Andrew W. **Lyjak**

Software Assurance/Verification/Development · Systems Engineer

46800 Bettyhill, Plymouth MI, 48170

(+1) (734)604-6163 | andrew.lyjak@gmail.com | alyjak | andrew-lyjak | @spacelyjak

"This is the real secret of life – to be completely engaged with what you are doing in the here and now. And instead of calling it work, realize it is play."

Education

University of Michigan — *Ann Arbor, Michigan*

M.E. in Space Systems Engineering — *2009 - 2010*

GPA: 3.769/4.000

University of Michigan — *Ann Arbor, Michigan*

B.E. Aerospace Engineering — *2005 - 2009*

- Minor in German
- GPA: 3.456/4.000

Skills

Programming Languages	Python, Bash, R, Rust, C, C++, Java
Web	Javascript, HTML, (S)CSS, SQL, Django, Webassembly, webpack, Vue
Software	Linux, Android, Ansible, Jenkins, Django, git, Trac, Subversion, JIRA, tokio, Docker, Inkscape
Systems Engineering	Regulation and Certification, Fault Tolerance Analysis, Requirements, Risk, and Change Management
Software Process	Software Development Processes, Agile Development, Software Quality Audits, AS9100, NPR7150
Security Clearance	Inactive U.S. Secret Security Clearance
Languages	English, German, Spanish
Art and Design	Drawing, High School Art Awards, Carving, Woodworking

Experience

Owner — *Ann Arbor, MI*

A Lyjak Cybernetics Consulting LLC — *2023 - Present*

- Consultation services related to Production Lifecycle Management tools and processes for businesses looking to coherently integrate multiple domain-specific workflows (i.e. production, sales, and engineering) into their enterprise application suite.
- Feature development, maintenance, testing for H3D production workflow service.
- Proposal development for Venturi Astrolab's NASA Lunar Vehicle RFP response. Helped design Astrolab's certification and verification strategy for a human-ready lunar vehicle.

Software Release Quality Engineer — *Ann Arbor, MI*

H3D, Inc. — *2020 - 2023*

- Designed, implemented, tested, and released a production workflow service using Docker, Ansible, Django, Webpack. The application uses configuration as code principles in order to re-deploy with a single command.
- Designed, and implemented a client-side distributed network procedure application intended to seamlessly integrate human decision making with configurable automation capabilities. Programmed in Rust.

Senior Systems Engineer II — *Indianapolis, IN*

Raytheon — *2019 - 2020*

- Designed, implemented, tested, and released a continuous integration platform. Architecture relies on on-premise cloud, managed through Ansible, and relies on Jenkins for performing unit-test, documentation, and integration testing of a multi-repository, multi-platform, multi-language hardware+software product.

Software Mission Assurance Engineer — *Hawthorne, CA*

SpaceX — *2010 - 2018*

Software Certification for NASA Safety and Quality Requirements *2010 - 2018*

- Coordinated with SpaceX and NASA engineering and management teams to define and approve the SpaceX Flight Software Development Process. Was the primary author, maintainer, and auditor for the **Flight Software Development Plan**, which describes the process. This document defines SpaceX's agile development process for designing, developing, and verifying flight software.
- Coordinated with SpaceX and NASA engineering teams to document the core flight software architecture in order to demonstrate its fault tolerance and intrinsic safety. Was the primary author and maintainer of the Cargo Dragon Flight Software **Computer Based Control Systems** documentation. This material is used to demonstrate compliance with NASA software safety requirements and also defines many of the software safety test and analysis activities executed for verification of software safety related to the Dragon cargo vehicle.

Independent Verification and Validation (IVV) Contract Management *2010 - 2018*

- Served as the technical point of contact for SpaceX's IVV Contracts for safety critical software, in this role I was responsible for communicating and coordinating exchanges between SpaceX Engineers and the third party assessor for every mission to the International Space Station.
- Managed separate IVV contracts associated with independent assessment of the safety of flight software for Crew Dragon, Cargo Dragon, and the Autonomous Flight Termination System.

Software Process Tool Development *2012 - 2018*

- **branchdiff** - Developed an application to view the differences between two Subversion branches to facilitate merge decisions between them. Displays differences as commits or as the set tickets referenced within those commit messages.
- Developed an application for viewing change over time for various Trac ticket queries.
- Performed trade studies on various software development ticketing systems. The study factored into SpaceX's decision to adopt JIRA across multiple business domains.
- **ReadTheManual** - Installed, modified, and administered an internal fork of ReadTheDocs for use within the SpaceX intranet. This service is used to build and distribute documentation for over 160 internal projects.
- **Tracegraph** - Designed and developed a library for defining systems relationships across information housed within multiple data silos. The library is used for verification tracking purposes to support compliance tracking of SpaceX processes to customer requirements.

Design and Development of the SpaceX Software Standard *2013 - 2018*

- Coordinated with SpaceX and NASA engineering and management teams to define and approve the **SpaceX Software Standard** as a controlling standard for the Commercial Crew Contract. The standard provides a set of requirements to be applied to different classifications of software development, classifications include A, for safety and mission critical software, through D, used for desktop, R&D, or other non-critical applications. The standard is used to evaluate the quality of all software processes related to the Commercial Crew system.

Fault Tolerance Analysis *2016 - 2018*

- Co-developed the **Fault Tolerance Analysis Process**. This process is used to evaluate autonomous and/or operator controlled electromechanical systems. Analysis is used to assess the fault tolerance of a design in order to determine its capabilities and develop fault detection, isolation, and recovery logic for managing redundant capabilities.
- Developed test and analysis plans for verifying fault tolerance as defined through the Fault Tolerance Analysis products.
- With a team of 6 other engineers, executed the Fault Tolerance Analysis Process against 25 separate autonomous control systems within the Crew Dragon Architecture over the course of a year.

Writing

https://worldbuild.ai/W-0000000101/

Author *June 2022*

- Co-authored finalist entry W-0000000101 in the Future of Life Institute's 2022 worldbuilding contest, proposing a realistic but optimistic scenario for life in the year 2045.

https://buildonomy.substack.com/p/the-importance-of-inferring-intention/

Author *August 2023*

- Wrote The Importance of Inferring Intention in which I outline a framework for legibly, reliably, and modularly improving our innate capacity to infer our own intentions as well as the intentions of other agents we interact with.