White Paper
August 2023

IT@Intel: Reliability Engineering Helps Intel Cut IT Manufacturing Systems Downtime in Half

Investing in a new Reliability Engineer role, tasked with increasing the resilience of its manufacturing IT systems, helped Intel IT reduce unscheduled downtime in its factories by 50% from 2019 levels

Authors

Patrick Ennis
Site Reliability Engineer, Manufacturing IT

Andrew Brown
Site Reliability Engineer, Manufacturing IT

Michael Herring
Principal Engineer – Systems & Architecture, Technology Development

Aidan O'Connor
Strategic Planner, Manufacturing IT

Robert O'Connor
Site Reliability Engineer, Manufacturing IT

Joe Sartini
Global Domain Lead for Industry 4.0

Table of Contents

Executive Summary 1
Background 2
Solution 3
Roles and Responsibilities 3
Architecture and Integration 4
FMEA and the RMM 4
CI/CD and Automation 5
Observability, Monitoring and AIOps 6
Incident Response and Continuous Learning 6
Results 6
Conclusion 7
Related Content 7

Executive Summary

Driven by the rising importance of keeping manufacturing sites operating at full capacity 24/7, Intel Manufacturing IT (MIT) has set a goal of achieving "four nines" (99.99%) availability (or 0.01% downtime). To help achieve this ambitious goal, we added a Reliability Engineer role to enhance the resilience of Intel's manufacturing facilities. Reliability engineering (RE), first developed by cloud-based digital service providers, focuses on designing systems to be failure-tolerant, so that service is maintained even when individual components fail.

Our Reliability Engineers proactively tackle potential vulnerabilities and develop strategies to mitigate the impact of failures on manufacturing operations. They play a critical role in identifying common failure modes, developing standards and designing solutions to lower the risk of failure.

- Reliability Engineers' use of the Failure Mode and Effects Analysis (FMEA) methodology enabled us to develop a Resiliency Maturity Model (RMM), which is applicable across all our systems.
- This approach has helped us to identify over 200 resilience improvement projects and add them to our development roadmap for the next two years.
- Through these RE initiatives and implementation of numerous operational improvement activities, unscheduled factory downtime has decreased by 50% from 2019 levels.

These results demonstrate how RE can be applied beyond the cloud-based microservice environments that have been its traditional focus to bring the benefits of resilience to the manufacturing environment.

Acronyms

AIOps	artificial intelligence for IT Operations
CI/CD	continuous integration/continuous delivery
FMEA	Failure Mode and Effects Analysis
IDM	Integrated Device Manufacturing
MIT	Manufacturing IT
NAS	Network Attached Storage
OT	Operational Technology
RE	reliability engineering
RMM	Resiliency Maturity Model

Background

With demand for microprocessors at unprecedented levels, it is imperative for Intel factories to operate at maximum output. But as factories have grown larger and more automated, the technology supporting them has become exponentially more complex. This high degree of manufacturing automation means that any problem with the IT automation systems can immediately impact the entire factory. Given that the cost of downtime in Intel's manufacturing facilities can amount to millions of dollars per hour, ensuring uninterrupted operation is of paramount importance.

For many years, Intel has been dedicated to improving its manufacturing processes. A key initiative involved the merger of Information Technology (IT) with Operational Technology (OT) about 15 years ago. This integration aimed to bring major enhancements in industry IT systems management to our OT systems through improved flexibility, redundancy and recovery techniques. Practices adopted as a result include the Information Technology Infrastructure Library; Agile software development; disaster recovery and business continuity; and establishing a playbook to optimize both system and factory recovery. These and other measures helped Intel to achieve an impressive factory uptime rate of 99.92% in 2019.

Intel's Integrated Device Manufacturing (IDM) 2.0 strategy further increases the negative consequences of downtime. IDM 2.0 is a strategy for expanding Intel's manufacturing capabilities and approach to building products. It combines three components: scaling Intel's global, internal factory network; expanding utilization of third-party foundry capacity; and building a world-class foundry business, Intel Foundry Services.

Under IDM 2.0, any disruptions or outages in Intel's manufacturing processes can have cascading negative impacts to delivery commitments for Intel, affecting Foundry customers' operations and Intel's ability to meet service-level agreements. An overarching goal of scaling factories to double the current capacity means that the velocity of operations must be increased without compromising reliability. Any impact to these scaled-up factories multiplies existing downtime costs, which are already in the millions. This is another reason why IDM 2.0 results in a stronger return on investment for resiliency measures.

It has become evident that new levels of resilience to IT-related failures are necessary to mitigate risks and maintain uninterrupted operations. Resilience refers to the ability to maintain service levels during unplanned failures. Resilient systems are designed to anticipate and tolerate failures to such an extent that failure is considered a normal state.

Our internal analysis (see Figure 1) showed that 50% of factory availability impacts have resiliency as a factor (42% resiliency failures plus 8% resiliency and change failures), while another 43% have change as a factor (35% change failures plus 8% resiliency and change failures). These types of failures suggested a substantial opportunity to reduce impacts by increasing system resilience.

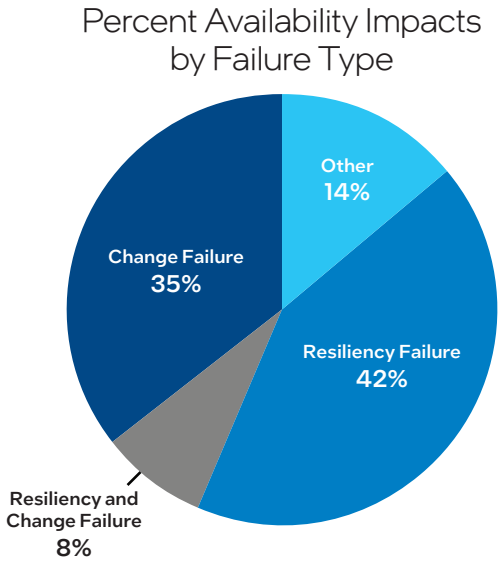

Figure 1. Availability impacts by failure type.

Copy Exactly!—How Intel Delivers Manufacturing Solutions Worldwide in Record Time

Copy Exactly! is a methodology used by Intel to transfer production solutions, updates and improvements from one site to another in order to enhance repeatability, efficiency and reliability across manufacturing facilities. Since all Intel factories are designed using similar hierarchies and equipment, the Copy Exactly! process minimizes the risk of introducing errors and problems into high-volume manufacturing by replicating every detail—including hardware and software components—that might affect the manufacturing process.

To address these challenges, we've set a new goal of achieving "four nines" (99.99%) availability, or 0.01% downtime, which is less than 53 minutes of downtime per

year per factory. To realize this ambitious target, our focus needed to shift to seeking more resilience in our systems, moving beyond firefighting individual issues to foster a more proactive and systemic approach. Our ultimate objective was to create a culture of resilience—a holistic way of helping to ensure that Intel's manufacturing processes could withstand and recover from failures.

Solution

In response to the need for increased resilience, Intel Manufacturing IT (MIT) implemented reliability engineering (RE), a practice that focuses on designing systems to be failure-tolerant, so that service is maintained even when individual components fail. Until now, this approach has primarily been developed and used by cloud-based digital service providers. By adopting the principles and practices of RE, Intel MIT aimed to embed resilience into Intel's microchip manufacturing processes, moving from a traditional emphasis on feature delivery and giving more consideration to improving resiliency in these systems. This approach allows us to build robust and dependable systems that meet the highest standards of performance and customer satisfaction.

At its core, RE entails identifying design patterns that promote continuity of service, both within individual applications and in their interactions. This approach involves collaboration between the RE team and the development team to help ensure that feedback on opportunities to enhance resilience is received and incorporated into system design. By closing the loop with developers, Reliability Engineers help align their overarching goals for resilience with the feature delivery objectives of the development team, enabling us to create robust and reliable solutions that meet the needs of our stakeholders.

While RE shares similarities with DevOps, it also possesses distinct characteristics. DevOps encompasses a culture and set of principles that promote collaboration and integration between development and operations teams. On the other hand, RE typically exists as a distinct role within the organization, with a focus on achieving specific outcomes, such as enabling rapid change and ensuring high availability and performance. RE requires technical expertise in implementing software solutions that encompass architectural and system design patterns for resilience.

A key aspect of RE is building and maintaining a partnership with the Technology Development organization to ensure solutions are implemented consistently and comprehensively, while prioritizing delivery of both resiliency and user features appropriately. The RE focus complements the developers' views, and collectively they deliver resilient solutions and capabilities to end users.

The following sections delve into different elements of Intel MIT's RE solution for manufacturing.

Roles and Responsibilities

To improve resilience across all services, Intel MIT created a new role and recruited a small team of Reliability Engineers. The role involves proactively tackling potential vulnerabilities and developing strategies to mitigate the impact of failures on manufacturing operations (see Table 1). Reliability Engineers play a critical role in identifying common failure modes, developing standards and designing solutions to lower the risk of failure.

One of the ongoing challenges in a fast-paced manufacturing environment is the prioritization of new capabilities over improved availability. Reliability Engineers actively address this challenge by advocating for the importance of resilience alongside new feature delivery. By highlighting the long-term benefits of a resilient infrastructure, they help strike a balance between innovation and operational stability.

Reliability Engineers also serve in a consulting and advisory capacity across different development domains. They transfer knowledge and expertise regarding best practices for resiliency, acting as guides for development teams in adopting and implementing these practices. Through collaboration and knowledge sharing, Reliability Engineers contribute to the continuous improvement of resilience across the organization.

In addition, Reliability Engineers are knowledgeable about IT industry best practices for building resilient systems. They stay updated with the latest advancements and trends in the field, incorporating relevant insights into manufacturing processes to help keep Intel at the forefront of resilient system design and implementation.

Table 1. Responsibilities of Intel MIT Reliability Engineers

Responsibility	Examples
Design integrated, cross-domain manufacturing automation architecture solutions for resiliency	▪ Timeout and retry design ▪ Auto-scaling ▪ Optimal system coupling/modularity ▪ Data path simplification
Design standards for individual system resiliency	▪ Auto-contain ▪ Auto-failover ▪ Purging ▪ Auto-scaling
Design architecture and standards to enable rapid change	▪ Change automation solutions ▪ Optimal system coupling/modularity ▪ Copy Exactly!/configuration management
Develop and champion reliability engineering (RE) methodologies	▪ Topology mapping ▪ Artificial intelligence for IT Operations (AIOps) ▪ Failure Mode and Effects Analysis (FMEA)
Understand and transfer industry best practices	▪ Timeout and retry exponential back-offs ▪ Architecture fitness functions ▪ Circuit breaker/throttling
Consult and coach on resiliency and architectures for change	▪ Design review ▪ Participation in the Developer Integration Focus Team (the forum where RE aligns plans with Automation Development)

Architecture and Integration

Our Reliability Engineers were able to standardize resilient solutions into a set of design patterns, applicable both across individual subsystems and the integrated architecture as a whole. Table 2 shows a non-exhaustive list of these resiliency design patterns.

Table 2. Resiliency Design Patterns

	Pattern
Redundancy	▪ Auto-contain ▪ Auto-failover ▪ Deep real-time health checks ▪ Automated materials-handling system redundancy
Capacity Management	▪ Burst capacity for maintenance ▪ Database purging ▪ Database maintenance ▪ Load balancing ▪ Dynamic capacity/auto-scaling
Preventing Cascading Impacts	▪ Throttling/circuit breakers ▪ Timeout and retry designs ▪ Coupling and modularity ▪ Elimination of artificial software limits ▪ Caching for resiliency

Each of these design patterns requires a comprehensive RE activity to define it for the integrated architecture and as a systems standard.

For example, in our current manufacturing automation architecture, multiple solutions are used to auto-contain servers and components that exhibit degraded performance. Each auto-contain solution has its own strengths and weaknesses. Some mission-critical systems have no validated auto-contain solution at all, despite a product architecture design that shows a high degree of redundancy "on paper" (for instance, multiple servers). Deep real-time health checks are critical to reliable auto-contain solutions; they consistently monitor components for errors, latency, timeouts and queues. RE focuses on closing these gaps and driving standards as appropriate.

Similarly, timeout and retry designs are critical for enabling full redundancy and preventing cascading impacts. Our manufacturing automation architecture does not have a holistic approach to timeout and retries, as it has largely been driven by individual domains. RE defines a holistic and standardized design, leveraging an extensive set of existing software industry best practices.

Cascading impacts are inherent to architectures with a high degree of coupling and many dependencies between subsystems and components. In these architectures, point failures may cascade into multi-system impacts due to resource exhaustion resulting from circular dependencies in data flows between systems. Mitigating this risk involves identifying dependencies and constraints, and optimizing the coupling between automation systems.

As an example of a cascading impact, the Network Attached Storage (NAS) solution used in our manufacturing automation architecture is inherently reliable; however, Multiple User Outage and Full Factory Down events have occurred when access to the NAS was interrupted. Our Reliability Engineers identified the small set of use cases that use NAS for critical transactions and designed out this redundant data path—for example, by using caching for resiliency instead.

FMEA and the Resiliency Maturity Model

Applications that are built on similar technology and architecture may have different levels of protection from a shared failure mode. Our team began to understand exposure to failure by analyzing common failure modes across manufacturing operations, utilizing the Failure Mode and Effects Analysis (FMEA) methodology to anticipate potential issues and failures. FMEA is a Six Sigma qualitative and systematic tool that can anticipate what might go wrong with a system or process. It not only identifies how the system might fail and the effects of each failure type, but also helps assess the probability of occurrence and the likelihood of detection of that failure type. The ability to predict potential issues enables practitioners to design-out failures and design-in reliable solutions.

We used the FMEA methodology to build a list of common failure modes across the full suite of critical automation systems. Analysis showed that failure modes tend to repeat across similar systems. This led to the development of a heat map, which shows failures matrixed against critical systems and highlights systems that are at risk of shared failure modes, with open risks shown in red or orange depending on likeliness of impact.

Using the FMEA methodology and heat map as a foundation, we developed a Resiliency Maturity Model (RMM). The first section of the model (shown in Table 3 on the following page) focuses on assessing the resilience of internal and external dependencies, including critical system components and external factors like third-party services. The second section addresses the resilience elements in our change implementation methods. These are vital to the reduction of the risk that comes with system updates while ensuring we preserve system resilience.

Each common failure mode was scored on a maturity scale of resilience to that failure mode on a scale from 1 to 5, with different resiliency capabilities as the criteria for each level. Mission-critical applications were then assessed to identify possible improvements.

Table 3. Resiliency Maturity Model

Resiliency Failure Mode *Resiliency Capability*	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Active/active app, middleware server or component is unresponsive or shows degraded response or high error count *Impact-free auto-contain capability*	No auto-contain capability.	...	...	...	Auto-contain running externally is impact-free.
Active/passive app or server is unresponsive or shows degraded response or high error count *Impact-free active/passive auto failover*	No auto-failover capability.	...	...	...	No active/passive component within the system architecture.
Database purger/archiving failures leading to performance impact *Impact-free purging capability*	No data purging capability.	...	...	...	Purging capability exists for all required objects with logging, monitoring and observability. Purging self-throttles to keep database performance optimal.
Connectivity failure to external dependency[a] *Application is resilient to connectivity failures (such as timeout and retries, caching, etc.)*	Application immediately impacts factory.	...	...	...	Application is not externally dependent or is automatically resilient to long outages (>4 hours).
Connectivity failure to internal dependency[b] *Application is resilient to connectivity failures (e.g., will auto-reconnect post-database failover)*	Application requires manual recycle to reset connection. No monitoring.	...	...	...	Full auto reconnect/recycle occurs—impact is avoided. No time impact during internal dependency failure (e.g., ability to use a caching facility while database is down).
Capacity or redundancy constraint *System has sufficient burst capacity (including app and database nodes)*	Capacity based on forecasted steady-state peak needs.	...	...	...	N+1 capacity exists at each Data Center and dynamic auto-scaling.
Change Failure Mode *Change Capability*	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Human error failures *Change is automated to avoid human error-induced change failure*	Changes are implemented with manual steps.	...	...	...	Changes are fully and centrally automated with standard automation tools—little human "glue" involved in change.
"Too big to fail" *Change has reduced scope of change and can be backed out quickly*	Iterations occur irregularly and include many major revisions spliced together or bundled with other feature releases.	...	...	...	Iterations occur weekly with deployment pipeline tracking and order of each iteration. The ability to canary test at a granular level. Systems use immutable infrastructure to support major revisions.
Observability *System includes observability for fast detection and root cause identification of issues due to planned or unplanned change*	Observability and health monitoring covers only high-level aspects of system health.	...	...	...	AIOps for automated scalability, performance, resilience and capacity metrics health monitoring and alerting. Observability is built into the development process so full internal state of the application is exposed through logging and telemetry.

[a] External Dependency: Active Directory, NAS, offline data store, WAN

[b] Internal Dependency: database, internal system component, SAN, network switch/LAN

Continuous Integration/Continuous Delivery and Automation

An internal analysis conducted by Intel MIT revealed that 43% of availability impacts were triggered by system change or had change as a factor (see Figure 1 on page 2). Reliability Engineers played a crucial role in addressing this challenge by defining standards and design patterns aimed at improving the quality of changes delivered to the production environment.

A key initiative in this regard was the implementation of continuous integration/continuous delivery (CI/CD) practices and change automation. These approaches helped reduce human errors as well as errors in transferring software upgrades from the development factory to our high-volume manufacturing factories. They also allowed for a higher frequency of smaller changes, thus reducing the likelihood of major outage.

By implementing CI/CD and automation, we were able to focus on optimizing the degree of coupling between systems in the architecture. This was achieved through thoughtful functional partitioning and the use of middleware, which helped enhance the quality of changes by reducing dependencies. This enabled individual systems to change with a reduced risk of an impact that affects other systems. There is also less risk of impact due to a dependency being missed, such as another system needing an update as a prerequisite to the current change.

Configuration management emerged as a significant source of change failures in our high-volume factories. Software defects are typically found in the factory during beta testing, but failure in high-volume manufacturing is more likely due to an unexpected delta in configuration. Modern configuration management and Copy Exactly! audit systems can help prevent these failures.

Through the adoption of CI/CD practices, automation and improved configuration management, we aimed to mitigate the risks associated with system changes, increase the quality of changes and enhance the overall availability of critical automation systems.

Observability, Monitoring and AIOps

Our past experiences in manufacturing automation highlighted that 15–20% of impacts could have been prevented or minimized if emerging signals had been detected sooner. Additionally, inadequate post-change validation (referred to as a "watch-it plan") contributed to extended impacts in 40% of change failures. These challenges can be mitigated through effective observability solutions, including application performance management. We have made significant improvements in observability for manufacturing automation, particularly with the introduction of our Data Analytics Platforms for Observability. However, opportunities exist to further enhance observability.

One approach is to integrate observability modules and standards into applications during the development stage. This enables us to gather additional application state and performance data (including individual system availability metrics, performance, latency and error rates) as a standard practice.

Another aspect of enhancing operational efficiency and reliability is the application of artificial intelligence for IT Operations (AIOps). According to Gartner, "There is no future of IT Operations that does not include AIOps."[1] As data volumes and the pace of change continue to grow, reliance on AI becomes indispensable.

In the context of manufacturing automation, AIOps becomes increasingly critical as systems scale out for IDM 2.0 factories and business continuity planning. By leveraging machine-learning algorithms on system data, we can detect emerging signals, improve mean time to recovery and reduce human toil and error.

Incident Response and Continuous Learning

Reliability Engineers have introduced a cross-domain approach based on integrated knowledge of all systems. They actively lead incident troubleshooting, breaking down silos and fostering collaboration among teams to resolve issues promptly.

Reliability Engineers also contribute valuable tools and expertise to incident response scenarios. These include troubleshooting and recovery playbooks; structured problem-solving techniques like model-based problem solving; the establishment of investigation task force processes; and observability dashboards for thorough problem characterization, containment and root cause identification.

Reliability Engineers take charge of capturing learnings from incidents and adjusting playbooks accordingly. Even when the root cause remains unidentified, resilience solutions can be implemented to prevent future impact. Engineering resilience into automation systems can mitigate the effects of unforeseen issues.

The culture of continuous learning promoted by RE involves researching best practices for reliability and operations, adopting them within the organization. Our Reliability Engineers are constantly sharing these learnings and providing training to different teams in the IT automation function, with the goal of ingraining these approaches in the wider organizational culture.

Results

With Intel on track to reach 99.99% uptime, the people and practices behind our RE team—along with the culture of excellence and resilience embodied across Intel MIT operations and a close partnership with the Technology Development organization—are key enablers to achieving that goal. These results are not just due to the impact of RE—they represent the latest evolution of our long-term efforts to enhance manufacturing reliability through enhanced incident response, continuous learning, playbooks and so on.

- As a direct result of RE activities, standards for observability, change and resilience are being adopted across all our diverse automation systems.
- RE's use of FMEA to develop the RMM produced a tool we can apply to all our systems, providing a structured approach to the development of priority resilience improvements to each application.
- Using this approach, we identified over 200 resilience improvement projects and added them to our development roadmap for the next two years.
- As a result of these and other activities, unscheduled factory downtime has decreased by 50% from 2019 levels (see Figure 2).

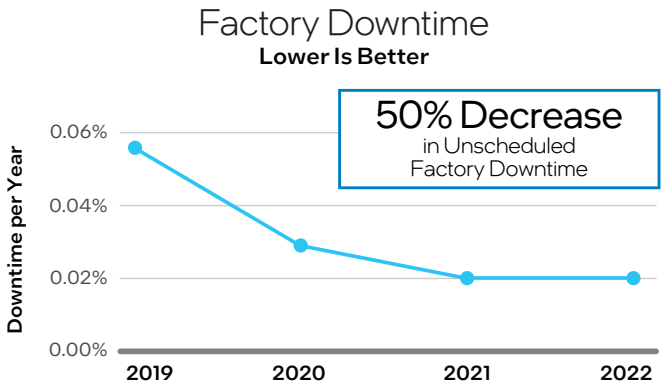

Figure 2. Factory downtime from 2019–2022.

[1] Gartner 2021 Market Guide for AIOps, gartner.com/en/documents/4000217

Conclusion

Our results show how an RE approach can extend the benefits of resilience to the manufacturing environment, preparing us for future adoption of cloud-based microservice environments. We have demonstrated that best-in-class reliability and availability of IT systems can be achieved by adopting a standard set of RE tools and proactively applying them to improve resilience.

Related Content

If you liked this paper, you may also be interested in these related stories:

- Minimizing Manufacturing Data Management Costs white paper
- Optimizing Factory Performance with Digital Twin Technology white paper
- Transforming Industrial Manufacturing with Software-Defined Networking white paper
- Accelerated Analytics Drives Breakthroughs in Factory Equipment Availability white paper
- Transforming Manufacturing Yield Analysis with AI white paper
- Streamline Deep-Learning Integration into Defect Classification white paper
- Optimizing Operations with Virtualized Industrial PCs white paper

For more information on Intel IT best practices, visit intel.com/IT.

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation on **Twitter** or **LinkedIn**. Visit us today at **intel.com/IT** if you would like to learn more.