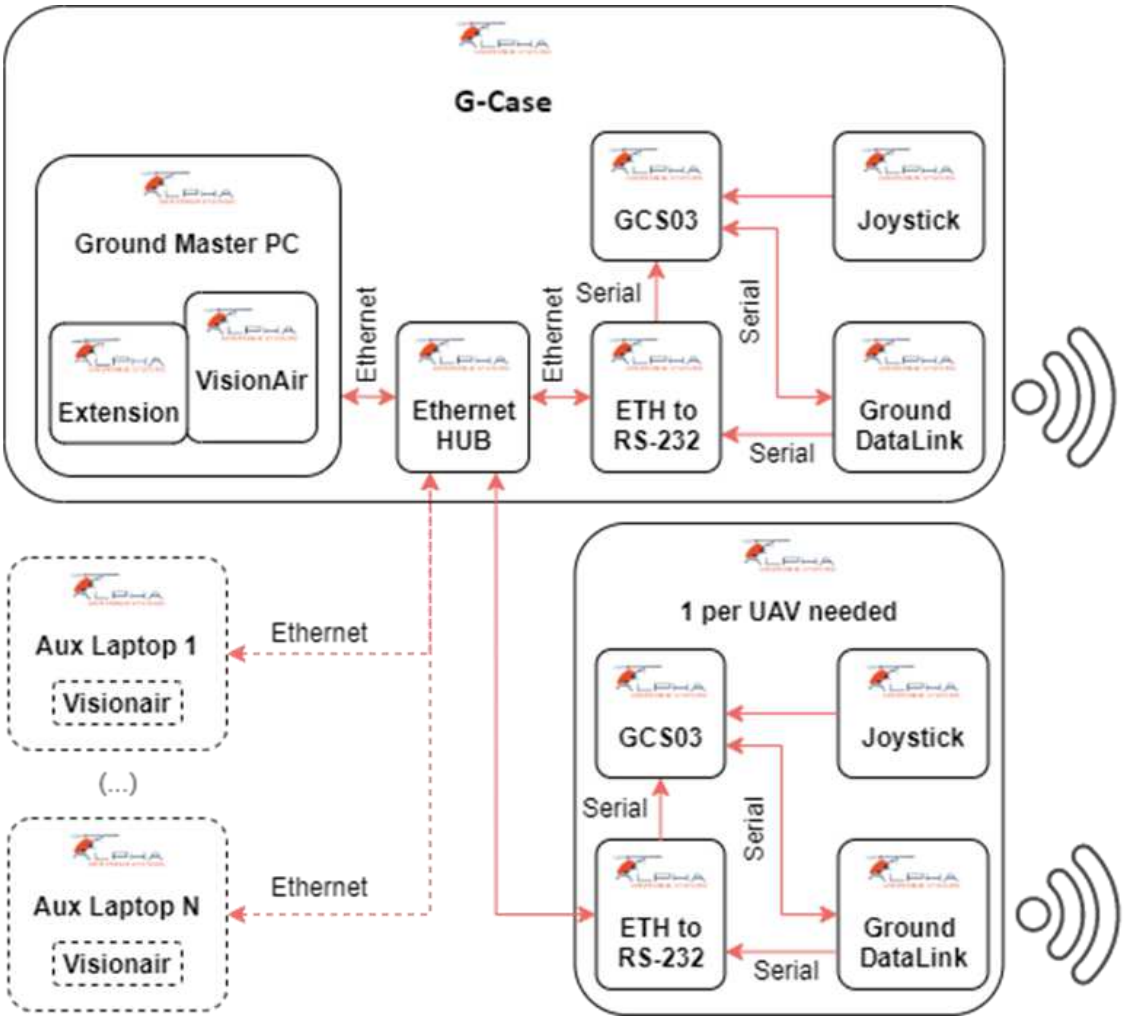


*Figure 4: G-Case Box Diagram*

- G-Case: the packed solution where all the communications and control systems are integrated into. It is composed of:
  - Ground Master PC: It will be the PC used to control all the drones in the air, and the point of contact for external communications.
    - Visionair: the flight control software from UAV Navigation.
    - Extension: a software adaptation to be developed in order to allow Visionair to communicate with external entities in a standardized way.
  - Ethernet switch.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

  - ETH to RS-232 adaptor: necessary to be able to communicate with the GCS03 in Multi-UAV environments and to avoid routing conflicts. [2]
  - GCS03: Ground Control Station 03 from UAV Navigation, which is the hardware interface required for communication between the flight control software (Visionair) and the autopilots on the drones.
  - Joystick: a control device used to take manual control of the aircraft in emergency situations, overriding commands sent from Visionair.
  - Ground Datalink: Radio link used to send commands to the drones.
- Auxiliary laptops: Computers that can be connected to the network, to be able to see and control the drones in parallel to the Ground Master PC. Auxiliary laptops (and operators) can be added in order to increase operational safety in complex scenarios and to comply with legal requirements.

The interfacing elements of the GCS will be:

- Extension: to communicate with TMM. Polling will be used between the TMM on the client side, and Visionair and the extension will act as the server side
- Ground Data link: to communicate with the drones and its payloads.

**Table 17. Multidrone G-Case Inputs**

| Source | Input |
|---|---|
| TMM | Points of interest.<br>Flight Plans.<br>Landing plans (point and heading).<br>Request of flight mode change.<br>No Fly Zones (circular).<br>Preflight configuration parameters. |
| Payloads | Data / commands (tunnelling). |

[2] GCS03 can only send broadcast messages, so if this element would not be present, the messages sent by the multiple GCS03 would be received by all the GCS03 on the network and would then be re-uploaded to the autopilots. This would generate a loop that could collapse radio data links. By adding the ETH to RS-232 adaptor and bypassing the downlink of the AP, we are able to route each AP outgoing messages just to Visionair's IP addresses.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 18. Multidrone G-Case Outputs | |
|---|---|
| **Output** | **Destination** |
| Available UAVs.<br>UAVs control status.<br>Loaded information: Flight Plans, Landing Plans, No Fly Zones.<br>Loaded points of interest.<br>Current preflight configuration. | TMM |
| Telemetry, including:<br>• Angles.<br>• Alarms.<br>• Flight mode.<br>• Range from GCS to AP.<br>• AGL/ASL.<br>• Mission Remaining time (Bingo & Mission Time).<br>• Destination WP.<br>• Camera target.<br>• Speed.<br>• Wind estimation.<br>• Timestamp.<br>• Active Flight Plan id. | TMM |

| Table 19. Multidrone G-Case Preliminary List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| GCS.ext.dev | Extension development (outsourced). | 16 weeks | API.gcs-tmm |
| GCS.ext.test | Extension testing. | 2 weeks | GCS.ext.dev<br>TMM.mock |
| GCS.net.test | Radio and network configuration testing. | 2 weeks | |
| GCS.sw | multi-UAV software and configuration validation (Visionair v8.6). | 3 weeks | GCS.net.test<br>GCS.ext.test |
| GCS.assembly | G-Case and antennas assembly. | 2 weeks | |
| GCS.flight | In-flight validation. | 4 weeks | GCS.assembly<br>GCS.sw |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

### 5.7 Ground Component – Video processing and broadcasting

The "Ground video processing and broadcasting component" will run on a dedicated computer system on-site and it will be responsible for:

- The interaction with TMM, OSCC and cloud-based components.
- The management of payload (cameras and gimbal) modes and commands.
- The broadcasting of video feeds from the payload cameras to the cloud for VOD and providing access to real-time streams.
- The ground video analytics algorithms processing.

Each UAV equipped with a camera payload will be able to send two video streams to the ground: one video stream for the visible camera and another one for the IR camera, along with the associated metadata. All these video streams will be available for live video analysis by means of a Real Time Streaming Protocol (RTSP) server accessible on this module.

The Ground Video Processing module will also offer non-critical video processing services used by FR in order to help them to efficiently analyze all the available video streams. The non-real-time video analytics algorithms will be the assessment of burned surface during wildfires and of flooded areas.

This ground component will also be responsible for the management of payload control modes and commands. This allows an end-user to control not only the gimbal orientation but also camera options like zoom, focus, etc. The payload control module will enable several types of modes, depending on the level of automatism required by the end-user:

- Full manual control
- Point-to-coordinate
- Area scan
- Visual servoing or tracking mode

With full manual control, the end-user will send commands directly to the gimbal. In this mode, two types of commands are available: rate (speed of movement) and orientation (movement). Commands for controlling camera parameters, e.g. zoom, will also be available. On Point-to-Coordinate control, the end-user will select on a map the target position as well as the direction the gimbal should point at. In order to compute the correct angles to be sent to the gimbal, this mode requires additional data like the UAV geographic position as well as the Above Ground Level (AGL) height from the autopilot telemetry. Moreover, the elevation of the target position, given by a Digital Elevation Model (DEM), will be also required. On visual servoing or track mode, the gimbal control will rely on video analytics to automatically track a target of interest (person, vehicle, etc.) during the movement of the UAV, of the target itself or even both. If

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

latency constraints allow us, this will be achieved by sending control commands to the UAV through the TMM module in order to follow the target, e.g., a person drifting in a flooded river.

Finally, this component will be responsible for pushing all the necessary information to the data repositories, e.g., cameras and gimbal telemetry, payload control mode status, video analytics metadata and geographic data like the ground footprint of both EO and IR cameras.

**Table 20. Ground video processing and broadcasting Inputs**

| Source | Input |
|---|---|
| TMM | EO/IR Mission requests:<br>- Point-to-coordinate (target GPS position + view direction).<br>- Follow path.<br>- Area scan (gimbal at nadir direction).<br>- Visual servoing (target: label + bounding box). |
| Repository: Discovery | Number of UAVs available.<br>Number of EO/IR UAVs available.<br>Location of UAVs transporting EO/IR Cameras. |
| OSCC | Payload manual control:<br>- Gimbal control commands.<br>- EO/IR cameras parameters (zoom, focus, etc.). |
| THALES Onboard video processing | Payload telemetry and video processing data:<br>- Cameras telemetry.<br>- Gimbal telemetry.<br>- Visual servoing commands.<br>- Video analytics metadata.<br>- Video streams. |
| Repository: No flight zones 4D | No flight zones, including fires. |
| GeoServer | Digital elevation model (DEM). |

**Table 21. Ground video processing and broadcasting Outputs**

| Output | Destination |
|---|---|
| Telemetry for each payload element, including:<br>• Gimbal orientation.<br>• EO/IR camera zoom, focus, etc.<br>• Status of visual servoing, e.g., target lost. | Telemetry Repository: Sensors |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 21. Ground video processing and broadcasting Outputs | |
|---|---|
| **Output** | **Destination** |
| (Geo) Video analytics. | Repository: Geo Video Analysis |
| Vector/Geo JSON Geographic data. | Repository: Geo JSON Geographic data |
| Videos for cloud storage. | Media Service: VOD |
| Videos for live streaming. | Media Service: Livestream + Meta Data |
| Mission Plan (as a response to a request by the TMM, based on an EO/IR-based mission request):<br>• List or envelope of acceptable points (lat/lon/alt) for fulfilling a given EO/IR-based mission. | TMM |
| Payload availability:<br>• Heartbeat message. | Repository: Discovery |

| Table 22. Ground video processing and broadcasting List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| VPG.ctrl.dev | Payloads control developments:<br>- State machine.<br>- Mission requests processing. | 3 weeks | VPG.icd.test |
| VPG.ctrl.test | Payloads control integration tests. | 2 weeks | VPG.ctrl.dev |
| VPG.geoctrl.dev | Geo related control and telemetry development. | 3 weeks | VPG.ctrl.test<br>WSR.gser |
| VPG.geoctrl.test | Geo related control and telemetry tests. | 2 weeks | VPG.geoctrl.dev |
| VPG.PLctrl.icd | Payload control interface development. | 3 weeks | API.vpg-wfe |
| VPG.PLtele.icd | Payload telemetry interface development. | 2 weeks | API.rep<br>API.vpg-tmm |
| VPG.mreq.icd | Mission request interface development. | 2 weeks | API.rep<br>API.vpg-tmm |
| VPG.icd.test | Interfaces integration and tests. | 2 weeks | API.vpg-wfe<br>VPG.PLtele.icd<br>VPG.mreq.icd |
| VPG.rstp.dev | RTSP server with KLV metadata development. | 4 weeks | VPG.icd.test |
| VPG.rstp.test | RTSP server with KLV metadata tests. | 1 week | VPG.rstp.dev |
| VPG.live.dev | Live video server developments. | 3 weeks | VPG.icd.test |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 22. Ground video processing and broadcasting List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| VPG.live.test | Live video server tests. | 1 week | VPG.live.dev |
| VPG.burned | Video analytics development and evaluation:<br>- Burned area detection. | 22 weeks | |

### 5.8 Payload – On board video processing and camera

The "On-board video processing and cameras" is composed of the following hardware components:

- an EO camera and an IR camera.
- an GPU-enabled embedded single board computer (SBC).
- a compact 3-axis gyro-stabilized gimbal.

Where the embedded SBC will be the central element of this module. Indeed, it will manage:

- The video grabbing from both cameras.
- The connection to the autopilot in order to collect real-time drone telemetry.
- The on-board video analytics.
- The video streaming and metadata transmission on downlink and gimbal/cameras control command on uplink.
- The connection to the gimbal in order to receive its telemetry (roll/pan/tilt angles) and send pan/tilt command to change its orientation.

The video analytics algorithms that will run on this module correspond to the most critical ones, i.e., those that should be processed on real-time. This allows tracking and following in real-time some specific moving targets, e.g., persons and vehicles, using the gimbal, but also the drone on a more advanced control mode: the visual servoing. Video analytics algorithms not requiring real-time processing will run on the Ground video processing and broadcasting module.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 23. On-board video processing and cameras component Inputs | |
|---|---|
| **Source** | **Input** |
| Alpha UAV Autopilot. | Drone telemetry |
| Thales ground video processing and broadcasting . | Payload and video processing control:<br>• Payload automatic mode.<br>• Cameras control.<br>• Gimbal manual control.<br>• Video analytics control. |

| Table 24. On-board video processing and cameras component Outputs | |
|---|---|
| **Output** | **Destination** |
| Payload telemetry and video processing data:<br>• Cameras telemetry.<br>• Gimbal telemetry.<br>• Visual servoing commands.<br>• Video analytics metadata.<br>• Video streams. | Thales ground video processing and broadcasting |

**Table 25. On-board video processing and cameras component List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| VPA.pnc.dev | Persons and vehicles detection development and evaluation. | 22 weeks | |
| VPA.fire.dev | Fire/smoke detection development and evaluation. | 22 weeks | |
| VPA.hw.dev | Payload hardware development. | 8 weeks | |
| VPA.hw.test | Payload hardware integration tests. | 2 weeks | VPA.hw.dev |
| VPA.aptele.dev | Autopilot interface development. | 4 weeks | VPA.hw.test |
| VPA.aptele.test | Autopilot interface integration tests. | 2 week | VPA.aptele.dev |
| VPA.gimb.dev | EO/IR cameras and gimbal hardware/software interface development. | 5 weeks | |
| VPA.gimb.test | EO/IR cameras and gimbal hardware/software interface integration tests. | 3 weeks | VPA.gimb.dev |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 25. On-board video processing and cameras component List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| VPA.ctrl.dev | Visual servoing control development. | 6 weeks | VPA.gimb.test<br>VPA.aptele.test |
| VPA.ctrl.test | Visual servoing control integration tests. | 3 weeks | VPA.ctrl.dev |

## 5.9 Ground Component- Airborne Network Control (ANC)

ResponDrone platform will offer a Communications Relaying Service that uses UAVs to transport INESC TEC's Communications Payload and deploy, on demand, an airborne private wireless network infrastructure for FR. This private and independent network will allow FR to communicate among themselves as well as to access the ResponDrone platform and use its services. The Communications Relaying Service will be requested on-demand by an On-field Commander to fulfil the dynamic communications needs of the scenario.

The Airborne Network Control (ANC) component, which is one part of the on-site ground systems, is responsible for processing the Communications Relaying missions, requested by the On-field Commander through the Web Frontend. Firstly, the Web Backend delivers the Mission Request to TMM, which then forwards it to the ANC component. The ANC calculates the necessary number of UAVs equipped with the Communications Payload and creates the related Mission Plan, i.e., the number of drones and the positions (3D coordinates + heading) where they should be hovering to offer communications coverage.

The ANC is triggered by TMM when a new Mission Request related to the Communications Payload is created and delivered to TMM by the Web Backend. Three types of missions will be supported by the ANC component: "Fixed Point", "Area Coverage" and "Follow First Responders", with the last two already detailed in Section 5.5 of D15.4. Depending on the type of mission, the ANC consults the Maps/GIS (to obtain terrain altitude), the First Responders' locations, the "no flight zones" (including fire locations), and the number of available UAVs and Communications Payloads. With this information, the ANC calculates the number of UAVs needed and their optimal locations (e.g., being as close as possible but not on top of First Responders, avoiding also the no fly zones) and creates the respective Mission Plan (UAV id, target 3D coordinate, heading), which is then delivered to the TMM. Depending on the type of Mission Request, new Mission Plans will be created/updated periodically by the ANC, adjusting the position of the UAVs transporting the Communications Payload to optimize the overall quality of service offered by the Radio Access Network. This is an important feature considering the dynamic needs of the scenario (e.g., First Responders' mobility to follow the fire as it evolves).

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 26. ANC Inputs | |
|---|---|
| **Source** | **Input** |
| TMM | Request for a communications Mission Plan<br>• "Fixed Point": 3D coordinate as the center of the area of interest.<br>• "Area Coverage": Polygon with requested coverage area.<br>• "Follow First Responders": List of First Responders to follow. |
| Repository: Discovery | Number of UAVs available.<br>Number of Comms Payloads available.<br>Location of UAVs transporting Comms Payload. |
| GeoServer | Digital Elevation Model (DEM). |
| Repository: No fly zones 4D | No fly zones, including fires. |
| Telemetry Repository: FR Location | Location of first responders. |
| Comms payload: | Current comms payload configuration:<br>• Radio Status (enabled or disabled).<br>• RF parameters: Channel center frequency, channel bandwidth, transmission power.<br>• Number of connected users. |

| Table 27. ANC Outputs | |
|---|---|
| **Output** | **Destination** |
| Desired number of UAVs, 3D position, heading and expected effective coverage for the requested polygon (area coverage helper). Note that this helper is called directly by IAI, to avoid creating a "fake" Mission Request. This helper's output is not restricted by the number of available UAVs with comms payload. For example, if this helper shows that 3 UAVs are needed, but in reality there are only 2 ready, this helps the Commander to request for an additional available UAV with comms payload, so afterwards it creates the actual mission request. | Repository: Geo JSON Geographic data |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 27. ANC Outputs | |
|---|---|
| **Output** | **Destination** |
| New Mission Plan, according to type of Comms Mission Request:<br>• "Fixed Point": Desired 3D position and heading of the UAV.<br>• "Area Coverage": list of UAVs, desired 3D position and heading of the UAVs.<br>• "Follow First Responders": Desired 3D position and heading of the UAVs. | TMM<br>Repository: Geo JSON Geographic data |
| Periodically updated Mission Plan of type "Follow First Responders". The updated mission plan includes the desired 3D position and heading of the UAVs, considering the recent First Responders positions. | TMM<br>Repository: Geo JSON Geographic data |
| Expected radio coverage of each UAV with Comms Payload. | Repository: Geo JSON Geographic data |
| UAV & Payload availability. | Repository: Discovery |
| Telemetry for each payload element, including:<br>○ Status.<br>○ RF parameters. | Telemetry Repository: Sensors |
| New payload configuration:<br>• Enable/disable radios.<br>• Change RF parameters. | Comms Payload |

| Table 28. ANC Preliminary List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| ANC.api | Initial implementation and validation of the RESTful API with sample data/mockup operations, based on previous specification.<br>• Initial test of interaction with interfacing modules for (1) early detection of possible integration problems and (2) reduce dependency on the implementation of other modules. | 3 weeks | API.rep<br>API.anc-tmm<br>API.anc-pcr |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

**Table 28. ANC Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| ANC.fixed | Implementation of the "Fixed Point" mission planning module. Calculate Mission Plans based on Mission Requests and resources availability, sending them to TMM. | 4 weeks | ANC.api<br>TMM.mock |
| ANC.area | Implementation of the "Area Coverage" mission planning module.<br>• Calculate Mission Plans based on Mission Requests and resources availability, sending them to TMM.<br>• Calculate area coverage based on the IAI Web Interface Backend request. | 3 weeks | ANC.fixed |
| ANC.follow | Implementation of the "Follow First Responders" mission planning module:<br>• Calculate Mission Plans based on Mission Requests and resources availability, sending them to TMM.<br>• Continuous optimization of the UAVs positioning, updating the ongoing missions whenever needed (updates Mission Plans sent to TMM). | 3 weeks | ANC.fixed |
| ANC.plmod | Implementation of the Comms Payload Setup and Monitoring module (Ground Component):<br>• Interfaces with the Comms Payload to setup the RF parameters and collect its status.<br>• Updates expected coverage of current Comms Missions with up to date information. | 4 weeks | ANC.api |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

**Table 28. ANC Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| ANC.cov | Implementation of the publishing of the expected coverage polygon and system status to the data repositories. | 2 weeks | ANC.area<br>ANC.plmod |
| ANC.test | Integration Testing with all interfacing modules<br>• Testing of the operation of all interfacing modules following a realistic sequence of interaction, validating the APIs now with all modules already developed.<br>• Implementation of the necessary adaptations to overcome unpredicted integration problems.<br>• Generate the final deployable software image. | 4 weeks | ANC.cov<br>TMM.impl.rm<br>WRP.test<br>PCR.plmod |

## 5.10 Payload- Communications Relay

The Communications Payload is a set of communications hardware components to be installed in the UAVs, capable of providing wireless network coverage for FR, allowing communications among themselves and access to the ResponDrone services. Being a private network, it works independently of the state of existing communications infrastructures. The Communications Payload is connected to the ANC using the GCS/UAV Ground DataLink, or using an optional dedicated Payload DataLink, to transport the Radio Access Network traffic between the FR and the ResponDrone system.

The hardware and software selection for the Communications Payload is in line with the objectives of the ResponDrone platform of being affordable, modular and upgradable (See D15.4 ResponDrone Concept/Mock-ups, Annex 1: Summary of the Design Thinking Workshop). In view of these objectives, the deployment of the Radio Access Networks based on the latest Wi-Fi standards is considered, focusing on using off-the-shelf, easily customizable, and upgradable wireless communications hardware, when possible. Using Wi-Fi it will: 1) be universally compatible with all mobile terminals (victims and FR) without requiring the use of custom SIM cards and carrier-unlocked phones, 2) be independent of available licensed spectrum in each area

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

(which may be none at all), and 3) avoid interfering (creating a denial-of-service) with current cellular networks operating over the respective licensed spectrum. With a strong focus on upgradability so that the system remains up-to-date and compatible (over time) with the end-user terminals accessing the network, the communications system architecture design also considers the possibility of future integration of Software Defined Radio (SDR) based LTE network interfaces. This selection also takes into account the payload weight, size and energy consumption limitations imposed by the current UAV platforms used in ResponDrone.

In order to work as expected, the Communications Payload needs to be permanently connected to the ANC on-ground component. For that purpose, the Communications Payload uses an Ethernet interface to connect the UAV Onboard DataLink.

**Table 29. Communications Relay Inputs**

| Source | Input |
|---|---|
| ANC | New payload configuration:<br>• Enable/disable radios.<br>• Change RF parameters. |

**Table 30. Communications Relay Outputs**

| Output | Destination |
|---|---|
| Current comms payload configuration:<br>• Radio Status (enabled or disabled).<br>• RF parameters: Channel center frequency, channel bandwidth, transmission power.<br>• Number of connected users. | ANC |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

**Table 31. Communications Relay Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| PCR.plcfg | Payload configuration benchmarking and final selection. | 8 weeks | |
| PCR.plmod | Implementation of the Comms Payload Setup and Monitoring module (Payload Component):<br>• Interfaces with the Airborne Ground Control module, setting up the RF parameters accordingly, and reporting its status. | 4 weeks | PCR.plcfg<br>API.anc-pcr |
| PCR.hwval | Integration and validation of correct operation with the Air-Ground Onboard DataLink hardware. | 3 weeks | PCR.plmod |
| PCR.swval | Configuration and validation of the final software image for the Comms payload:<br>• Create final software image to be used on the Comms payload, based on the best selection of OS and Radio Access Network software. | 4 weeks | PCR.hwval |

## 5.11 Web Service- Decision Support

The Decision Support System (DSS) will be provided as a web-service. The DSS web-service is based on ExpressIF®, which is a rule-based system. It analyses available data and provides the results by applying automatically the relevant rules to the situation. The DSS web-service can be used for different goals thanks to the configuration of the rules: it may assess the situation, compute scores, raise alerts or recommend actions.

The DSS web-service will be invoked through the ResponDrone Web Front End. Then, the DSS web-service will call the other services and databases, including a GIS, to fetch all information needed to the rule-based system. Among these data, current weather, weather forecasts and video analytics results will be used. It will then apply the predefined rules to the current situation. The results will be computed on the fly and may, in some cases, change from time to time. The results of the decision making service will be send back to the requester. The ResponDrone Web Front End will provide a way to display the DSS web-service GeoJSON results as a new layer or the DSS web-service JSON results as a message to the FR and OSCC. The DSS web-service results will be registered in a data repository.

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

| Table 32. Decision Support Inputs | |
|---|---|
| **Source** | **Input** |
| Repository: Geo JSON Geographic data | Situational data: starting point of the fire, location of the active fire, location of the flood, etc. |
| Geoserver | Maps, GIS layers (critical infrastructure, water, roads, buildings, Point of activities and interests, etc.) |
| Geoserver | Digital Terrain Model |
| Repository: Geo JSON Geographic data | Fire lines, floods, persons and cars detected by Thales |
| Telemetry Repository: FR Location | Location of first responders |
| Repository: Weather | Current weather |

| Table 33. Decision Support Outputs | |
|---|---|
| **Output** | **Destination** |
| DSS outputs in GeoJSON Format | Repository: Geo data and maps |
| DSS outputs in JSON Format | OSCC |

| Table 34. Decision Support Preliminary List of developments | | | |
|---|---|---|---|
| **Task Code** | **Description** | **Duration** | **Predecessors tasks** |
| DSS.wthrSim | Simulated weather forecast web-service. | 2 weeks | API.rep |
| DSS.situ | Integration of Situational Data (Events). | 4 weeks | API.rep<br>WRP.serv |
| DSS.maps | Integration of Maps . | 4 weeks | API.rep<br>WSR.gser |
| DSS.gis | Integration of GIS layers. | 2 weeks | API.rep<br>WSR.gser |
| DSS.dtm | Integration of DEM. | 3 weeks | API.rep<br>WSR.gser |
| DSS.video | Integration of video analytics data provided by Thales (fire lines, floods, cars and persons) with sample data /mockup. | 6 weeks | API.rep<br>WRP.serv |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

**Table 34. Decision Support Preliminary List of developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| DSS.frLocation | Integration of FR Location. | 1 week | API.rep<br>WRP.tele |
| DSS.wthr | Integration of Weather data. | 2 weeks | API.rep<br>DSS.wthrSim |
| DSS.api.test | Initial implementation of the DSS Web-service RESTful API with sample data/mockup rule base. | 3 weeks | API.rep<br>API.dss |
| DSS.api | Implementation of the DSS Web-service RESTful API with integrated data and mock-up rule base.<br>Interfaces with the different data sources. | 3 weeks | DSS.wthrSim<br>DSS.situ<br>DSS.maps<br>DSS.gis<br>DSS.dtm<br>DSS.video<br>DSS.frLocation<br>DSS.wthr<br>DSS.api.test |
| DSS.test | Integration and validation of correct operation with the ResponDrone Web Front End and repositories:<br>• Request from the Web Front End.<br>• Display DSS GeoJSON outputs.<br>• Display DSS JSON outputs Store DSS GeoJSON outputs (WRP.int).<br>• Store DSS JSON outputs (WRP.int). | 3 weeks | DSS.api<br>WFE.dss |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## 5.12 RESTful API Implementation

In this section we list the RESTful APIs that need to be defined, developed and implemented in order to enable the correct exchange of information between the different components.

**Table 35. API Developments**

| Task Code | Description | Duration | Predecessors tasks |
|---|---|---|---|
| API.gcs-tmm | Detailed API specification: Visionair – TMM. | 2 weeks | |
| API.rep | Detailed API specification: Repositories. | 2 weeks | |
| API.anc-tmm | Detailed API specification: ANC – TMM. | 2 weeks | |
| API.anc-pcr | Detailed API specification: ANC - Comms Relay PL. | 2 weeks | |
| API.vpg-tmm | Detailed API specification: Video Processing and broadcasting – TMM. | 2 weeks | |
| API.vpg-wfe | Detailed API specification: Video Processing and broadcasting - Web Front End for payload control. | 3 weeks | |
| API.dss-wfe | Detailed API specification: DSS - Web Front End. | 2 weeks | |
| API.anc-wfe | Detailed API specification: ANC - Web Front End. | 2 weeks | |

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717
https://respondroneproject.com/

## 6. Updated architecture

Figure 5: shows the current status of the architecture. Changes were made as a consequence of the introduction on new subcontractors and networking limitations on the new UAV's Flight Control Software.

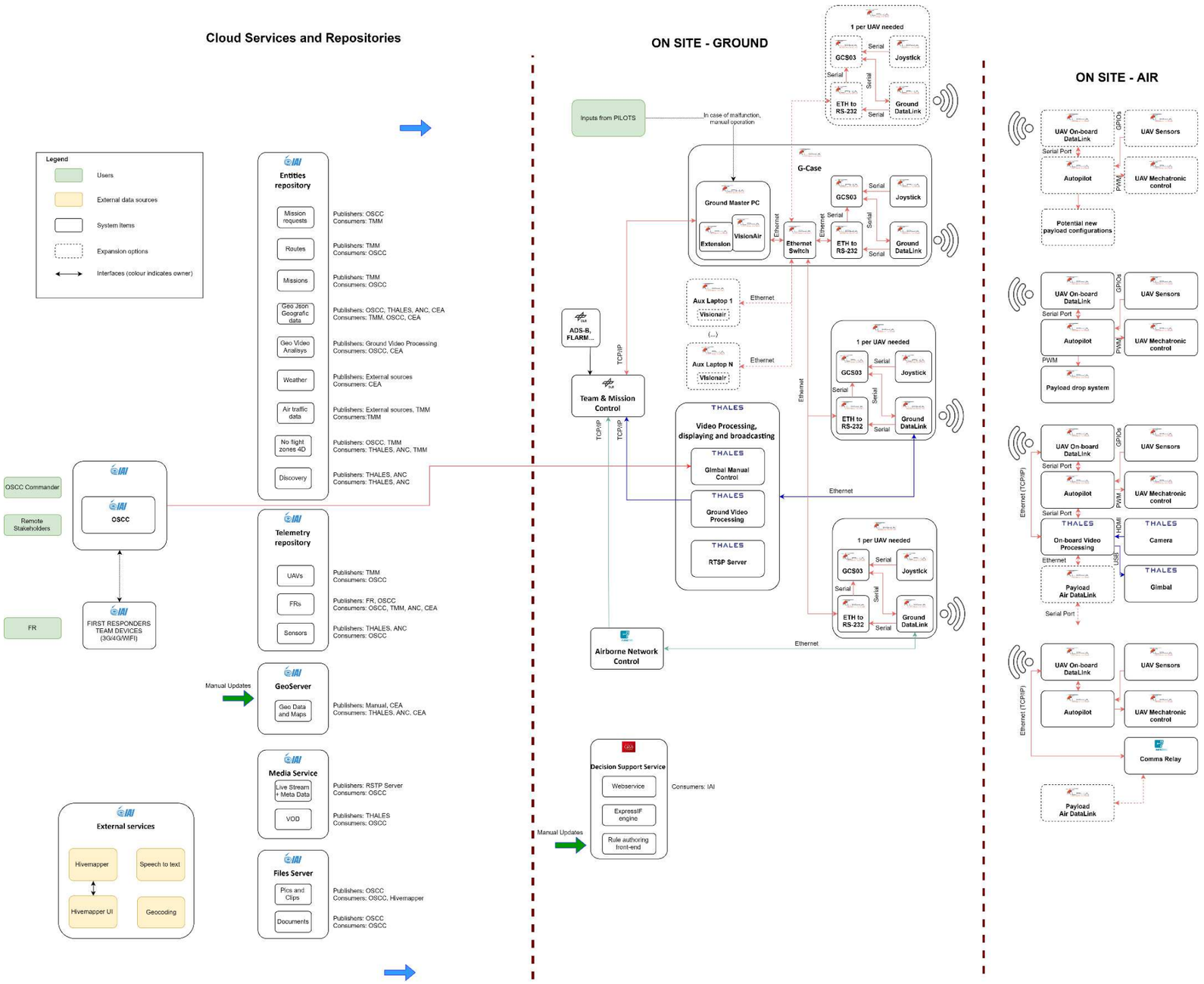


*Figure 5: Updated architecture*

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717 https://respondroneproject.com/

## 7. Dependencies diagram

The diagram below, although it has the format of a Gantt Chart, shall be used to understand the different dependencies between tasks in the project. As a consequence of using an agile framework, this diagram and its dates will be constantly subject to revision, considering each time factors such as acceptable workload, workforce availability, deadlines, national or personal holidays, introduction or changes on tasks, etc.

| Name | Du... | ID | Predecessors |
|---|---|---|---|
| API | 35 | 97 | |
| APIgcs-tmm | 10 | 98 | |
| APIrep | 10 | 102 | |
| APIanc-tmm | 10 | 99 | |
| APIanc-pcr | 10 | 100 | |
| APIvpg-wfe | 15 | 119 | |
| APIvpg-tmm | 10 | 103 | |
| APIdss-wfe | 10 | 101 | |
| APIanc-wfe | 10 | 109 | |
| WFE | 90 | 0 | |
| WFE.cloud | 10 | 6 | |
| WFE.oscc | 60 | 1 | |
| WFE.frapp | 50 | 2 | |
| WFE.mreq | 15 | 4 | 1 |
| WFE.mmon | 10 | 7 | 1 |
| WFE.gimb | 10 | 8 | 1 |
| WFE.vid | 10 | 9 | 63 |
| WFE.dss | 10 | 11 | 47 |
| WFE.repo.int | 10 | 10 | 93 |
| WFE.test | 15 | 5 | 4,7,8,9,10,11 |
| WSR | 50 | 60 | |
| WSR.gser | 5 | 66 | |
| WSR.xmpp | 20 | 61 | |
| WSR.azure | 20 | 62 | |
| WSR.oauth | 10 | 64 | |
| WSR.t2s | 5 | 67 | |
| WSR.gcod | 5 | 68 | |
| WSR.disc | 5 | 69 | |
| WSR.video | 20 | 63 | |
| WSR.test | 10 | 65 | 61,62,63,64,66,67,68,69 |
| WRP | 50 | 92 | |
| WRP.serv | 15 | 93 | 102 |
| WRP.tele | 10 | 95 | |
| WRP.int | 15 | 94 | 93,95 |
| WRP.test | 10 | 96 | 94 |
| TMM | 90 | 20 | |
| TMM.mock | 15 | 23 | 98,99,102,103 |
| TMM.impl.api | 10 | 25 | 98-FS=P15D,99-FS=P15D,102-FS=P15D,103-FS=P15D |
| TMM.impl.rm | 40 | 21 | |
| TMM.e2e.mock | 20 | 22 | 21,23 |
| TMM.e2e.real | 15 | 24 | 17,21,80,91,96 |
| PDS | 45 | 54 | |
| PDS.prod | 25 | 55 | |
| PDS.config | 10 | 56 | |
| PDS.halt | 10 | 57 | 55 |
| PDS.fenv | 10 | 58 | 56 |
| PDS.test | 10 | 59 | 56,57,58 |
| GCS | 125 | 13 | |
| GCS.ext.dev | 80 | 14 | 98 |
| GCS.ext.test | 10 | 17 | 14 |
| GCS.net.test | 10 | 19 | |
| GCS.sw | 15 | 16 | 17,19 |
| GCS.assembly | 10 | 18 | 14-FF |
| GCS.flight | 20 | 15 | 16,18 |
| VPG | 110 | 70 | |
| VPG.ctrl.dev | 15 | 72 | 80 |
| VPG.ctrl.test | 10 | 76 | 72 |
| VPG.geoctrl.dev | 15 | 73 | 66,76 |
| VPG.geoctrl.test | 10 | 77 | 73 |
| VPG.PLtele.icd | 10 | 78 | 102,103 |
| VPG.mreq.icd | 10 | 79 | 102,103 |
| VPG.icd.test | 10 | 80 | 78,79,119 |
| VPG.live.dev | 15 | 75 | 80 |
| VPG.rstp.dev | 20 | 71 | 80 |
| VPG.live.test | 5 | 82 | 75 |
| VPG.rstp.test | 5 | 81 | 71 |
| VPG.burned | 110 | 83 | |
| VPA | 125 | 26 | |
| VPA.pnc.dev | 110 | 35 | |
| VPA.fire.dev | 110 | 36 | |
| VPA.hw.dev | 40 | 27 | |
| VPA.hw.test | 10 | 33 | 27 |
| VPA.aptele.dev | 20 | 30 | 33 |
| VPA.aptele.test | 10 | 34 | 30 |
| VPA.gimb.dev | 25 | 29 | |
| VPA.gimb.test | 15 | 31 | 29 |
| VPA.ctrl.dev | 30 | 28 | 31,34 |
| VPA.ctrl.test | 15 | 32 | 28 |
| ANC | 80 | 84 | |
| ANC.api | 15 | 88 | 99,100,102 |
| ANC.fixed | 20 | 85 | 23,88 |
| ANC.area | 15 | 89 | 85 |
| ANC.follow | 15 | 90 | 85 |
| ANC.plmod | 20 | 86 | 88 |
| ANC.cov | 10 | 91 | 23,86,89 |
| ANC.test | 20 | 87 | 21,39,91,96 |
| PCR | 95 | 37 | |
| PCR.plcfg | 40 | 38 | |
| PCR.plmod | 20 | 39 | 38,100 |
| PCR.hwval | 15 | 41 | 39 |
| PCR.swval | 20 | 40 | 41 |
| DSS | 75 | 42 | 102 |
| DSS.wthrSim | 10 | 50 | |
| DSS.situ | 20 | 44 | 93 |
| DSS.maps | 20 | 45 | 66,102 |
| DSS.gis | 10 | 51 | 66 |
| DSS.dtm | 15 | 46 | 66 |
| DSS.video | 30 | 43 | 93,102 |
| DSS.frLocation | 5 | 53 | 95 |
| DSS.wthr | 10 | 52 | 50 |
| DSS.api.test | 15 | 47 | 101 |
| DSS.api | 15 | 48 | 43,44,45,46,47,50,51,53 |
| DSS.test | 15 | 49 | 11,48 |

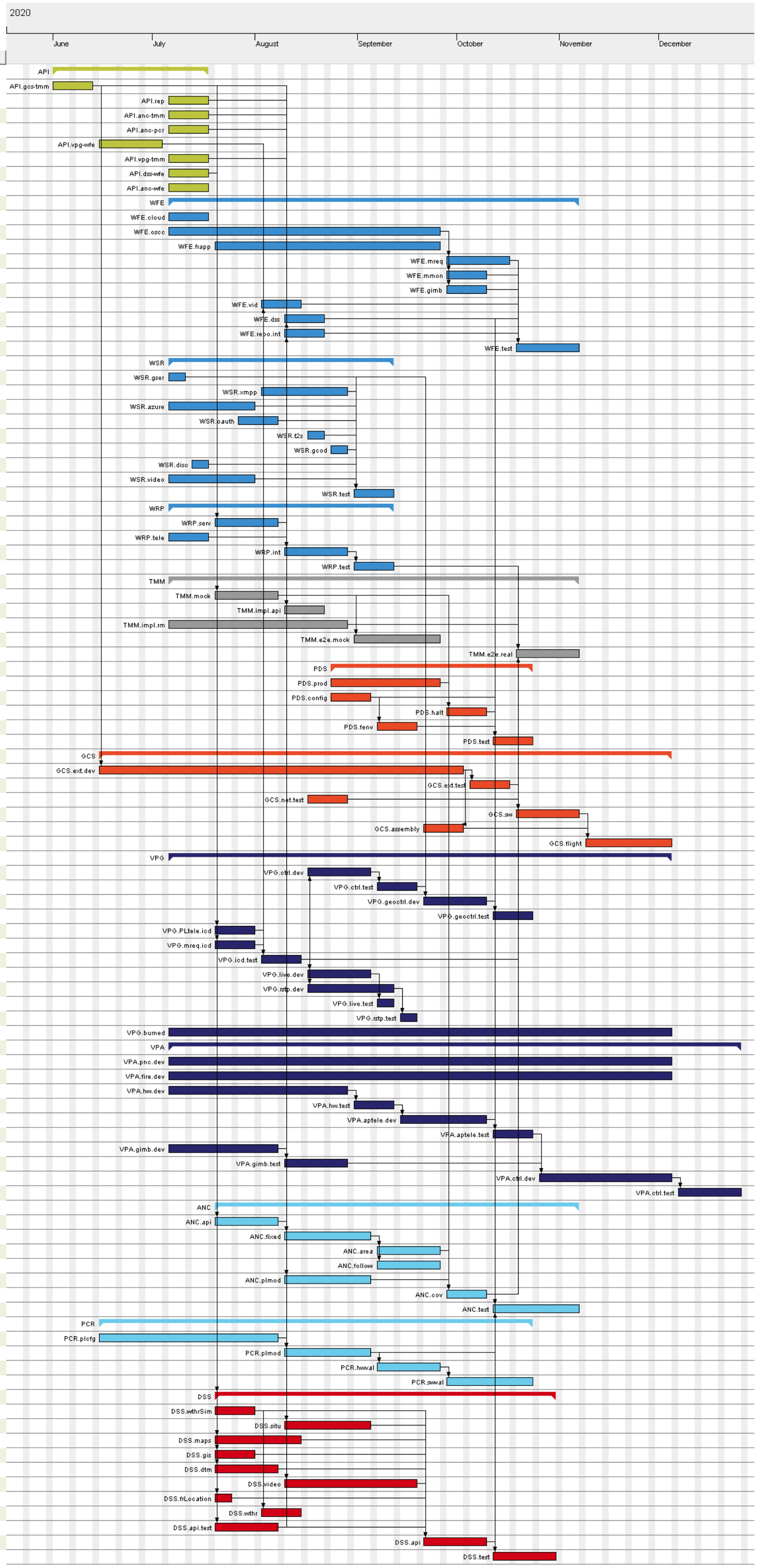


*Figure 6: Dependencies diagram*

# 8. Annex 1: The Scrum Guide

This project is funded by the European Union's H2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833717

# The Scrum Guide™

The Definitive Guide to Scrum:
The Rules of the Game

*November 2017*

*Developed and sustained by Scrum creators: Ken Schwaber and Jeff Sutherland*

# Table of Contents

## Purpose of the Scrum Guide

Scrum is a framework for developing, delivering, and sustaining complex products. This Guide contains the definition of Scrum. This definition consists of Scrum's roles, events, artifacts, and the rules that bind them together. Ken Schwaber and Jeff Sutherland developed Scrum; the Scrum Guide is written and provided by them. Together, they stand behind the Scrum Guide.

## Definition of Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum is:

- Lightweight
- Simple to understand
- Difficult to master

Scrum is a process framework that has been used to manage work on complex products since the early 1990s. Scrum is not a process, technique, or definitive method. Rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and work techniques so that you can continuously improve the product, the team, and the working environment.

The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.

The rules of Scrum bind together the roles, events, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout the body of this document.

Specific tactics for using the Scrum framework vary and are described elsewhere.

## Uses of Scrum

Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide, to:

1. Research and identify viable markets, technologies, and product capabilities;
2. Develop products and enhancements;
3. Release products and enhancements, as frequently as many times per day;
4. Develop and sustain Cloud (online, secure, on-demand) and other operational environments for product use; and,
5. Sustain and renew products.

Scrum has been used to develop software, hardware, embedded software, networks of interacting function, autonomous vehicles, schools, government, marketing, managing the operation of organizations and almost everything we use in our daily lives, as individuals and societies.

As technology, market, and environmental complexities and their interactions have rapidly increased, Scrum's utility in dealing with complexity is proven daily.

Scrum proved especially effective in iterative and incremental knowledge transfer. Scrum is now widely used for products, services, and the management of the parent organization.

The essence of Scrum is a small team of people. The individual team is highly flexible and adaptive. These strengths continue operating in single, several, many, and networks of teams that develop, release, operate and sustain the work and work products of thousands of people. They collaborate and interoperate through sophisticated development architectures and target release environments.

When the words "develop" and "development" are used in the Scrum Guide, they refer to complex work, such as those types identified above.

## Scrum Theory

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk.

Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

### Transparency

Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

For example

- A common language referring to the process must be shared by all participants; and,
- Those performing the work and those inspecting the resulting increment must share a common definition of "Done".

### Inspection

Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

### Adaptation

If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation, as described in the *Scrum Events* section of this document:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Scrum Values

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum roles, events, and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values. People personally commit to achieving the goals of the Scrum Team. The Scrum Team members have courage to do the right thing and work on tough problems. Everyone focuses on the work of the Sprint and the goals of the Scrum Team. The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work. Scrum Team members respect each other to be capable, independent people.

## The Scrum Team

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of "Done" product ensure a potentially useful version of working product is always available.

### The Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one can force the Development Team to work from a different set of requirements.

## The Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. A "Done" increment is required at the Sprint Review. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness.

Development Teams have the following characteristics:

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members, regardless of the work being performed by the person;
- Scrum recognizes no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations, or business analysis; and,
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

### Development Team Size

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination. Large Development Teams generate too much complexity for an empirical process to be useful. The Product Owner and Scrum Master roles are not included in this count unless they are also executing the work of the Sprint Backlog.

## The Scrum Master

The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values.

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

### Scrum Master Service to the Product Owner

The Scrum Master serves the Product Owner in several ways, including:

- Ensuring that goals, scope, and product domain are understood by everyone on the Scrum Team as well as possible;
- Finding techniques for effective Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Understanding product planning in an empirical environment;
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
- Understanding and practicing agility; and,
- Facilitating Scrum events as requested or needed.

### Scrum Master Service to the Development Team

The Scrum Master serves the Development Team in several ways, including:

- Coaching the Development Team in self-organization and cross-functionality;
- Helping the Development Team to create high-value products;
- Removing impediments to the Development Team's progress;
- Facilitating Scrum events as requested or needed; and,
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

### Scrum Master Service to the Organization

The Scrum Master serves the organization in several ways, including:

- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development;
- Causing change that increases the productivity of the Scrum Team; and,
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

## Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events, such that every event has a maximum duration. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved, ensuring an appropriate amount of time is spent without allowing waste in the process.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

### The Sprint

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product increment.

Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cost.

### Cancelling a Sprint

A Sprint can be cancelled before the Sprint time-box is over. Only the Product Owner has the authority to cancel the Sprint, although he or she may do so under influence from the stakeholders, the Development Team, or the Scrum Master.

A Sprint would be cancelled if the Sprint Goal becomes obsolete. This might occur if the company changes direction or if market or technology conditions change. In general, a Sprint should be cancelled if it no longer makes sense given the circumstances. But, due to the short duration of Sprints, cancellation rarely makes sense.

When a Sprint is cancelled, any completed and "Done" Product Backlog items are reviewed. If part of the work is potentially releasable, the Product Owner typically accepts it. All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog. The work done on them depreciates quickly and must be frequently re-estimated.

Sprint cancellations consume resources, since everyone regroups in another Sprint Planning to start another Sprint. Sprint cancellations are often traumatic to the Scrum Team, and are very uncommon.

## Sprint Planning

The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

### Topic One: What can be done this Sprint?

The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

The input to this meeting is the Product Backlog, the latest product Increment, projected capacity of the Development Team during the Sprint, and past performance of the Development Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint.

During Sprint Planning the Scrum Team also crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.

### Topic Two: How will the chosen work get done?

Having set the Sprint Goal and selected the Product Backlog items for the Sprint, the Development Team decides how it will build this functionality into a "Done" product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

The Development Team usually starts by designing the system and the work needed to convert the Product Backlog into a working product Increment. Work may be of varying size, or estimated effort. However, enough work is planned during Sprint Planning for the Development Team to forecast what it believes it can do in the upcoming Sprint. Work planned for the first days of the Sprint by the Development Team is decomposed by the end of this meeting, often to units of one day or less. The Development Team self-organizes to undertake the work in the Sprint Backlog, both during Sprint Planning and as needed throughout the Sprint.

The Product Owner can help to clarify the selected Product Backlog items and make trade-offs. If the Development Team determines it has too much or too little work, it may renegotiate the selected Product Backlog items with the Product Owner. The Development Team may also invite other people to attend to provide technical or domain advice.

By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

### Sprint Goal

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

As the Development Team works, it keeps the Sprint Goal in mind. In order to satisfy the Sprint Goal, it implements functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

## Daily Scrum

The Daily Scrum is a 15-minute time-boxed event for the Development Team. The Daily Scrum is held every day of the Sprint. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity.

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint.

The structure of the meeting is set by the Development Team and can be conducted in different ways if it focuses on progress toward the Sprint Goal. Some Development Teams will use questions, some will be more discussion based. Here is an example of what might be used:

- What did I do yesterday that helped the Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or replan, the rest of the Sprint's work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master teaches the Development Team to keep the Daily Scrum within the 15-minute time-box.

The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting.

Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.

## Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.

This is at most a four-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendees understand its purpose. The Scrum Master teaches everyone involved to keep it within the time-box.

The Sprint Review includes the following elements:

- Attendees include the Scrum Team and key stakeholders invited by the Product Owner;
- The Product Owner explains what Product Backlog items have been “Done” and what has not been “Done”;
- The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;
- The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;
- The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed);
- The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;
- Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,
- Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

### Sprint Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose.

The Scrum Master ensures that the meeting is positive and productive. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of “Done”, if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

## Scrum Artifacts

Scrum’s artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

## Product Backlog

The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is never complete. The earliest development of it lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. If a product exists, its Product Backlog also exists.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate, and value. Product Backlog items often include test descriptions that will prove its completeness when "Done."

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product. A Product Backlog attribute that groups items may then be employed.

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail. Product Backlog items that will occupy the Development Team for the upcoming Sprint are refined so that any one item can reasonably be "Done" within the Sprint time-box. Product Backlog items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in a Sprint Planning. Product Backlog items usually acquire this degree of transparency through the above described refining activities.

The Development Team is responsible for all estimates. The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate.

### Monitoring Progress Toward Goals

At any point in time, the total work remaining to reach a goal can be summed. The Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Various projective practices upon trending have been used to forecast progress, like burn-downs, burn-ups, or cumulative flows. These have proven useful. However, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

## Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a “Done” Increment.

The Sprint Backlog makes visible all the work that the Development Team identifies as necessary to meet the Sprint Goal. To ensure continuous improvement, it includes at least one high priority process improvement identified in the previous Retrospective meeting.

The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

### Monitoring Sprint Progress

At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Development Team tracks this total work remaining at least for every Daily Scrum to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress.

### Increment

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be "Done," which means it must be in useable condition and meet the Scrum Team's definition of "Done." An increment is a body of inspectable, done work that supports empiricism at the end of the Sprint. The increment is a step toward a vision or goal. The increment must be in useable condition regardless of whether the Product Owner decides to release it.

## Artifact Transparency

Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis. To the extent that the artifacts are incompletely transparent, these decisions can be flawed, value may diminish and risk may increase.

The Scrum Master must work with the Product Owner, Development Team, and other involved parties to understand if the artifacts are completely transparent. There are practices for coping with incomplete transparency; the Scrum Master must help everyone apply the most appropriate practices in the absence of complete transparency. A Scrum Master can detect incomplete transparency by inspecting the artifacts, sensing patterns, listening closely to what is being said, and detecting differences between expected and real results.

The Scrum Master's job is to work with the Scrum Team and the organization to increase the transparency of the artifacts. This work usually involves learning, convincing, and change. Transparency doesn't occur overnight, but is a path.

## Definition of “Done”

When a Product Backlog item or an Increment is described as *“*Done”, everyone must understand what “Done*”* means. Although this may vary significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of “Done” for the Scrum Team and is used to assess when work is complete on the product Increment.

The same definition guides the Development Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team’s current definition of “Done.”

Development Teams deliver an Increment of product functionality every Sprint. This Increment is useable, so a Product Owner may choose to immediately release it. If the definition of "Done" for an increment **is** part of the conventions, standards or guidelines of the development organization, all Scrum Teams must follow it as a minimum.

If "Done" for an increment is **not** a convention of the development organization, the Development Team of the Scrum Team must define a definition of “Done” appropriate for the product. If there are multiple Scrum Teams working on the system or product release, the Development Teams on all the Scrum Teams must mutually define the definition of “Done.”

Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work together.

As Scrum Teams mature, it is expected that their definitions of “Done” will expand to include more stringent criteria for higher quality. New definitions, as used, may uncover work to be done in previously “Done” increments. Any one product or system should have a definition of “Done” that is a standard for any work done on it.

## End Note

Scrum is free and offered in this Guide. Scrum's roles, events, artifacts, and rules are immutable and although implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

## Acknowledgements

### People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

### History

Ken Schwaber and Jeff Sutherland worked on Scrum until 1995, when they co-presented Scrum at the OOPSLA Conference in 1995. This presentation essentially documented the learning that Ken and Jeff gained over the previous few years, and made public the first formal definition of Scrum.

The history of Scrum is described elsewhere. To honor the first places where it was tried and refined, we recognize Individual, Inc., Newspage, Fidelity Investments, and IDX (now GE Medical).

The Scrum Guide documents Scrum as developed, evolved, and sustained for 20-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide you with patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.