

Software Systems that Enable Agile Manufacturing at Scale



Presented by Kyle Racette <kyle@planet.com>

Authors: Kyle Racette, Andrew Fisher, Julien Rhodes, James Muehlner, Jacob Stern, Chester Gillmore

+ THE MISSION

Planet's mission is to image Earth's land mass every day, and make that data visible, accessible, and actionable. To achieve this mission, Planet designs, builds, and operates a large fleet of Earth-observing satellites. Planet is constantly improving its technology in order to meet the demands of the market while operating hardware in the challenging environment of space. Since the first launch in 2013, Planet has developed 14 major iterations of its medium resolution "Dove" spacecraft.

This "agile" approach to designing and building spacecraft intersects with the challenges posed by scaling manufacturing operations, culminating in a unique set of capabilities that Planet has mastered over the years. A strong software system that supports assembly, integration, and testing operations is a key part of Planet's manufacturing capabilities.

+ AGILE AEROSPACE

Quick design sprints mean we handle multiple hardware revisions and deviations at any given time.

Integration of new changes happens quickly. Hardware gets in the hands of engineers right away, so learning and documentation happen in real time.

Rework is part of the process. Rapid hardware design changes and responding to manufacturing process improvements means hardware is often reworked, both in a one-off fashion and systematically.

+ SCALING

Automation is essential for reducing overhead of common or repeated operations. Testing requires minimal human interaction and produces an immediate pass or fail result.

Consistent and repeatable design ensures that all lab stations are capable of running any test, and all interfaces to the satellite and ground support equipment are standard and consistent.

Expect the unexpected. Tooling should support the process, but should provide enough flexibility to break out of the process when needed.

Test early, test often. Test as much as possible at the subassembly level. Test critical subsystems in between each stage of assembly and integration.

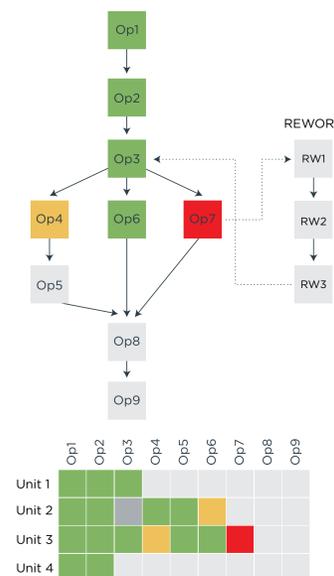
+ MANUFACTURING EXECUTION SYSTEM

Processes define the order of assembly and test operations. Some operations may be non-sequential.

Nominal processes define the ideal set of operations. Each part number has one nominal process.

Rework processes are used for non-nominal situations, such as fixing a hardware failure or applying a sudden design change.

Work-in-progress (WIP) trackers are essential for keeping track of everything that's going on.

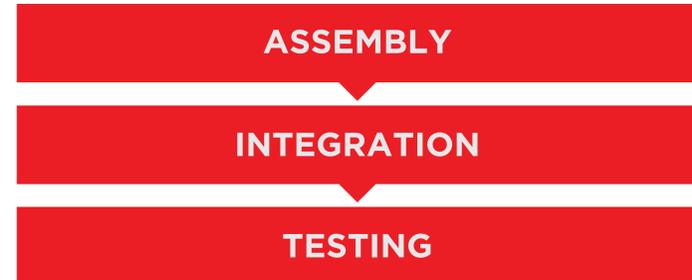
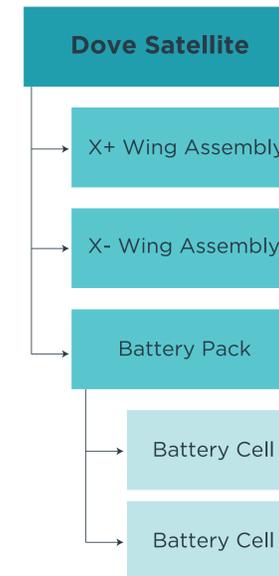


Genealogy tracking tools document which sub-assemblies were integrated into parent assemblies.

Traceability is key. All actions are logged with timestamps and the user who performed them.

Photos and comments are captured by technicians during most assembly and integration operations.

Tablets and mobile devices are the platform of choice for technicians due to the touch interface and high-quality integrated cameras.



Test Results

Consist of..

- Subresults / limits
- Freeform data
- Attachments
- Metadata

Are used to...

- View
- Analyze
- Compare

Test results are generated by **automated tests**, and have a **standard** and **consistent** format. From the subresults, an immediate **pass / fail** status is derived.

Test results can be looked up using a **unique ID** assigned to each result, or through a combination of the **serial number**, **test key**, and **stage of AIT**.

Key Features of Software in an Agile Environment

Microservices allow codebases to remain small, and applications to evolve at their own pace.

Code Reviews, Automated Testing, and Release Checklists ensure stable, reliable, and high-quality software.

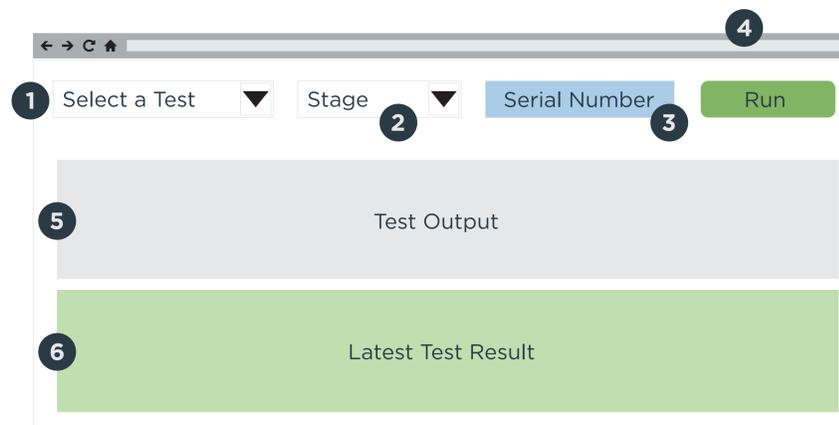
Mix of home-built and enterprise software: Off-the-shelf software is useful when the needs are clearly defined and the software satisfies those needs. Software that defines too much of the process can lock an organization in to certain ways of doing things. If the requirements or the process is not well defined, then it may pay off to develop that software in house.

Common Interfaces and a shared domain model facilitate integration of software components.

Open Source and Developer Friendly software provides more flexibility, allowing the software to adapt to the organization.

+ AUTOMATED TEST RUNNER

Dozens of test stations test every subsystem of the spacecraft and its subassemblies. Each station is powered by an in-house test automation platform and user interface.



1 Each test station is capable of running any test, provided the correct ground support equipment is connected.

Tests are implemented using an in-house framework. When new tests are deployed to a station, they will instantly appear here.

4 The test runner UI runs in a browser, and can be operated completely remotely.

2 Some tests are run at different stages of AIT, such as before and after environmental testing. Their behavior can be varied depending on the selected stage.

All fields can be input with barcode scanners, keeping the user interface simple and easy to operate.

5 Test output streams in real-time. All output is logged and attached to the test result.

3 The serial number encodes information about the device under test, including what type of hardware it is and its revision.

Tests can alter their behavior based on this information.

6 The latest test result appears as soon as a test and serial number are selected.

The test status is apparent as soon as a test is complete.