INCOSE Michigan Chapter

Monthly Newsletter

August 2026

In This Issue

President's Message

July was an interesting month; the *Salute to America* at Greenfield Village was canceled due to a passing storm that soaked the area and winds that destroyed the stage! It made me remember that the environment always needs to be considered as part of any system's context.

I did visit the Freedom Plane documents at The Henry Ford Museum; the exhibit I enjoyed the most was one I do not recall seeing before: a copy of the Treaty of Paris signed by John Adams, John Jay, and Benjamin Franklin.

My wife and I traveled to New York City to visit our daughter; we visited the Theodore Roosevelt Birthplace, saw *Titanique* and *Hamilton* on Broadway, and visited a variety of famous graves, including Alexander Hamilton's. It is the first time we had experienced the Big Apple…and it was nice to experience the history and overall vibrance of the city.

August and September are going to be full of great systems engineering events; I hope you will join us for the SysModeler.ai presentation on August 11 and our next in-person Chapter event, visiting the MRover team in Ann Arbor (followed by some great BBQ at Relish).

We were sorry to learn that the Western States Regional Conference has been canceled.

This month's Historical Corner is another double feature: The Manhattan Project and the IBM PC.

Michael J. Vinarcik, ESEP-Acq, P.E., FESD
President

Salute to America stage post-storm

Painting of Theodore Roosevelt Sr.
at the Theodore Roosevelt Birthplace

Alexander Hamilton's grave,
Trinity Church Cemetery

Chapter News

We will be adding a few pages to the public-facing INCOSE Michigan website…stay tuned for more details.

Upcoming Events Calendar

Date	Event	Organization
August 6	Skirmish, Siege, and Surrender: Detroit in 1812	Detroit Historical Society
August 11	AI-Powered SysML V2.0 with SysModeler.ai	INCOSE Michigan Chapter
August 11	Test Like You Fight (TLYFi)	INCOSE Los Angeles Chapter
August 19	Webinar 203: SysFly — Flight Simulator with SysML	INCOSE
August 25	SE Lab Demo Day: Modeling as Text — A Practical Guide to SysML v2 with SysIDE	INCOSE
August 27	Engineering Expedition: Fuel for Thought: Route 66 at 100	Stahls Motors and Music Experience
September 2	Webinar 204: Agility as Resilient Emergent Behavior	INCOSE
September 15	MRover (University Rover Challenge Team), Ann Arbor + Systems Soirée (Relish Restaurant)	INCOSE Michigan Chapter
September 16	Webinar 205: Product Line Engineering in Aerospace & Defense	INCOSE
September 16	Systems Engineering of the James Webb Space Telescope	INCOSE Enchantment Chapter
October TBD	Virtual Meeting (TBD)	INCOSE Michigan Chapter
October 3	Sidequest Social: An Idiot's Guide to Wine	INCOSE Michigan Chapter
October 21-24	PMI Global Summit 2026 "Built to Move"	PMI
November TBD	Virtual Meeting (TBD)	INCOSE Michigan Chapter
December TBD	F2F Technical / Social Event	INCOSE Michigan Chapter

Upcoming Chapter Events

AI-Powered SysML V2.0 with SysModeler.ai

Date: Tuesday, August 11, 2026, 6:00 PM - 8:00 PM

Location: Virtual

Cost: Free

Join us for an in-depth, two-hour presentation and live demonstration of AI-powered SysML v2 modeling with SysModeler.ai. This session is open to all INCOSE chapter members, students, and guests looking to expand their understanding of the modern Model-Based Systems Engineering (MBSE) tool landscape and experience next-generation, AI-driven modeling capabilities in practice.

About the Presenter

Md Tawhid Bin Waez, PhD, is the Founder and CEO of SysModeler, Inc., an AI-native platform designed to automate Model-Based Systems Engineering (MBSE) and accelerate digital engineering. With over 20 years of experience in the design, modeling, and verification of safety-critical systems, he previously spent nine years at Ford leading both AV and Software Formal Methods. During his tenure, he spearheaded industry-first formally verified collision avoidance systems and company-wide ISO 26262 software V&V processes. Dr. Waez holds a PhD in Computer Science from Queen's University. He has authored 20 peer-reviewed works, including three patents in safe autonomous systems, and has driven collaborative engineering initiatives with top-tier institutions including MIT, GM R&D, and the University of Michigan.

Register at: https://lp.constantcontactpages.com/ev/reg/z429v6w/lp/c9222244-b3ef-4cfd-bd20-e6ac7d5cda5f

Engineering Expedition:
Stahls Automotive Foundation: Fuel for Thought — Route 66 at 100

Date: Thursday, August 27, 2026, 5:00–8:00 PM (doors 5:00, talk 6:00)

Location: Stahls Motors & Music Experience, 56516 N Bay Drive, Chesterfield, MI 48051

Cost: Free (RSVP recommended)

This free evening talk marks the 100th anniversary of Route 66, the "Mother Road" that ran from Chicago to Santa Monica. For a chapter of systems engineers, the highway is worth a look as a large-scale infrastructure system: a continent-spanning corridor whose value came not from any single segment but from the emergent network of diners, motels, service stations, and roadside attractions that grew up around it — an early example of a transportation backbone shaping, and being shaped by, the economy and culture along its length. The presentation traces how that ecosystem formed, how the route defined American road-trip travel for a generation, and how it faded as the interstate system re-architected long-haul mobility around speed and throughput. Held amid Stahls' collection of vintage automobiles and mechanical music machines, it pairs well with a walk through the exhibits. Seating is first-come to all RSVP guests; walk-ins are welcome to open seats at 6:00 PM.

RSVP / Information: https://www.stahlsauto.com/event/fuel-for-thought-route-66/

Celebrating Systems Engineering Excellence with the University of Michigan MRover Team

Date: Tuesday, September 15, 2026, 6:00–8:00 PM

Location: University of Michigan / Relish Restaurant

Cost: $25 ($10 student) (Dinner included)

Join INCOSE Michigan for a special September dinner meeting as we visit the Michigan Mars Rover Team (MRover) on the University of Michigan's Campus and present the INCOSE Foundation's Excellence in Systems Engineering Award trophy.

MRover is a student-run, multidisciplinary team that designs, builds, and tests a complete Mars rover each year to compete in the University Rover Challenge (URC). At the 2026 URC finals—concluding May 30, 2026 at the Mars Desert Research Station in Utah, where the top 38 finalists from a field of 116 teams across 11 countries competed—MRover placed fifth overall, earned the highest score in the Autonomous Navigation mission, and won the first-ever INCOSE Foundation Excellence in Systems Engineering Award, taking first place in the competition's Systems Engineering category with their rover, Phil.

The award is a fitting recognition of real systems engineering in action. Fielding a rover that navigates rough terrain, manipulates objects with a five-degree-of-freedom robotic arm, performs on-board science and sampling, and executes autonomous missions demands exactly the disciplines our members practice daily—requirements management, interface control, subsystem integration, verification, and risk management across mechanical, electrical, software, and science teams. This is systems engineering made tangible, by students.

We will tour the team's workspace and see the competition rover up close, present the INCOSE Foundation trophy, and then continue the conversation over dinner at nearby Relish Restaurant. It is a great opportunity to celebrate the next generation of systems engineers, talk shop with the team, and connect with fellow members.

Register at: https://lp.constantcontactpages.com/ev/reg/m6qvdnj

Sidequest Social: An Idiot's Guide to Wine

Date: Saturday, October 3, 2026, 6:00 PM

Location: Epiphany 313, 10103 Kercheval Avenue, Detroit, MI 48214

Cost: Varies (General Admission and Front Row tickets available)

Join your fellow chapter members for an evening that pairs hilarious stand-up comedy with a curated tasting of six wine varietals from around the world. The Australian smash-hit *An Idiot's Guide to Wine* has arrived in Detroit, delivering a rare blend of laughter and learning—whether you're a seasoned connoisseur or a complete beginner.

Why This Event?

The show was written by Merrick Watts, the only comedian in the world to hold a WSET (Wine & Spirit Education Trust) Level 3 qualification—a reminder that domain expertise and audience engagement aren't mutually exclusive. Systems engineers will appreciate the underlying challenge: distilling a complex, technical subject (wine) into an accessible, repeatable experience that lands consistently for a diverse audience. It's requirements decomposition and stakeholder communication, served by the glass.

This is a relaxed, social evening at the same Kercheval Avenue venue that hosted our Dungeon Cocktail Experience earlier this year—an ideal setting for members and guests to mingle.

Details:

- Age: 21+ with valid ID
- Language: English, performed by a local comedian
- Accessibility: Venue is ADA accessible
- Note: Alcohol consumption is voluntary; non-alcoholic options available for an additional fee
- Tickets: Reserve your own at https://feverup.com/m/667936

RSVP: Contact michael.vinarcik@incose.net if you plan to attend (we may coordinate dinner reservations if there is sufficient interest).

Partner Organization Events

PMI Great Lakes Chapter

PMI Global Summit 2026 Comes to Detroit — "Built to Move"

Date: October 21-24, 2026

Location: Huntington Place Convention Center

Cost: Varies

PMI's flagship annual conference, the Global Summit, lands in Detroit this October, putting a premier project-management event right in our members' backyard. This year's theme, "Built to Move," centers on navigating accelerating change—choosing agility over rigidity, shaping what comes next, and equipping professionals to stay relevant as the pace of work increases. For INCOSE Michigan members, it's a strong cross-discipline development opportunity under the INCOSE–PMI MOU, and a rare chance to attend a global-scale program without travel. The Summit also runs alongside PMI's Leadership Institute Meeting (LIM), which focuses on leadership training, PMI strategy, and collaborative breakout sessions for chapter leaders.

Register:
https://events.pmi.org/flow/pmi/gsdetroit26/landing/page/home?utm_source=Chapter+Communications&utm_medium=Link&utm_campaign=GlobalSummit2026

Other INCOSE Chapter Events

INCOSE Los Angeles Chapter : Test Like You Fight (TLYFi)

Date: August 11, 2026, 5:30 PM PT / 8:30 PM ET·

Format: Virtual (chapter meeting, webcast)

Cost: Free

Sharpen how you think about verification under real-world threat conditions. This session introduces the "Test Like You Fight" process—an evolution of Test Like You Fly—to evaluate the resiliency of integrated enterprises against new and evolving threats and validate system and enterprise readiness. The resilience and enterprise-readiness themes map directly to SE verification and validation work.

Register: https://lp.constantcontactpages.com/ev/reg/nabs7ay/lp/d4deaa8a-6cff-4f7a-b11e-e9c0b3f8cbf5

Webinar 203: SysFly — Flight Simulator with SysML

Date: August 19, 2026, 11:00 AM EDT

Format: Virtual

Cost: Free

A hands-on look at connecting models to test environments. The session presents a reusable, model-based integration framework using the Functional Mock-up Interface (FMI) standard to bridge SE models and high-fidelity verification environments—integrating FlightGear, SysML v1 simulation, FMI/FMU, Dymola, and hardware-in-the-loop. Strong value for members working in MBSE, digital engineering, and model-based verification.

Register: https://www.incose.org/event/webinar-203-flight-simulator-with-sysml-sysfly-integrating-flightgear-sysml-v1-simulation-fmi-fmu-dymola-and-hardware-in-the-loop/

SE Lab Demo Day: Modeling as Text — A Practical Guide to SysML v2 with SysIDE

Date: Tuesday, August 25, 2026, 11:00 AM – 12:00 PM EDT

Format: Virtual

Cost: Free

A practical look at what changes when the model becomes text. For decades systems engineering models lived in proprietary binary formats that were expensive to edit, hard to diff, and impossible to review line by line; SysML v2 makes the model a standardized, rigorous textual artifact that lives comfortably in a Git repository. Using SysIDE as the working example, the session tours the textual representation and what it means for collaboration, the Automator Python API for programmatic transformations, the Sysand package ecosystem for reuse, and SysIDE Cloud for browser-based access — plus where AI assistants fit in a System-as-Code workflow. A satellite-program case study shows the model driving generated diagnostics software, a monitoring portal, and a telemetry pipeline, and the talk closes with adoption paths for individuals, teams, and educators, including a free editor and 25 free lessons. The technical level is light and the session is aimed at systems engineers, software engineers, architects, and anyone curious about modern MBSE tooling.

For our members, this is a natural follow-on to the chapter's SysML v2 programming this year — the April bootcamp, the June System-as-Code Fest, and our June SysIDE presentation — and a useful counterpoint to the AI-assisted modeling we will see from SysModeler.ai on August 11. Configuration management, peer review, and reuse are perennial MBSE pain points; treating the model as text puts all three back into the toolchain engineers already use for software.

Register: https://www.incose.org/event/se-lab-demo-day-modeling-as-text-a-practical-guide-to-sysml-v2-with-syside/

Webinar 204: Agility as Resilient Emergent Behavior

Date: September 2, 2026, 11:00 AM – 12:00 PM EDT

Format: Virtual

Cost: Free

A framework for building agility into how systems and teams operate. The talk presents eight strategies as fundamental to SE agility—attentive situational awareness, common-mission teaming, shared-knowledge management, iterative incremental development, adaptable modular architectures, attentive decision making, continual integration and test, and being agile. Useful for practitioners and leaders applying agile principles within an SE lifecycle.

Register: https://www.incose.org/event/webinar-204-agility-as-resilient-emergent-behavior/

Webinar 205: Product Line Engineering in Aerospace & Defense

Date: September 16, 2026, 11:00 AM EDT

Format: Virtual

A practitioner-focused session with a notable industry panel. As Aerospace & Defense faces pressure to deliver increasingly complex, configurable products while accelerating innovation and maintaining compliance, this session draws product line engineering lessons from Lockheed Martin, Safran Seats, and PTC. Directly relevant to reuse, configurability, and engineering-process transformation in SE.

Register: https://www.incose.org/event/webinar-205-accelerating-aerospace-defense-transformation-with-product-line-engineering-lessons-from-lockheed-martin-safran-seats-and-ptc/

INCOSE Enchantment Chapter: Systems Engineering of the James Webb Space Telescope

Date: September 16, 2026, 4:45-6:00 PM MT

Format: Virtual

A compelling applied-SE case study on one of the most complex systems ever fielded. The JWST—launched December 25, 2021, as NASA's successor to Hubble—was designed to observe “first light” objects in the nascent universe and the evolution of galaxies. High-interest for members who enjoy large-scale integration and lifecycle SE stories. Registration is by request; contact the chapter via the event page for the meeting invitation.

Register: https://enchantment.incose.org/event/enchantment-chapter-systems-engineering-of-the-james-webb-space-telescope/

Other Events

Detroit Historical Society:
Skirmish, Siege, and Surrender: Detroit in 1812 - quad anniversary lecture series

Date: August 6, 2026, 6:30 – 7:45 PM (Detroit Historical Society)

Cost: Free

From the DHS event description: Presented by Eden Mitchell, DHS Wayne County History Project Research Associate, this talk examines Detroit’s unique distinction as the only American city to have ever been surrendered to a foreign invading power. It explores Detroit’s critical role in the opening months of the War of 1812, highlighting its strategic importance to both the United States and Great Britain as they vied for control of the Great Lakes frontier.

The presentation also traces the events leading to the city’s surrender in August of 1812, situating them within the broader military and political pressures of the period. William Hull and Isaac Brock, the respective commanders of the American and British forces, will be discussed and compared in terms of their leadership, strengths, available resources, and decision-making, as well as how effectively each was able to use those factors to their advantage—or how they ultimately contributed to their setbacks.

Registration: https://www.detroithistorical.org/events/skirmish-siege-and-surrender-detroit-1812-quad-anniversary-lecture-series

Historical Corner

Two August anniversaries, thirty-six years apart, bookend a central question in systems engineering: when should the hard work of experimentation stay inside a single integrating organization, and when should it be pushed out to a market? In August 1945, the Manhattan Project delivered a weapon through the most tightly integrated, centrally directed engineering program the nation had ever attempted. In August 1981, IBM introduced a personal computer built on the opposite philosophy — open, modular, and deliberately designed for others to extend. Both programs used parallel experimentation to manage risk. They differed in who was permitted to run the experiments, and in who captured the value that resulted.

This Month in SE History: The Manhattan Project — Integration at the Limit

August 2026 — 81 Years Since Hiroshima and Nagasaki

On August 6 and August 9, 1945, atomic bombs were dropped on Hiroshima and Nagasaki, concluding a three-year effort that remains one of the largest and most consequential engineering programs in history. At its peak the Manhattan Project employed roughly 130,000 people and cost on the order of $2 billion — the equivalent of tens of billions of dollars today. What makes it a systems engineering story is not the physics, remarkable as that was, but the integration problem: converting an unproven laboratory phenomenon into an industrial enterprise, on a compressed wartime schedule, under profound uncertainty about whether any of it would work at all.

The Problem: A Science Project That Had to Become an Industry

Nuclear fission was discovered only in the late 1930s. Knowing that fission was possible, however, was a world away from knowing one could build a weapon — and further still from knowing one could manufacture the fissile material at scale. The program had to advance the underlying science and commit to industrial-scale production essentially in parallel, before the science was settled. Enrico Fermi's first controlled chain reaction, achieved in Chicago on December 2, 1942, proved the principle; the Manhattan Engineer District had already been established under the U.S. Army Corps of Engineers that August. Production plants were being designed while the processes they would house were still being invented.

Parallel Paths as Risk Reduction

No one knew which method of producing fissile material would succeed, so the program pursued several at once. Uranium-235 was separated primarily by electromagnetic means and by gaseous diffusion at Oak Ridge, Tennessee, while plutonium-239 was bred in reactors at Hanford, Washington. More than eighty percent of the total project cost went into building and operating these fissile-material plants. This was deliberate, expensive redundancy — buying schedule certainty by running several independent technical bets simultaneously. The two weapon designs followed the same logic: a gun-type uranium device ("Little Boy") simple enough to be fielded without a full-system test, and an implosion plutonium device ("Fat Man") so novel it required the Trinity test of July 16, 1945 to validate before use.

Integration Under a Single Authority

Then-Brigadier General Leslie Groves of the Army Corps of Engineers directed the overall program; J. Robert Oppenheimer directed weapon design at Los Alamos. Three geographically separated principal sites — Los Alamos for design, Oak Ridge for enrichment, Hanford for plutonium — had to be coordinated with major industrial contractors and university laboratories across the country. The integrating authority was singular and strong. Organizational interfaces, and the decision to hand reactor construction and operation to industry, were managed from the center. It is difficult to imagine the schedule surviving a more diffuse structure.

SE Lessons for Practitioners

- **Parallel experimentation buys schedule at the cost of resources.** When you cannot know in advance which technical approach will succeed and the schedule cannot wait for you to find out, running several in parallel is a rational hedge — but an expensive one. The Manhattan Project could afford it because its stakes and budget were effectively unbounded. Most programs cannot, which is precisely why the discipline of timely down-selection matters.
- **Concurrency compresses schedule and concentrates risk.** Production plants were designed and built before the processes they housed were proven. This concurrency likely saved years, but it committed enormous resources to approaches that might have failed. Concurrent engineering is a schedule tool with a risk bill attached, and the bill comes due when an unproven assumption turns out wrong.
- **Know which decisions can be frozen and which must be validated.** The gun-type and implosion devices embody two answers to the question "how confident must we be before we commit?" The uranium gun was simple enough to freeze and field untested; the plutonium implosion device was not, and Trinity was the price of that confidence. Distinguishing decisions that can be frozen early from those that require validation before commitment is a core integration judgment.
- **A large program needs a single integrating authority.** Groves' role — coordinating the Army, industrial contractors, and scientific laboratories across a continent — is the integrator function writ large. Authority spread thinly across that many interfaces would have been fatal to the schedule.
- **Compartmentalization protects secrets and starves integration.** Security compartmentalization deliberately limited who knew what. It protected the program, but it also impeded the horizontal information flow that integration depends upon. The tension between need-to-know and need-to-integrate is one that engineers on sensitive programs still navigate today.

No account of the Manhattan Project is complete without acknowledging its cost: well over one hundred thousand people died as a direct result of the two bombings, and the program opened enduring questions about the engineer's responsibility for the uses of what is built. For systems engineers it stands as both a demonstration of what integrated, well-directed engineering can accomplish and a reminder that accountability for a system's effects does not end at its technical boundary.

Sidebar: Parallel Programs, Then and Now

The Manhattan Project's willingness to fund several competing approaches and keep only the winner did not end in 1945. Through the 1950s the United States ran multiple ballistic-missile programs concurrently: the Atlas and Titan intercontinental missiles were developed in parallel, soon joined by the solid-fuel Minuteman, while the Air Force's Thor and the Army-led Jupiter were pursued as deliberately competing intermediate-range designs. The redundancy was intentional — with the stakes of the early Cold War, the aim was to guarantee at least one fielded success rather than to bet everything on a single design.

Equally important was what happened to the programs that lagged. In 1957 the Defense Department reoriented the effort: Atlas kept top priority, Titan was demoted to second, and Thor and Jupiter were referred to a joint committee. Both were briefly fielded — several squadrons each — but neither endured; Jupiter and Thor alike were retired within a few years of deployment as more capable systems came online. Underperforming programs were curtailed, not sustained indefinitely.

In *Mobilize: How to Reboot the American Industrial Base and Stop World War III* (2026), Palantir chief technology officer Shyam Sankar and co-author Madeline Hart argue that this combination — funding genuine competition and then ruthlessly cancelling what is not working — is precisely what modern defense acquisition has lost, with programs now tending to persist regardless of performance. The authors write from an explicit reform agenda, and readers will weigh that case on its merits. But the underlying systems-engineering point stands independent of the policy debate: parallel experimentation delivers its risk-reduction value only when it is paired with the discipline to terminate the losers. Run several bets at once and you buy insurance against any one failing; refuse to close the losing bets and you simply pay for all of them. The Manhattan Project and the early missile age captured the upside because they were willing to stop.

References

- “Manhattan Project,” Encyclopædia Britannica: https://www.britannica.com/event/Manhattan-Project
- Manhattan Project National Historical Park, U.S. National Park Service: https://www.nps.gov/mapr/index.htm
- “An Engineering History of the Manhattan Project,” Construction Physics: https://www.construction-physics.com/p/an-engineering-history-of-the-manhattan
- Sankar, S., and Hart, M., Mobilize: How to Reboot the American Industrial Base and Stop World War III, Bombardier Books, 2026.

This article was drafted with assistance from Claude.ai by Anthropic.

This Month in SE History: The IBM PC — When Modularity Unlocks Value and Surrenders Control

August 2026 — 45 Years Since the IBM 5150

On August 12, 1981, IBM introduced the Model 5150 Personal Computer at a New York hotel. Priced at about $1,565, built around an Intel 8088 processor and running Microsoft’s PC-DOS, it was developed in roughly a year by a small team in Boca Raton, Florida, under the codename “Project Chess.” Its specifications were modest even for 1981. Its architecture was not. The IBM PC is the textbook case study in the economic and engineering power of modularity — and in the control an integrator gives up to unlock that power.

A Modular Architecture, Openly Published

IBM did something unusual for the era: it built the machine from off-the-shelf components and published a technical reference documenting the circuit designs and the BIOS source code, so that other firms could build compatible hardware and software. In the language of Carliss Baldwin and Kim Clark’s *Design Rules: The Power of Modularity*, IBM had defined a complete set of design rules. Baldwin and Clark hold that a complete set of design rules must specify three things: architecture — which modules exist and what their roles will be; interfaces — detailed descriptions of how the modules interact; and integration protocols and testing standards — the procedures for assembling the system and judging whether a given module conforms. The published PC reference supplied all three. That was the enabling act.

Why Modularity Multiplies Value

Baldwin and Clark’s central claim is quantitative. Combining a modular structure — splitting a system into modules — with independent, parallel experimentation on those modules — substituting a better module for a lesser one — can increase system value many times over, by their estimate as much as twenty-five-fold. The mechanism is options. An interconnected design offers a single option: take the whole result or leave it. A modular design creates many, because each module becomes an independent unit of selection, and any module that obeys the rules can be swapped for a superior one. A portfolio of options, as they put it, is worth more than an option on a portfolio. The PC realized this at once: memory boards, graphics adapters, disk controllers, and eventually entire compatible systems appeared as independent modules, each an arena for competition and improvement, all conforming to IBM's published rules.

The Control IBM Surrendered

Here is the part every integrator contemplating an open architecture should weigh. The same open interfaces that unlocked the value also made the modules substitutable — including IBM's own machine. Because the design rules were public, competitors could build not merely peripherals but complete "IBM-compatible" computers. Baldwin and Clark quantify the consequence starkly: IBM had once captured on the order of 71 percent of the economic value of the computer industry; after the industry modularized around the PC standard, no single firm held more than roughly 15 percent. The value did not vanish — it multiplied, exactly as the theory predicts. But it dispersed to the makers of modules — Intel in processors, Microsoft in operating systems, and a host of firms in cards and clones — rather than accruing to the integrator that had authored the rules. Within little more than a decade, IBM had lost its market leadership in the personal-computer business it created — overtaken by the very compatible-makers its open architecture had enabled.

SE Lessons for Practitioners

- **Design rules are architecture, interfaces, and tests — specify all three.** A modular strategy is only as sound as the rules that govern it. The PC succeeded because IBM defined not just what the modules were, but precisely how they connected and how conformance would be judged. Underspecify the interfaces or the test standards and modularity collapses back into interdependence.
- **Modularity converts uncertainty into options.** A modular system can be worth many times its integral equivalent because it lets many parties run independent experiments and keeps only the winners. Baldwin and Clark note that the value of experimentation rises with technical risk and complexity. Where the design space is large and uncertain, modularity is not merely an engineering convenience — it is a value strategy.
- **Enforce the rules or lose the structure.** Baldwin and Clark warn that design rules must be enforced for the life of the effort, or the system "collapses back into interconnectedness." IBM's rules held — arguably too well. The compatibility standard was so stable that anyone could build to it reliably.
- **Open interfaces disperse value as surely as they create it.** Modularity multiplied the total value of personal computing and simultaneously eroded the dominance of the firm that architected it. An integrator that opens its interfaces should expect the modules — including its own — to become substitutable, and should decide in advance where it intends to capture value once they are. Owning the processor or the operating system, as Intel and Microsoft did, proved far more durable than owning the integration. IBM's loss was not modularity as such but a sourcing choice within it: by licensing the processor from Intel and the operating system from Microsoft rather than controlling either, it authored the rules while ceding the two modules where value proved most durable. Modularity disperses value; which interfaces the integrator retains determines where that value lands.
- **Choose your chokepoints deliberately.** Baldwin and Clark describe "hidden modules" — components with no standalone value that exist only because they obey the design rules. In a modular ecosystem, deciding which modules you will supply, which you will let others supply, and which interface you will control is a strategic architecture decision, not merely a technical one.

The IBM PC is often told as a story about a beige box. For systems engineers it reads better as a controlled experiment in modular design theory — one that confirmed both halves of the theory at once. Modularity delivered the value multiplication Baldwin and Clark describe, and it redistributed that value away from the integrator with equal reliability. Both outcomes were predictable from the architecture. Both are worth recalling the next time an interface is about to be published.

References

- Baldwin, C. Y., and Clark, K. B., Design Rules, Volume 1: The Power of Modularity, MIT Press, 2000 (see pp. 12, 15, 59–77, 133, 236–260, 280–285).
- “The IBM PC,” IBM History: https://www.ibm.com/history/personal-computer
- “IBM Personal Computer,” Wikipedia: https://en.wikipedia.org/wiki/IBM_Personal_Computer

This article was drafted with assistance from Claude.ai by Anthropic.

INCOSE Michigan Chapter

https://www.incose.org/michigan | Follow us on LinkedIn

Fostering a world-class systems engineering environment

Portions of this newsletter were researched and drafted with assistance from Claude.ai (Anthropic, 2026).