

GNC FLIGHT SOFTWARE DEVELOPMENT AND VERIFICATION

A process, its evolution, and the challenges involved.

Leonardo B. Farconi¹, V. Gabriel Moyano², Tamara G. Rojo², Jörg Brauer³, Christof Efkemann³, Jose L. Redondo Gutierrez¹, Pascal Pieper¹, Erin C. Cold¹, Jan Sommer², Felix C. Passenberg¹, Janosch Reinking¹, Andrea Monti²

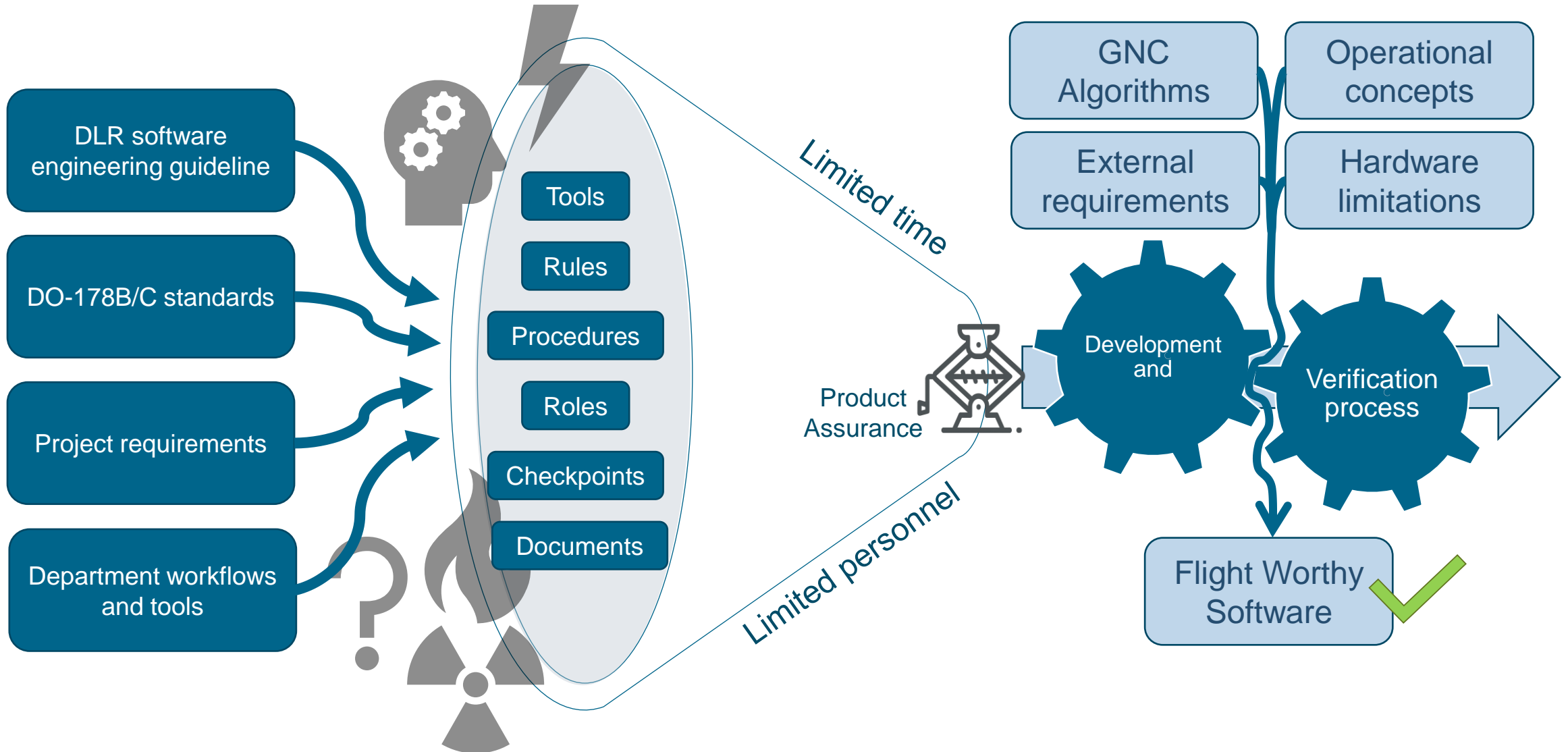
(1) Institute of Space Systems, German Aerospace Center, Bremen;

(2) Institute of Software Technology, German Aerospace Center, Braunschweig;

(3) Verified Systems International GmbH



The problem to be solved



Past and future challenges



.R

Past challenges	Current solution	Current challenge	Possible solutions
Absence of a development or verification process , relying on unwritten rules, personal memory, and personal taste.	Established process with clear roles, actions and frameworks . Virtual “Suggestion box” and recurrent “Idea rounds” allow all participants to propose changes.	Team perceives the process to be complex (compared to no process). Applicability to model-based programming questionable.	Look into standards focused on model-based design (DO-331?)
Too many quality gates would focus too much on design before implementing , or create unnecessary bottlenecks.	Only “review to merge”, with the requirement to keep the architecture and issues updated . Allows prototyping solutions.	Checking architecture and issues design description consistency takes time.	Automated consistency check with use of LLMs?
Architecture specification too complex , causing concentration of knowledge and difficulty in understanding.	Adoption of C4 diagrams , different levels of abstraction, and Clean Architecture concepts.	?	?
No traceability to NEEDs created lack of context and left features not implemented or not tested.	Strong focus on traceability in the specification templates gives direct context to persons involved.	Keeping the traceability between all issues consistent takes time.	?
Too many roles (Product assurance support), created bottlenecks for merging.	DEVs have the highest and most distributed responsibility for PA.	PA responsibility slows DEVs down and reduces interest. Quality checklists are tedious but hard to automate.	Use of LLMs for automated quality checks? Should DEVs be focused on development only and not take care of PA tasks?
Concentration of responsibilities and knowledge on the Lead led to bottlenecks.	Architect role helps disseminate and keep consistency of architecture decisions.	Code modifications either introduce overhead in traceability or are avoided due to the overhead. Participation in architecture choices still reduced.	?
Manual testing consumed too much time, lead to human error, lack of repeatability, or late access to testing in the target.	Automated testing with the target computer with queue for running CI/CD at every MR to stable branch.	Pipeline takes too long and one resource blocks parallel MRs. Choosing which tests to NOT run for every MR is not trivial.	Create parallel resources?

Thema: GNC Flight Software Development and Verification: A process, its evolution, and the challenges involved.

Datum: 09.07.2025

Autor: Leonardo B. Farconi¹, V. Gabriel Moyano², Tamara G. Rojo², Jörg Brauer³, Christof Efke³, Jose L. Redondo Gutierrez¹, Pascal Pieper¹, Erin C. Cold¹, Jan Sommer², Felix C. Passenberg¹, Janosch Reinking¹, Andrea Monti²

Institut: (1) Institute of Space Systems, German Aerospace Center, Bremen; (2) Institute of Software Technology, German Aerospace Center, Braunschweig; (3) Verified Systems International GmbH

Bildcredits: DLR (Created by the author)