CubeSat Flight Software Workshop

FSW Development Process

Aadil Rizvi
Flight Software Engineer
6/3/2019

Jet Propulsion Laboratory
California Institute of Technology

Agenda

- **FSW Development Process Overview**

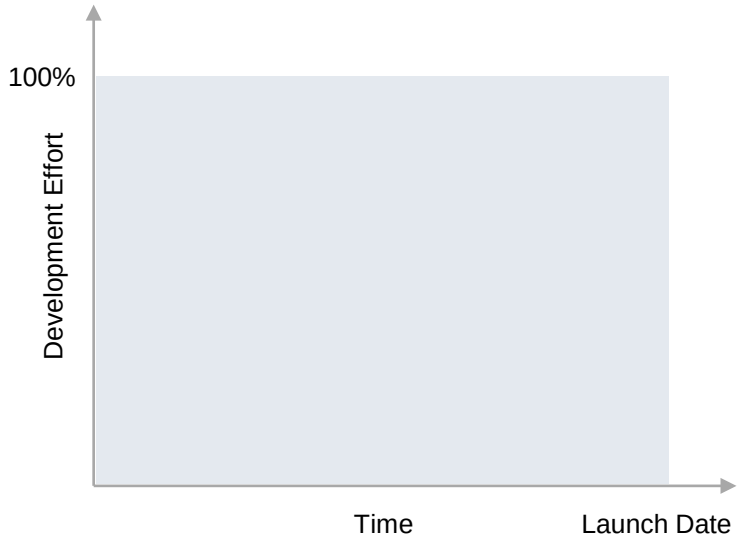
- **FSW Subsystem Level Process**
  - Requirements Phase
  - Prototyping
  - Design Phase
  - Implementation Phase
  - Version Control
  - Verification Phase
  - Delivery Review
  - Change Requests and Maintenance

- **Component Level Process**
  - Software Design Document
  - Implementation
  - Unit Testing
  - Integrated Testing
  - Code Analysis and Review
  - Checklist

- **Status Reports**
  - Weekly Progress Summary
  - Issue Tracking

FSW Development Process Overview

Student CubeSat Project Timeline



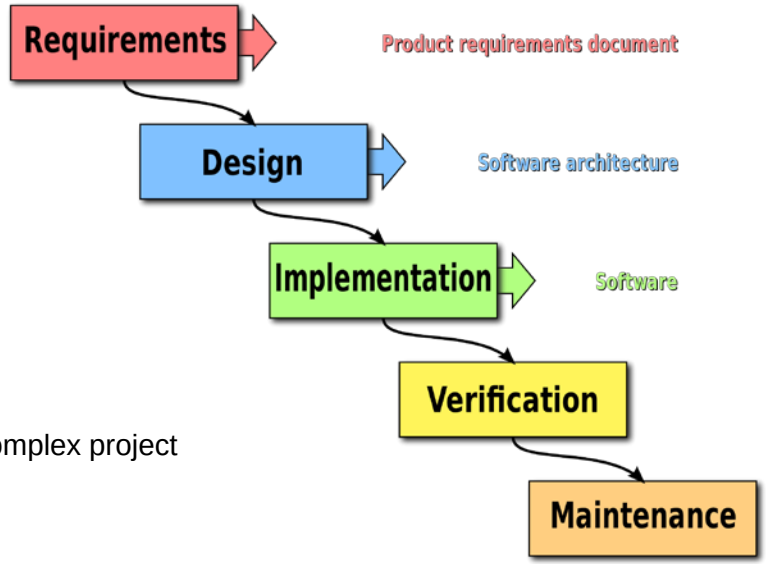
FSW Development Process Overview

Why is it important to have a development process?

Increases: -
- Reliability
- Testability
- Maintainability
- Portability

- Waterfall model fits in line with deadline driven development

- The right level of process is important
  - Too much process can bog you down
  - Too little process makes it impossible to manage a complex project
  - Either can lead to bad outcomes



Waterfall Model (Peter Kemp / Paul Smith [CC BY 3.0])

FSW Development Process Overview

Development Phases

Requirements	- Provide measurable constraints and characteristics from concept of operations
Design	- Provides blue print for software implementation given a set of requirements
Implementation	- Provides a testable product for verification
Verification	- Ensures implementation functionality and correctness

- Each phase has a review to ensure readiness for the next phase and address any issues

FSW Subsystem Level Process

Requirements Phase

- Why is it important to gather requirements: -
    - Map from concept of operations to specific capabilities that can be designed and implemented
    - Manages assumptions
    - Heads off disagreements/misunderstanding between designers/implementors and their stakeholders
        - "That's not what I wanted you to build!" Or
        - "That's not how I assumed it would work!"
    - Provides the structure for
        - Measuring progress of design and implementation
        - Verifying that we have built/delivered what is needed

FSW Subsystem Level Process

Requirements Derivation

- Understand project level requirements and concept of operations (ConOps) i.e. what is needed for the project
  - Decompose into various software components at high level
  - Functional breakdown rather than design

- Artifact: Requirements specification document

- Conduct review of requirements

Example requirement

FSW Interfaces

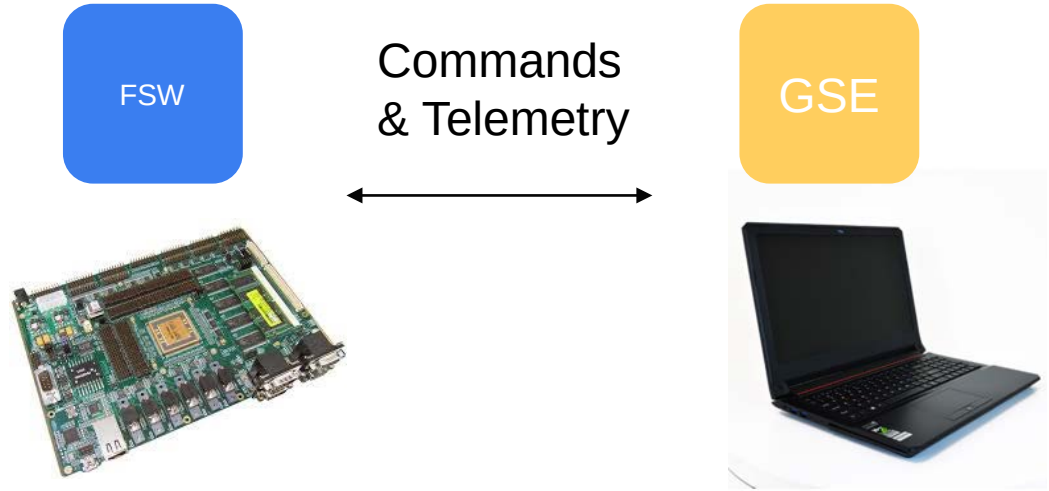
REQ ID	Short Title	Level 4 Requirement	Rationale	L3 Parent	V&V Strategy	V&V Plan
L4-FSW-1	Radio Uplink Interface	The FSW shall interface with the XYZ radio over an SPI connection to perform uplink for ground commands and files	FSW must provide software interface to receive uplinked data	L3-FSW-34, L3-FSW-35	Test	Test uplink frame formats with GDS. Both command and file uplink frames will be tested for appropriate decoding and execution. Part of FSW Rel-1.0 integrated testing.

What makes a good requirement?

FSW Subsystem Level Process

Proof of concept and Prototyping

- Target OS and hardware platform
- Compile and execute software on target
- Communicate over planned interfaces
- Data bandwidth and performance analysis



FSW Subsystem Level Process

Design Phase

- Trade studies and prototyping

- Component list
  - Services
  - Communication
  - Hardware managers
  - Hardware drivers
  - Guidance and control
  - Science
  - Fault protection and mode management

- Various design views
  - Block diagrams
  - Sequence diagrams
  - Data flow diagrams
  - State transition diagrams
  - Class diagrams

- Resource utilization, performance and concurrency issues discussion

- List of planned releases

Conduct design review
Artifact: software architecture and design documentation

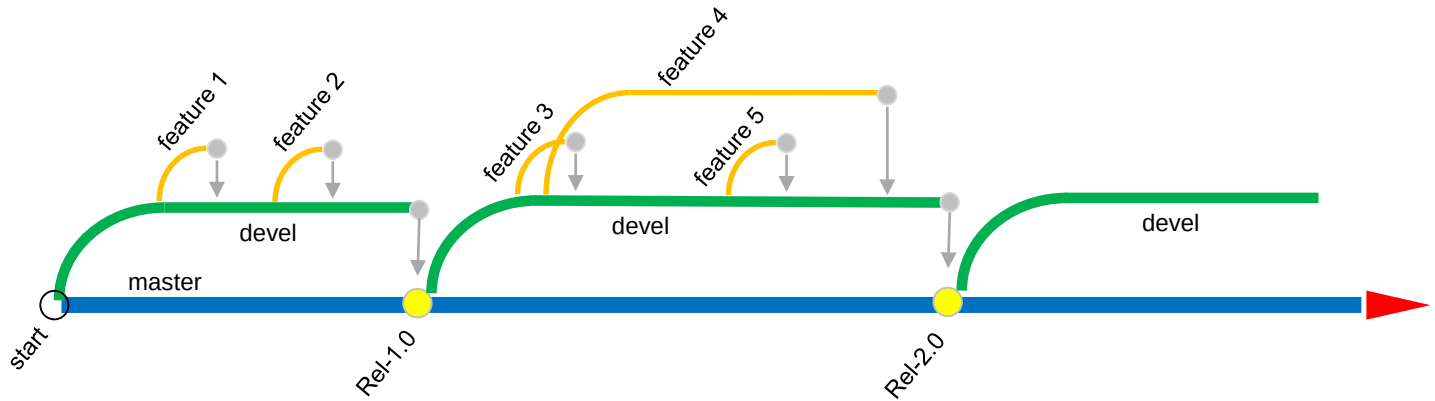
FSW Subsystem Level Process

Implementation Phase

- Coding
  - With good design, this should not be too complicated
  - May require design updates
  - Majority of design expected to be completed in design phase
- Unit Testing
  - Component level
- Deployment
  - Functional integration of software components
- Conduct code reviews

FSW Subsystem Level Process

Version Control



= Master Branch
= Devel Branch
= Feature Branch

= Branch Merge

= Release Tag

Git Workflow

FSW Subsystem Level Process

Verification Phase

- Critical to overall software functionality and mission success
- Catching bugs early is cheaper and easier to fix
- Driven by requirements verification
    - Performed using test scripts executed against a release deployment

Artifact: Requirements verification and validation matrix

REQ ID	Short Title	Level 4 Requirement	Rationale	L3 Parent	V&V Strategy	Status	V&V IDs
NEASC-L4-FSW-3	Data Storage Interface	The FSW shall interface with the non-volatile data storage memory on board the flight CDH unit for read/write access for atleast 4 GB] bytes.	FSW needs access to this memory to manage science and engineering data	NEAS-FS-L3-32	Test	PASS	jpl_ffs-VI-1 thru jpl_ffs-VI-7, prmDb-VI-1

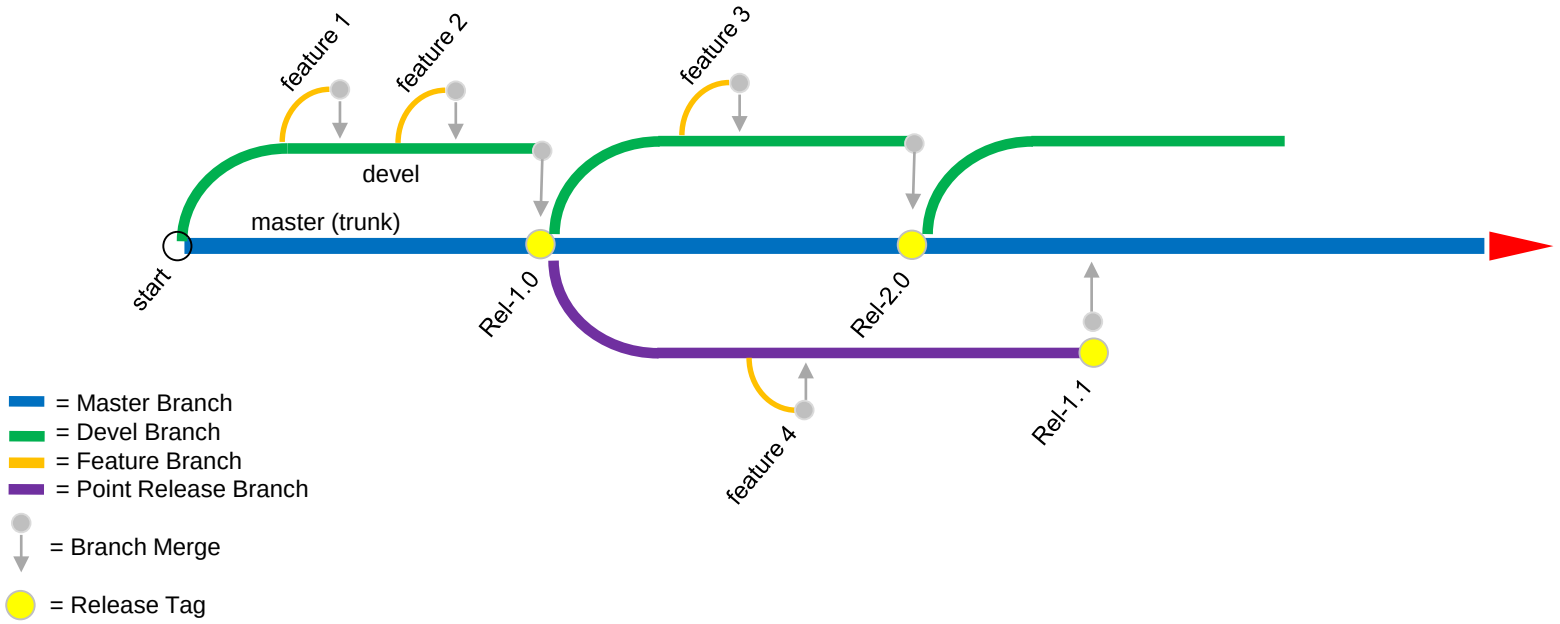
FSW Subsystem Level Process

Delivery Review

- Release description document (RDD)
  - Change log
  - Version Identification
  - Project Overview and Release Description
  - Controlling Documents
  - Test Reports
  - Requirements Verification Summary
  - Idiosyncrasies and Known Issues
  - Problem Disposition
  - Detailed Contents
- Users guide
  - Operational constraints
  - Usage guidelines
- Software design documents

FSW Subsystem Level Process

Change Requests and Maintenance



Component Level Process

Software Design Document

- Component overview
- Component level requirements
- Design
    - Component block diagram
    - Sequence, dataflow, state transition, class diagrams
    - Port List
    - Custom data types
    - State
    - Port Behaviors
    - Commands
    - Telemetry
    - Events
- Unit Test Output and Coverage Results
- Reference datasheets and other technical documents as applicable

Component Level Process

Implementation

- Port handler behaviors
- State management
- Command handlers
- Telemetry & events
- Topology integration
- Build with deployment

Component Level Process

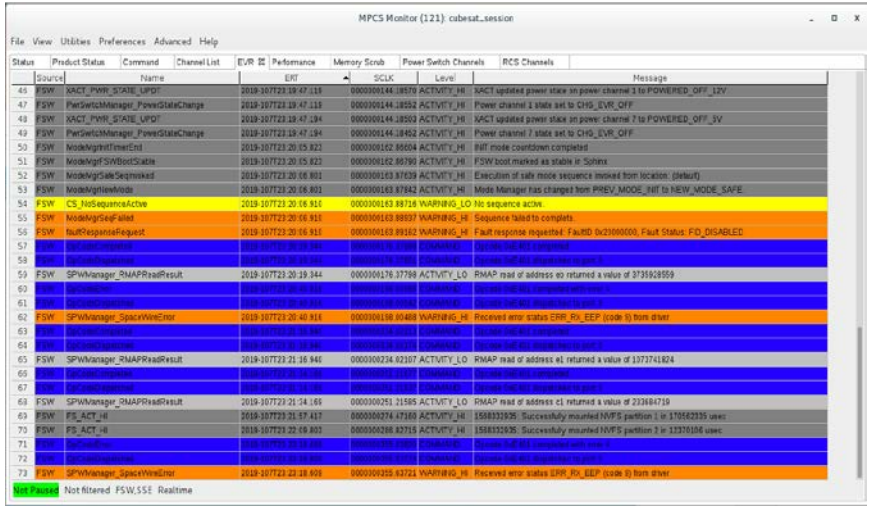
Unit Testing

- F Prime unit test harness
- Traceability to component level requirements
- Code coverage analysis
- Unit test output and coverage results

Component Level Process

Integrated Testing

- Test venue
- Test scripts
- Test reports
- Requirements V&V




```
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO                  ===== Beginning Testxactif05Cmd05 =====
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO                  Test Case Description:
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO                    Send command 5
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO  MTAK_PASS        EHA found/verified XACT_CMD_ACCEPT_COUNT (XACT-0275) is DN 32
2019-039T01:15:29.008  FSWSR  0000002983.27052  MPCS  FIT INFO                  Sending FSW Command (number=218): XACT_NO_OP
2019-039T01:15:29.540  FSWSR  0000002983.27052  MPCS  FSW CMD   XACT_NO_OP      [1]
2019-039T01:15:29.543  FSWSR  0000002983.27052  MPCS  LOG INFO  General         FSW::Forwarded transmitted command(s) to SOCKET.                  [1]
2019-039T01:15:32.103  1970-0011  0000002992.83012  FSW RT  EVR COMMAN  OpCodeDispatched   Opcode 0xF605 dispatched to port 7                [15]
2019-039T01:15:32.339  1970-0011  0000002993.09743  FSW RT  EVR COMMAN  OpCodeCompleted    Opcode 0xF605 completed                           [16]
2019-039T01:15:35.008  FSWSR  0000002993.09743  MPCS  FIT INFO                  Opcode 0xF605 completed
2019-039T01:15:35.008  FSWSR  0000002993.09743  MPCS  FIT INFO                  Expected command stages: 'success'=True (expect=True)
2019-039T01:15:35.008  FSWSR  0000002993.09743  MPCS  FIT INFO
2019-039T01:15:35.008  FSWSR  0000002993.09743  MPCS  FIT INFO  MTAK_PASS
2019-039T01:15:35.210  FSWSR  0000002993.09743  MPCS  LOG INFO  Performance     FSW::Heap: Health=GREEN, Used=1177Mb, Max=1484Mb, High Water=1224Mb, Percentage=79%  [1]
2019-039T01:15:35.210  FSWSR  0000002993.09743  MPCS  LOG INFO  Performance     FSW::Overall health of the application is GREEN                    [1]
2019-039T01:15:36.008  FSWSR  0000002993.09743  MPCS  FIT INFO  MTAK_PASS        EHA found/verified XACT_CMD_ACCEPT_COUNT (XACT-0275) is DN 33
2019-039T01:15:36.008  FSWSR  0000002993.09743  MPCS  FIT INFO                  wait 5 seconds
```

Component Level Process

Code Analysis and Review

- Static analysis tools
  - Coverity
  - Semmle
  - Power of 10 rules
  - GCC (compiler)
- Code peer review
- GitHub pull request

Component Level Process

Checklist

Component: FSW/Components/FSWImageManager Component Owner: John Component Contributors: Mike, Peter		
	Status	**Notes**
Modeling		
Model generated in MagicDraw with interfaces defined	YES	
Component auto-coded using component autocoder	YES	
Auto-coded component builds successfully for SPHINX platform	YES	
Implementation		
Behaviors, states, commands, telemetry, and events implemented	YES	
Component builds with the topology	YES	
Deployment		
Static analysis of code performed using SCRUB	YES	
Component Unit Tested	YES	
Executes with topology on SPHINX platform without any issues	YES	
Close-Out		
SDD generated	YES	
Component reviewed by peer(s)	YES	Peers: David
All open issues and action items related to the component have been addressed and closed out	YES	

Status Reports

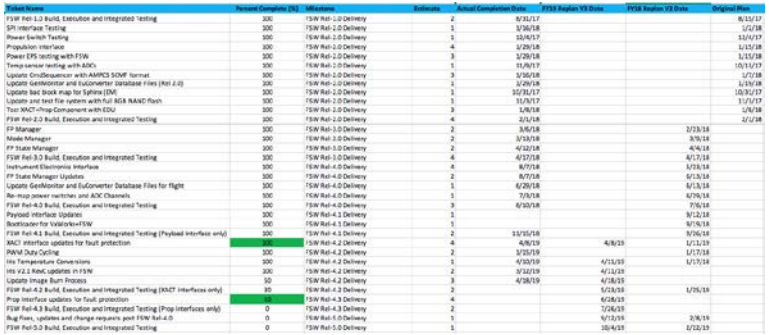
Weekly Progress Summary

- Highlight accomplishments and progress
- Indicate delays in receivables
- Describe pending items
- Estimated upcoming release delivery date
- Describe current progress against development plan schedule
- Communicate problems to stakeholders early on to facilitate timely action

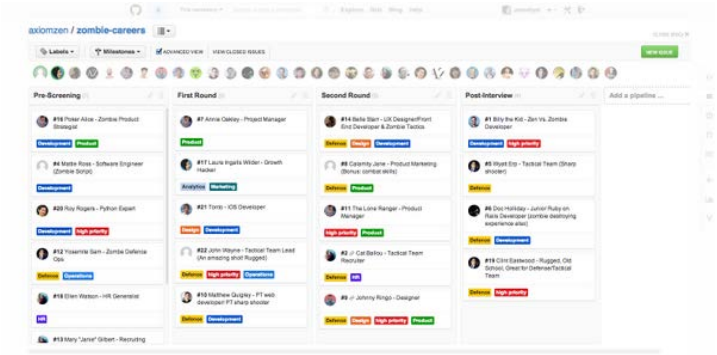
Status Reports

Issue Tracking

- Track current progress to know how we're doing
- Estimate delivery dates and forecast delivery slips early on



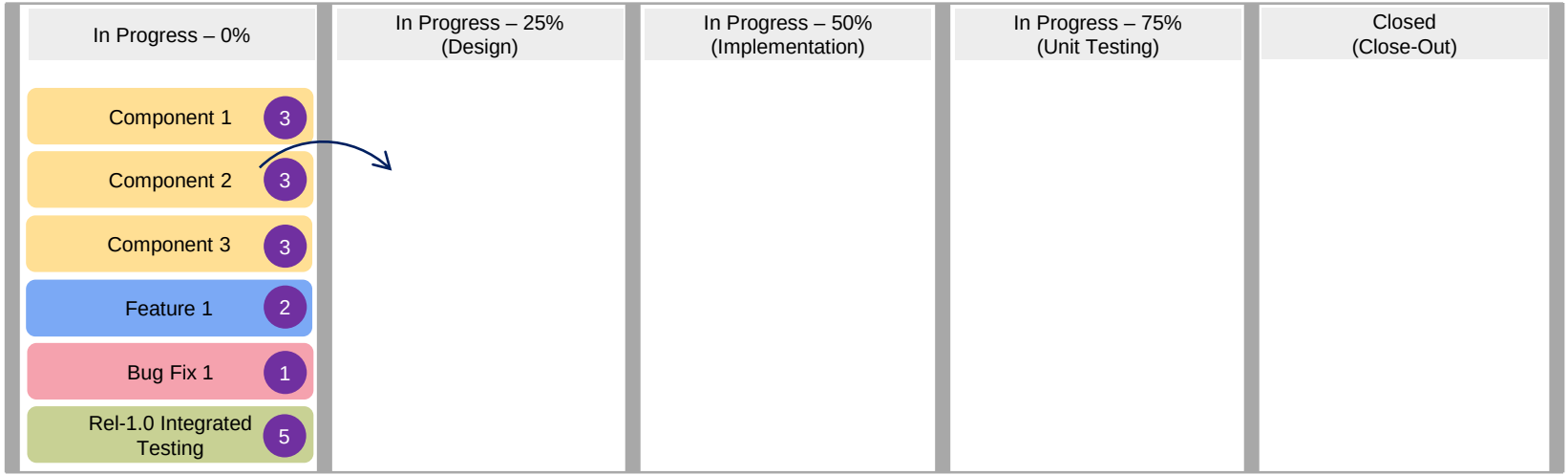
Completion Points



ZenHub Pipelines

Status Reports

Issue Tracking



Total Planned Completion Points for Rel-1.0 = 17

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov