SpeakUp

A Systems-Engineering Demonstration

Bruce Dombrowski

December 22, 2025

Repository: `https://github.com/brucedombrowski/SpeakUp`

Generated: December 23, 2025 09:55

About This Document

Purpose: This briefing is designed for asynchronous review by managers and customers. It can be read independently without a presenter.

What SpeakUp Is:

- A response to organizational calls for constructive feedback
- A response to customer requests for process improvement recommendations
- A demonstration of systems-engineering discipline applied to knowledge work
- Vendor-neutral at the requirements level—no specific tool is proposed
- Open source (MIT License)—free to use, modify, and distribute

Repository: All artifacts, verification evidence, and this briefing are available at:

`https://github.com/brucedombrowski/SpeakUp`

Problem Statement

The current operating environment has systemic constraints that limit effectiveness:

	Constraint	**Impact**
1.	Fragmented workflows	Disconnected mobile, desktop, and execution environments
2.	Tool accessibility	AI (Artificial Intelligence) unavailable or outside trust boundaries
3.	Inbox-centric work	Critical decisions buried in email threads
4.	Untracked coordination	Limited traceability and auditability
5.	Knowledge attrition	Institutional knowledge lost along with personnel
6.	Budget constraints	Reduced investment in tooling, training, modernization
7.	Legacy systems	Aging infrastructure facing decommissioning
8.	Regulatory burden	Compliance overhead diverts resources from mission work

Governing Principles

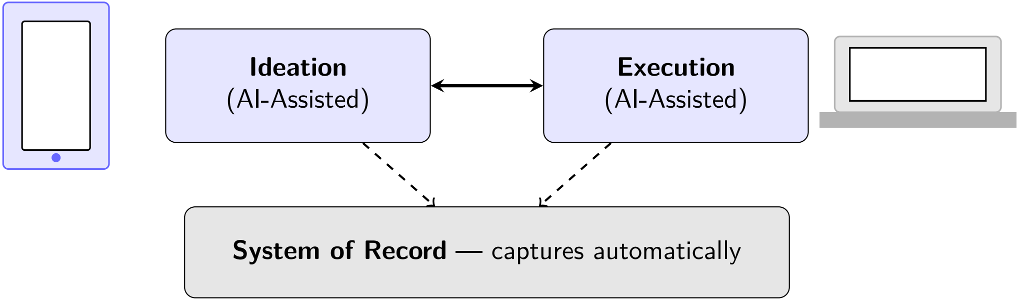
Resources should be organized to maximize utility.

This is achieved through three guiding principles:

Principle	**Application**
Transparency	Work is visible to stakeholders; decisions based on shared understanding
Inspection	Artifacts and progress reviewed frequently; issues identified early
Accountability	Work captured in tracked systems; history preserved for audit

Reference: Adapted from Scrum Guide empirical pillars (Schwaber & Sutherland, 2020)

Proposed Workflow Model



Ideation
(AI-Assisted)

Execution
(AI-Assisted)

System of Record — captures automatically

Ideation

- Frame objectives, constraints, and risks rapidly
- Capture rough drafts and decision notes (mobile or desktop)
- Use text or voice to keep momentum anywhere
- AI assists synthesis and recommends next steps

Execution

- Turn drafts into versioned, reviewable artifacts
- Build inside an IDE with reproducible commands
- AI assists implementation, refactoring, review, and turnover files
- Execute inside managed, trusted boundaries

Functional Requirements (Solution-Agnostic)

These requirements define *what* is needed, not *how* to implement it:

ID	Type	Requirement
FR-1	Mandatory	Mobile ideation capability (smartphone, text/voice input)
FR-2	Mandatory	IDE-centric execution with integrated, replaceable AI assistance
FR-3	Mandatory	Git-based system of record capturing artifacts, history, and rationale
FR-4	Mandatory	Identity and trust boundary alignment (security at identity and device)
FR-5	Recommended	High-signal communication model (email for notification only)

Full requirements with verification traceability are in the repository: `README.md`

Security and Compliance

This project is defensible. You can review it without relaxing any rules.

What This Repository Contains

- No sensitive PII (Personally Identifiable Information)
- No CUI (Controlled Unclassified Information)
- No proprietary information
- No classified information
- Verified by automated scans producing objective evidence

Your Implementation Choices

- AI location (cloud, on-prem, local)
- Repository hosting (GitHub, GitLab, self-hosted)
- Device policies (BYOD—Bring Your Own Device, managed, air-gapped)
- Security policies (your existing rules apply)

The workflow pattern is agnostic. Your security posture is your choice.

Verification evidence: `verification/Compliance-Statement.md`

Value Proposition

The Core Point

With the right environment, one person can do the work of an entire team.

Example: This briefing—IDE, AI agent, LaTeX documents, professional PDFs, version control—all produced by one person. The constraint is not capability. It is environment.

Capability	Current State	Proposed State
Work capture	Fragmented, untracked	Structured, version-controlled
AI assistance	Outside boundary or unavailable	In-boundary, modular
Knowledge preservation	At-risk	Durable artifacts
Automation readiness	Limited	Maximized
Auditability	Manual effort	Built-in traceability

Implementation Approach

This project demonstrates the pattern by being the pattern:

- **Concrete enough to execute**
  - Working repository with all artifacts
  - Defined outputs and verification evidence
  - Reproducible workflow documented in `artifacts/Workflow-Log.md`
- **Abstract enough to remain vendor and environment neutral**
  - Requirements specify *what*, not *how*
  - Implementation choices documented separately
  - Alternative tools and environments can satisfy same requirements
- **Self-demonstrating**
  - This briefing was created using the proposed workflow
  - Ideation on mobile, execution in IDE, artifacts in Git

Repository Contents

All project artifacts are available for review:

File	Purpose
README.md	Authoritative requirements and project specification
briefing/SpeakUp-Briefing.pdf	This document
verification/Compliance-Statement.md	Information handling verification evidence
verification/Requirements-Traceability.md	Requirements to evidence mapping
verification/PII-Scan-Results.md	Automated PII scan test results
verification/scripts/check-pii.sh	Automated verification script
artifacts/Workflow-Log.md	Execution workflow documentation

https://github.com/brucedombrowski/SpeakUp

Handling Constraints and Blockers

When a constraint is encountered, the workflow captures it explicitly:

Example: Export Control Constraint

During BPv7 implementation, NASA TReK was identified as ideal tooling—but is export-controlled (EAR ECCN 9D515.B.5). Rather than stop, the constraint was documented and alternatives evaluated.

Option	Trade-off	Decision
NASA TReK	Export controlled (EAR)	Future path
NASA JPL ION-DTN	Open source	Selected default
Clean-room Python	Full control	Learning/customization

Key Principle: Constraints become traceable decisions, not invisible blockers.
Artifact: `artifacts/Decision-Log.md`, `src/bpv7/simulation/EXPORT_CONTROL.md`

Requirements Anti-Patterns

Bad specifications cost billions and cause project failures:

Failure	**Cost**	**Root Cause**
Mars Climate Orbiter	$320M	Metric/imperial unit mismatch
FBI Virtual Case File	$170M	Vague requirements, scope creep
Ariane 5 Rocket	$370M	Reused incompatible code

Anti-Pattern (FAR 11.104: “least acceptable”)	**Better: Performance Spec**
“Shall use Microsoft Word”	“Shall produce PDF/A archival format”
“Shall be written in Java”	“Shall execute on target platform”
“Shall use Oracle database”	“Shall persist data with ACID guarantees”

SpeakUp approach: LaTeX source (diffable, traceable) → PDF output (portable, archival)

Verification Summary

This project produces verification evidence as first-class artifacts:

Method	Application	NIST* Control
Manual Inspection	Document review	—
PII Pattern Scan	Phone, SSN, IP detection	SI-12
Malware Scan	ClamAV detection	SI-3
Secrets Scan	API keys, credentials	SA-11
MAC Address Scan	Hardware identifiers	SC-8
Host Security	OS configuration	CM-6
File Integrity	SHA-256 hashes	SI-7

*NIST: National Institute of Standards and Technology (SP 800-53 Rev 5)

Security Attestation: All automated scans **PASS**

Policy: Only passing results are published. Vulnerability details are never exposed.

Standards Alignment

Alignment status (clause-level mapping in work):

Standard	Applicability	Status
ISO/IEC/IEEE 15288	Lifecycle process structure, verification/validation cadence	Aligned
ISO/IEC/IEEE 29148	Requirements quality (shall/should/may, traceability)	Aligned
ISO/IEC/IEEE 12207	Software lifecycle for BPv7 implementation/tests	Aligned
ISO/IEC/IEEE 42010	Architecture rationale and viewpoints (workflow model)	In work
ISO/IEC 25010	Product quality characteristics (security, reliability, maintainability)	In work
ISO/IEC 27001	Security controls baseline; scan attestation supports ISMS evidence	In work

Clause-level mapping to be added to verification artifacts in the next iteration.

Recommendation

Adopt the SpeakUp workflow model as a pattern for:

- Converting thinking into durable, reviewable artifacts
- Preserving institutional knowledge as personnel transition
- Enabling automation and reducing manual audit effort
- Maintaining security and trust boundaries while using AI assistance
- Improving signal-to-noise in organizational communication

This pattern is applicable to:

- Engineering work
- Analytical work
- Knowledge work generally

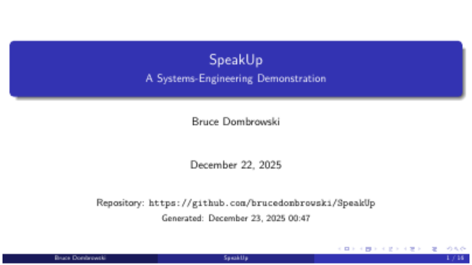
Commit History (Evidence of Work)

15 commits shown (excerpt) — full history is 30 commits over 8 person-hours:

Hash	Commit Message
`9e5aca8`	Update documentation and scan handling
`f1f386e`	Add master build script
`8855611`	Add long-duration DTN test with LOS/AOS
`4090a0a`	Add BPv7 tests and Wireshark demo
`f5d1959`	Update TCPCL to RFC 9174 wire format
`1f3cb42`	Add constraints slide
`cd86850`	Add decision log
`6015094`	Add Docker DTN simulation
`0aecf3c`	Add TCPCL v4 (RFC 9174)
`318800a`	Add BPv7 core (RFC 9171)
`acc6be5`	Add security verification suite
`2b27e68`	Strengthen value proposition
`35a13c7`	Add lifecycle management
`79ac56e`	Add LaTeX briefing deck
`76bd5bc`	Initial commit

Iteration Example: Theme Selection

Four theme variants produced via rapid iteration:



SpeakUp
A Systems-Engineering Demonstration

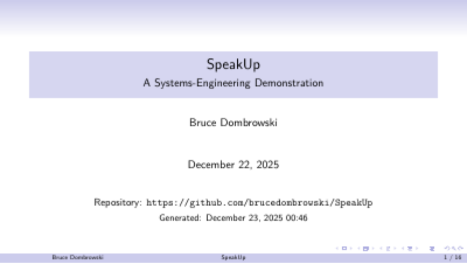
Bruce Dombrowski

December 22, 2025

Repository: `https://github.com/brucedombrowski/SpeakUp`
Generated: December 23, 2025 00:47

Bruce Dombrowski SpeakUp 1 / 16

Madrid (Default)



SpeakUp
A Systems-Engineering Demonstration

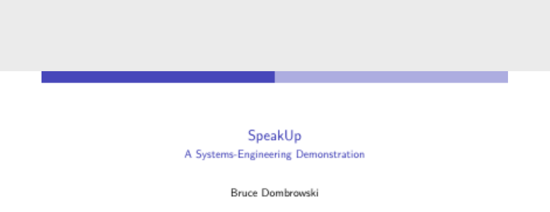
Bruce Dombrowski

December 22, 2025

Repository: `https://github.com/brucedombrowski/SpeakUp`
Generated: December 23, 2025 00:46

Bruce Dombrowski SpeakUp 1 / 16

Singapore (Seahorse)



SpeakUp
A Systems-Engineering Demonstration

Bruce Dombrowski

SpeakUp
A Systems-Engineering Demonstration

Bruce Dombrowski

Next Steps

1. **Review this briefing** and the repository contents
2. **Identify a pilot application area** where the workflow could be applied
3. **Establish repository and workflow** for the pilot
4. **Iterate** between ideation and execution phases
5. **Measure and refine** based on results

This briefing was produced using the SpeakUp workflow model it describes.

Repository: `https://github.com/brucedombrowski/SpeakUp`

Contact: Bruce Dombrowski (Creator)

References

- Schwaber, K. & Sutherland, J. (2020). *The Scrum Guide*. `https://scrumguides.org`
- NIST SP 800-53 Rev 5. *Security and Privacy Controls for Information Systems*
- NIST SP 800-171. *Protecting Controlled Unclassified Information*
- FAR 11.104. *Use of Brand Name or Equal Purchase Descriptions*
- RFC 9171. *Bundle Protocol Version 7* (CCSDS 734.20-O-1)
- RFC 9174. *Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4*
- RFC 8949. *Concise Binary Object Representation (CBOR)*
- ISO 32000-2:2020. *PDF/A Archival Format*
- Git SCM. *Git Reference Manual (git-scm.com/docs)*