

G2SWS

A Standard for the next generation Ground Software in Galileo

G2G-REQ-ESA-GSEG-X-0005 i1

01/09/2025

1. Introduction
2. Scalable, iterative and incremental
3. Software Approach
4. Criteria to start/end DoR/DoD
5. Software Robustness
6. Qualification and Acceptance
7. Product Assurance
8. Documentation in an iterative world
9. ISO/IEC/IEEE 32675:2022

What's Galileo? And what's GSWS?

Europe's independent Global Navigation Satellite System

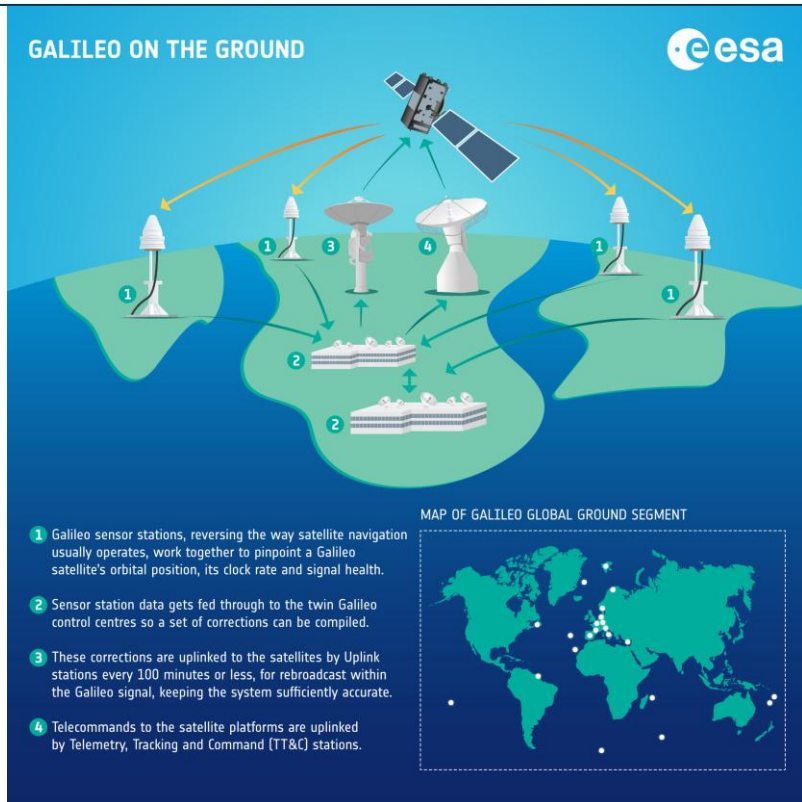
Most accurate GNSS, better than GPS 😊

GSWS-G: Galileo (1st Generation) Software Standard for Ground Segment

- Flat criticality level. DAL-D
- V-cycle

Change of paradigm

G2SWS: Galileo 2nd Generation Software Standard for Ground Segment



Galileo Navigation/Ground Segment

GALILEO IN NUMBERS

1995

ESA starts GNSS R&D under ARTES-9 programme

1998 - 2000

ESA, EC and Eurocontrol enter cooperation and Galileo becomes EU programme

2005 - 2008

First satellites (GIOVE-A and B) enable first Galileo signal in space

2009

Start of ground segment development

2011 - 2013

Launch of In-Orbit Validation satellites and first position fix

SERVICES



OPEN SERVICE



HIGH ACCURACY SERVICE



SEARCH AND RESCUE SERVICE



TIMING SERVICE



SAFETY OF LIFE SERVICE



PUBLIC REGULATED SERVICE



AUTHENTICATION SERVICE



EMERGENCY WARNING SERVICE

THE SYSTEM, USERS AND INDUSTRY

4+ billion smartphone users worldwide

1:50 return of public investment to European industry

3 managing partners (EC, ESA and EUSPA)

SPACE SEGMENT

Satellites

12+ launches

6 ultra-precise atomic clocks

5 frequency bands

12 types of signals

2 payloads

2014

Deployment of worldwide ground segment

2015

Launch of 12 satellites in one year

2016

Declaration of Initial Services

2016 - 2023

Roll out of additional services

2024

Galileo Second Generation under development

2025 +

Completion of First Generation and start of Second Generation launches



Orbit

3 medium-Earth orbit planes at

23 222 km above the Earth,

inclined **56** degrees to the equator

GROUND SEGMENT

16 GROUND SENSOR STATIONS IN 5 CONTINENTS

20 UPLINK STATIONS

7 TELEMETRY, TRACKING AND COMMAND STATIONS

4 CONTROL CENTRES

20+ SERVICES FACILITIES AND TESTBEDS

Constellation

2 GIOVE satellites

38 First Generation satellites (30+ in orbit)

12 Second Generation satellites (in production)

GALILEO IN NUMBERS

1995

ESA starts GNSS R&D under ARTES-9 programme

1998 - 2000

ESA, EC and Eurocontrol enter cooperation and Galileo becomes EU programme

2005 - 2008

First satellites (GIOVE-A and B) enable first Galileo signal in space

2009

Start of ground segment development

2011 - 2013

Launch of In-Orbit Validation satellites and first position fix

SERVICE

1 M ACCUR

OPEN SERV

SAFETY OF SERVICE

THE SYS

4+ b

1:50

SPACE S

Satel

12+ la

6 ultra-p

**GALILEO SOFTWARE
STANDARD
FOR GROUND
GSWS-G**

Constellation

2

GIOVE satellites

38

First Generation satellites (30+ in orbit)

12

Second Generation satellites (in production)

es at

the Earth,

the equator

20

UPLINK STATIONS

CONTROL CENTRES

TRACKING
AND STATIONS

20+

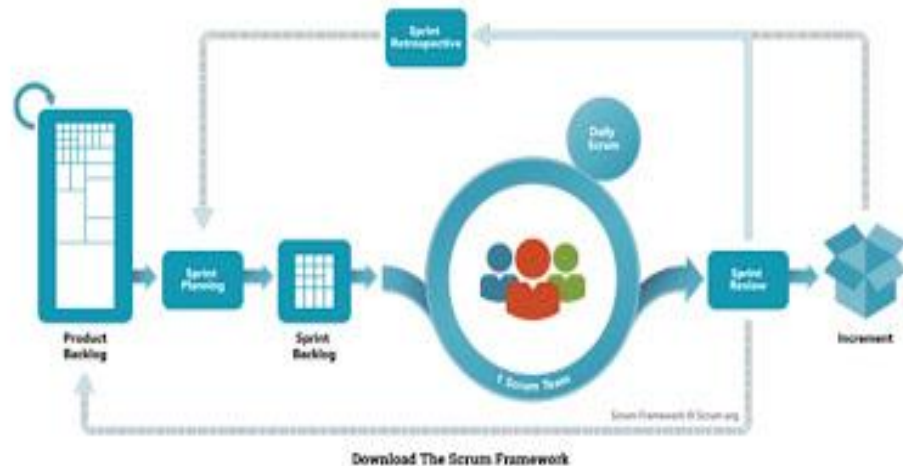
SERVICES FACILITIES
AND TESTBEDS

Iterative, Incremental, scalable to size/effort => faster feedback, adaptable

Small => Medium => BIG

Small Team:

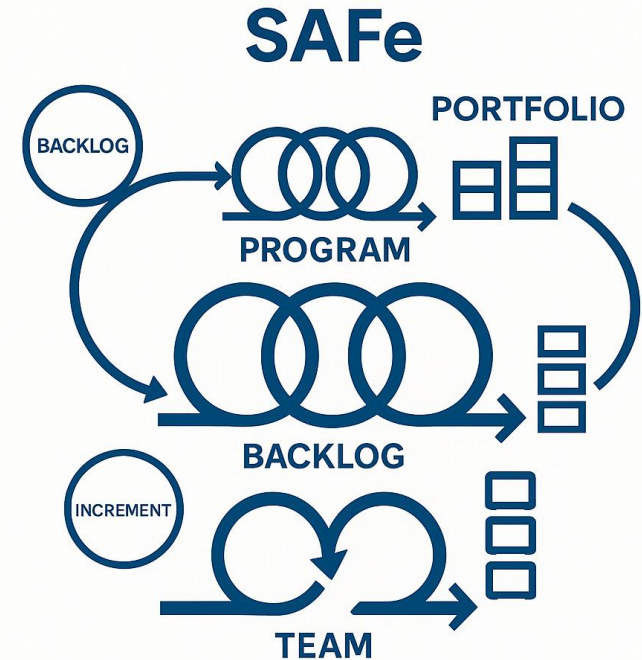
- Small projects up to **9/10 persons** is the minimum team
- Short cycles **1 to 4 weeks**
- 3 roles as in Scrum
- **Continuous improvement** in “retrospective”
- Plan bugs along with new features using a **prioritised** backlog
- Demo to get **feedback**
- **Roadmaps** for long term plans and external synchronisation



Scalable with levels:

- Medium can have **2 levels with ~10 Teams**, large up to **3 levels**
- ~100 people for 2 levels, a **few hundreds** for 3
- Bigger cycles/iterations as multiple of small, i.e. 6 cycles x 2 weeks
- **Replicate roles** at each level
- Continuous **improvement** at each level
- **Demos**, Roadmaps, backlog
- Roadmaps with several horizons: **short term**, **medium term**, **long term**.

Scaled Agile Framework (SAFe) is compatible with G2SWS.



Software is decomposed into small bits of functionality: User stories at different levels if scaled.

Small Team:

- Using cycles with **objectives at the beginning and end**
- Beginning of cycle objectives
 - User Stories are ready for the target level: **Definition of Ready**
 - Software Problems are ready for investigation and fix.
- End of cycle objectives
 - User Stories are done for the target level.
 - Done by **Definition of Done**
 - **Traced up/down** to other user stories
 - Software Problems are done: Fixed
- **Objectives are formal: independently verified by PA with evidence**



[GSWG-52110]

The Definition of Ready - DoR shall include as a minimum:

- Objective: Clearly defined purpose according to the format defined in SDMP.
- Benefit Hypothesis: value obtained from this User Story.
- Acceptance criteria: detailed and non-ambiguous definition and in line with User Story definition.
- Priority defined
- Traced up to Higher level User Stories (or requirements if relevant)

All according to the format defined in the SDMP for the applicable level.

§ [All iterations] - (SDMP)

Note: The Benefit Hypothesis should in general follow a format such as:

"As a <Role> I want to have <this> so that I can do <that> with"

[GSWG-52210]

The Definition of Done - DoD shall include as a minimum:

- traceability to lower levels (user stories of a lower level/design/code)
- check Static and Security Code Analysis rules. Gate passed.
- check coding and design guidelines/standards. Gate passed.
- check quality model metrics and thresholds. Gate passed.
- Design updated.
- Usage Manual updated.
- Tests defined and fully aligned to Acceptance Criteria.
- Other specific tests.
- tests passed, including robustness:
 - minimum code coverage achieved grouping all levels: Unit/Integration/Element Tests. Gate passed.
 - resource usage collected (RAM/Disk/Network) only for higher-level user stories
- COTS dependencies license compliance to contractual requirement. Gate passed.

§ [End of iteration] - (SSD, SBR, TR, SPAR, PSJF, SOM, SDD, SRL, VCD, SVTS, SATP)

Note: See section 6.4 for pass gates.

Note: Gate not passed means not Done (or not compliant for licenses).

Note: Resource usage are collected individually per high-level user story. See ## [GSWG-52270].

Note: Other specific tests could be mandated, i.e. Behaviour Driven or Consumer Driven Contract.

Definition of Ready/Done

[GSWG-52110]

The Definition of Ready - Do

- Objective: Clearly def
- Benefit Hypothesis: v
- Acceptance criteria: definition.
- Priority defined
- Traced up to Higher le

All according to the format de

§ [All iterations] - (SDMP)

Note: The Benefit Hypothesis

"As a <Role> I want t

Ready

- ✓ Objective
- ✓ Benefit Hypothesis
- ✓ Acceptance Criteria
- ✓ Priority
- ✓ Traced to User Stories

Done

- ✓ Traceability to Lower Levels
- ✓ Static/Security Code Analysis
- ✓ Coding/Design Guidelines
- ✓ Usage Manual
- ✓ Tests including Robustness

rel/design/code)

passed.

passed.

sed.

1.

all levels: Unit/Integration/Element Tests.

c) only for higher-level user stories

il requirement. Gate passed.

3D, SRL, VCD, SVTS, SATP)

licenses).

el user story. See ## [GSWG-52270].

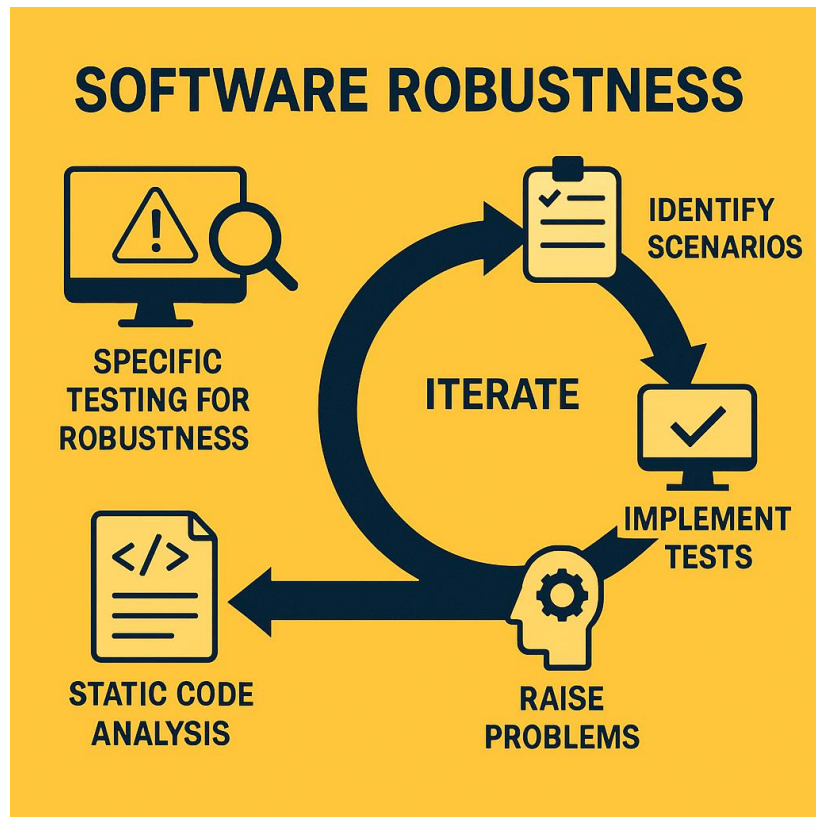
Driven or Consumer Driven Contract.

Software Robustness needs to be **enforced as an independent target** with 2 main measures:

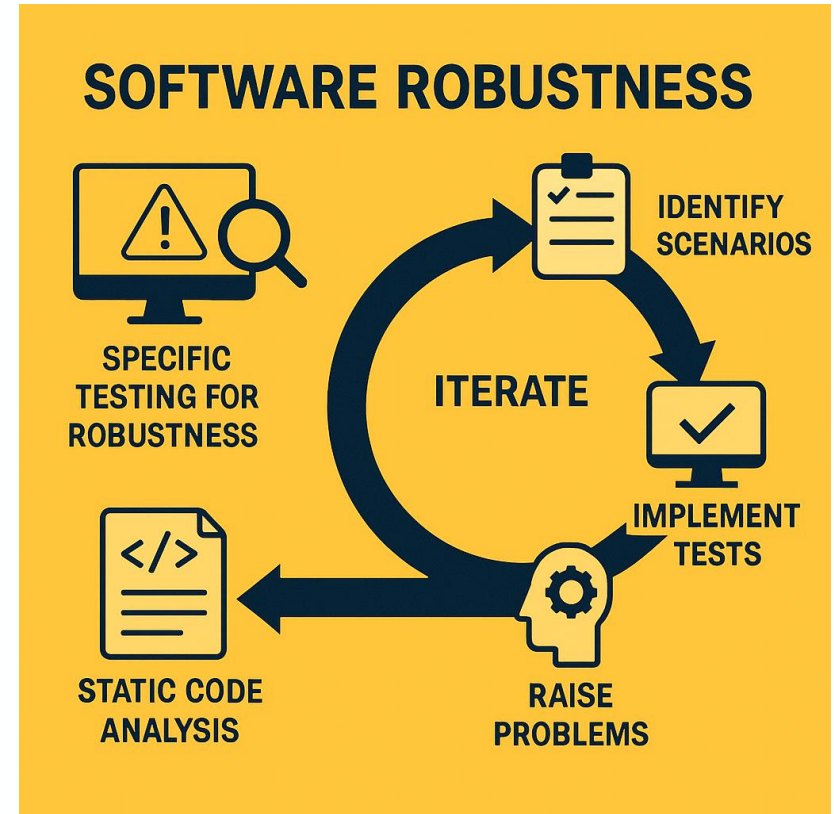
- Specific testing for robustness, 'a-la' **chaos engineering**
- Static Code Analysis.

Software testing is always included in standards.

In previous generation of GSWS as well, but it was often **neglected** or simply **ignored in favour of the nominal case.**



- Robustness **achieved iteratively** in cycles along with development
- Identify and define robustness scenarios **using RAMS inputs**.
- Identify key parameters with **target values**
- Implement the robustness test or tests to cover the scenario
- Test are **always automated**
- Run it, and **collect** your key **parameters**
- Raise **problems** for parameters not met, or behaviour considered abnormal



Continuous Qualification, that is, qualify every cycle.

[GSWG-56110]

Qualification shall be achieved by validating implemented user stories at the end of each iteration, plus

validating all user stories from all previous iterations, ensuring verification of high-level requirements, verification of the processes followed, and issuing formal documentation for candidate versions or full qualification.

[GSWG-56150]

A Software Qualification Review shall be called as needed with the following objectives:

- Scope and purpose for this qualification review.
- Verification of all High-level user stories:
 - o Done according to their DoD
 - o Review the Acceptance criteria.
 - o Verify all user stories contributing to High-level user stories are also Done.
- Verify software (see Software in 7.3) and all issued documents are under configuration control.
- Review of the PSJF, including SRL (IPR, licensing).
- Review of known SPRs/NCRs, and closed SPRs/NCRs since last qualification.
- Verify PA Status: SPAR (including SVR), CoC.
- Review of RFWs.
- Review of status of obsolescence regarding COTS.
- Security acceptance criteria as per section 5.7.2.
- Evaluation of readiness for defined purpose

[GSWG-56110]

- validating all user stories from all projects
- ensuring verification of high-level requirements
- verification of the processes followed
- formal documentation for candidate selection
- qualification.

CONTINUOUS QUALIFICATION QUALIFIES EVERY CYCLE

- ✓ VALIDATE USER STORIES AT END OF EACH ITERATION

- ✓ VALIDATE PREVIOUS ITERATION USER STORIES

✓ **VERIFY
HIGH-LEVEL
REQUIREMENTS**

✓ **PROCESS VERIFICATION**

✓ EXPOSURE TO REPRESENTATIVE ENVIRONMENT

✓ OPERATOR USAGE
UNDER
REPRESENTATIVE
CONDITIONS

✓ ADJUSTMENTS
AFTER AT LEAST 2
ITERATIONS

er stories:

ng to High-level user stories are

in 7.3) and all issued documents

SRL (IPR, licensing).
and closed SPRs/NCRs since last

ng SVR), CoC.

ce regarding COTS.
per section 5.7.2.
ned purpose

- Attends ceremonies ## [GSWG-62305] and reviews ## [GSWG-62310]
- Verifies **Definition of Ready** criteria and evidences to support it ## [GSWG-62360]
- Verified **Definition of Done** criteria and evidences to support it ## [GSWG-62365]
- provides an iterative **Software Verification Report**, and **Product assurance Report** ## [GSWG-63010]
- Tracking Software Problems, and Non-conformance down to Waivers when **qualification** is achieved: ## [GSWG-62510], ## [GSWG-62520]
- And many more things..... Leading to qualification and acceptance.

G2SWS includes development with Product Assurance

- Attends ceremonies
- Verifies **Definition** 62360]
- Verified **Definition** 62365]
- provides an item **Report** ## [GSWG-62310]
- Tracking Software **qualification** in the development process
- And many more



for the
by Software

62310]
[GSWG-

[GSWG-
assurance

vers when

e.

This Standard removes all formal doc deliveries replacing it by an information delivery by tools that is every bit as formal as before but a lot faster as there's no signatures, and information is continuously accessible.

Document Delivery	Usage	Format/Accessability
Tagged Live Document (TLD)	Cycle/Iteration	Accessible via tools
Formal Document (FD)	Qualification/Acceptance	Formally issued and approved documents
Document Snapshot (DS)	External visibility	Informally issued documents

Documentation is under configuration control all the time, and only committed to documents when strictly required, i.e. qualification. All iterations are TLD, unless external informal visibility is needed so DS, and then for qualification FD.

With the main purpose of this standard being Agile development of Sw,
there's **no friction** foreseen with the **ISO/IEC/IEEE 32675:2022**.

Bonus: ISO/

Regarding this
software and t

much wider c
knowledge ma

The G2SWS i
Customer foc
like business/

With the main
there's **no fric**



ising on the
andard is
risks,

principles like
igned to them

opment of Sw,
5:2022.

G2SW5

THE NEXT GENERATION

